

# ■ TypeScript Project Code Export

## ■ TypeScript/TSX Files & Safe Config:

- [TSX] app\admin\page.tsx
- [TS] app\api\addiction-images\route.ts
- [TS] app\api\allergies-image\route.ts
- [TS] app\api\comprehensive-report-data\route.ts
- [TS] app\api\digestive-images\route.ts
- [TS] app\api\lifestyle-images\route.ts
- [TS] app\api\lifestyle-upload\route.ts
- [TS] app\api\nutrition\route.ts
- [TS] app\api\patients-data\route.ts
- [TS] app\api\share-report\route.ts
- [TS] app\api\shared-access\route.ts
- [TS] app\api\sleep-image\route.ts
- [TS] app\api\sports-images\route.ts
- [TS] app\api\upload-image\route.ts
- [TS] app\api\upload-signature\route.ts
- [TSX] app\layout.tsx
- [TSX] app\page.tsx
- [TSX] app\report\[id]\page.tsx
- [TSX] app\report\pdf\page.tsx
- [TSX] app\report\preview\page.tsx
- [TSX] app\shared\[token]\page.tsx
- [TSX] components\admin\GenomicAnalysisAdmin.tsx
- [TSX] components\admin\ReportPreview.tsx
- [TSX] components\admin\allergies-sensitivity-admin.tsx
- [TSX] components\admin\content-admin.tsx
- [TSX] components\admin\diet-analysis-admin.tsx
- [TSX] components\admin\digestive-health-admin.tsx
- [TSX] components\admin\dynamic-diet-field-admin.tsx
- [TSX] components\admin\dynamic-field-helpers.tsx
- [TSX] components\admin\family-genetic-impact-admin.tsx
- [TSX] components\admin\gene-results-admin.tsx
- [TSX] components\admin\genes-addiction-admin.tsx
- [TSX] components\admin\genetic-parameters-admin.tsx
- [TSX] components\admin\health-summary-admin.tsx
- [TSX] components\admin\import-export-admin.tsx
- [TSX] components\admin\lifestyle-conditions-admin.tsx
- [TSX] components\admin\metabolic-core-admin.tsx
- [TSX] components\admin\nutrition-admin.tsx
- [TSX] components\admin\patient-info-admin.tsx
- [TSX] components\admin\pdf-generator.tsx
- [TSX] components\admin\preventive-health-admin.tsx
- [TSX] components\admin\report-settings-admin.tsx
- [TSX] components\admin\report-summaries-admin.tsx
- [TSX] components\admin\settings-admin.tsx
- [TSX] components\admin\share-management.tsx
- [TSX] components\admin\sleep-rest-admin.tsx
- [TSX] components\admin\sports-fitness-admin.tsx
- [TSX] components\comprehensive-report-viewer.tsx
- [TSX] components\theme-provider.tsx
- [TSX] components\ui\accordion.tsx
- [TSX] components\ui>alert-dialog.tsx
- [TSX] components\ui>alert.tsx
- [TSX] components\ui\aspect-ratio.tsx
- [TSX] components\ui\avatar.tsx
- [TSX] components\ui\badge.tsx
- [TSX] components\ui\breadcrumb.tsx
- [TSX] components\ui\button.tsx
- [TSX] components\ui\calendar.tsx
- [TSX] components\ui\card.tsx
- [TSX] components\ui\carousel.tsx
- [TSX] components\ui\chart.tsx
- [TSX] components\ui\checkbox.tsx
- [TSX] components\ui\collapsible.tsx
- [TSX] components\ui\command.tsx
- [TSX] components\ui\context-menu.tsx
- [TSX] components\ui\dialog.tsx
- [TSX] components\ui\drawer.tsx
- [TSX] components\ui\dropdown-menu.tsx
- [TSX] components\ui\form.tsx

- [TSX] components\ui\hover-card.tsx
- [TSX] components\ui\input-otp.tsx
- [TSX] components\ui\input.tsx
- [TSX] components\ui\label.tsx
- [TSX] components\ui\menubar.tsx
- [TSX] components\ui\navigation-menu.tsx
- [TSX] components\ui\pagination.tsx
- [TSX] components\ui\popover.tsx
- [TSX] components\ui\progress.tsx
- [TSX] components\ui\radio-group.tsx
- [TSX] components\ui\resizable.tsx
- [TSX] components\ui\scroll-area.tsx
- [TSX] components\ui\select.tsx
- [TSX] components\ui\separator.tsx
- [TSX] components\ui\sheet.tsx
- [TSX] components\ui\sidebar.tsx
- [TSX] components\ui\skeleton.tsx
- [TSX] components\ui\slider.tsx
- [TSX] components\ui\sonner.tsx
- [TSX] components\ui\switch.tsx
- [TSX] components\ui\table.tsx
- [TSX] components\ui\tabs.tsx
- [TSX] components\ui\textarea.tsx
- [TSX] components\ui\toast.tsx
- [TSX] components\ui\toaster.tsx
- [TSX] components\ui\toggle-group.tsx
- [TSX] components\ui\toggle.tsx
- [TSX] components\ui\tooltip.tsx
- [TSX] components\ui\use-mobile.tsx
- [TS] components\ui\use-toast.ts
- [TSX] hooks\use-mobile.tsx
- [TS] hooks\use-toast.ts
- [TS] lib\encryption.ts
- [TS] lib\prisma.ts
- [TS] lib\utils.ts
- [TS] next-env.d.ts
- [PKG] package.json
- [TS] scripts\migrate-data.ts
- [TS] tailwind.config.ts
- [TSC] tsconfig.json
- [TS] types\html2pdf.d.ts
- [TS] types\report-types.ts

## ■ File: app\admin\page.tsx

```
=====
1: "use client";
2:
3: import { Tabs, TabsContent, TabsList, TabsTrigger } from "@components/ui/tabs";
4: import { Button } from "@components/ui/button";
5: import { useState, useEffect } from "react";
6: import { useToast } from "@components/ui/use-toast";
7: import { Input } from "@components/ui/input";
8: import ComprehensiveReportViewer from "@components/comprehensive-report-viewer";
9: import ReportPreview from "@components/admin/ReportPreview";
10: import { useSearchParams, useRouter } from "next/navigation";
11: import ShareManagement from "@components/admin/share-management";
12:
13: import {
14:   Select,
15:   SelectContent,
16:   SelectItem,
17:   SelectTrigger,
18:   SelectValue,
19: } from "@components/ui/select";
20:
21: import type {
22:   Patient,
23:   Report,
24:   PatientInfo,
25:   ReportContent,
26:   ReportSettings,
27:   NutrientData,
28:   GeneTestResult,
29:   ReportSummaries,
30:   SportsAndFitness,
31:   LifestyleConditions,
32:   HealthConditionStatus,
33:   MetabolicCore,
34:   DigestiveHealth,
35:   DigestiveHealthEntry,
36:   GenesAndAddiction,
37:   SleepAndRest,
38:   SleepData,
39:   AllergiesAndSensitivity,
40:   PreventiveHealth,
41:   FamilyGeneticImpact,
42:   DynamicDietFieldDefinition,
43:   PatientDietAnalysisResult,
44:   GenomicAnalysisTable,
45:   GenomicCategoryGroup,
46:   HealthSummary,
47:   HealthSummaryEntry,
48:   ComprehensiveReportData,
49: } from "@types/report-types";
50:
51: // Corrected imports for admin components
52: import PatientInfoAdmin from "@components/admin/patient-info-admin";
53: import ContentAdmin from "@components/admin/content-admin";
54: import SettingsAdmin from "@components/admin/settings-admin";
55: import HealthSummaryAdmin from "@components/admin/health-summary-admin";
56: import DynamicDietFieldAdmin from "@components/admin/dynamic-diet-field-admin";
57: import GeneResultsAdmin from "@components/admin/gene-results-admin";
58: import ImportExportAdmin from "@components/admin/import-export-admin";
59: import NutritionAdmin from "@components/admin/nutrition-admin";
60: import SportsFitnessAdmin from "@components/admin/sports-fitness-admin";
61: import LifestyleConditionsAdmin from "@components/admin/lifestyle-conditions-admin";
62: import DigestiveHealthAdmin from "@components/admin/digestive-health-admin";
63: import GenesAddictionAdmin from "@components/admin/genes-addiction-admin";
64: import SleepRestAdmin from "@components/admin/sleep-rest-admin";
65: import AllergiesSensitivityAdmin from "@components/admin/allergies-sensitivity-admin";
66: import PreventiveHealthAdmin from "@components/admin/preventive-health-admin";
67: import FamilyGeneticImpactAdmin from "@components/admin/family-genetic-impact-admin";
68: import PDFGenerator from "@components/admin/pdf-generator";
69: import MetabolicCoreAdmin from "@components/admin/metabolic-core-admin";
70: import GeneticParametersAdmin from "@components/admin/genetic-parameters-admin";
71: import GenomicAnalysisAdmin from "@components/admin/GenomicAnalysisAdmin";
```

```

72:
73: const getEmptyReport = (name?: string): Report => ({
74:   id: crypto.randomUUID(),
75:   name: name || "", // Add the name field
76:   content: {
77:     introduction: "",
78:     genomicsExplanation: "",
79:     genesHealthImpact: "",
80:     fundamentalsPRS: "",
81:     utilityDoctors: "",
82:     microarrayExplanation: "",
83:     microarrayData: "",
84:   },
85:   settings: {
86:     title: "",
87:     subtitle: "",
88:     companyName: "",
89:     headerColor: "#000000",
90:     accentColor: "#000000",
91:     fonts: {
92:       primary: "Arial, sans-serif",
93:       secondary: "Georgia, serif",
94:       mono: "Courier New, monospace",
95:     },
96:   },
97:   dynamicDietFieldDefinitions: [],
98:   patientDietAnalysisResults: [],
99:   dietFieldCategories: [],
100:  nutritionData: {
101:    quote: "",
102:    description: "",
103:    data: {},
104:  },
105:  sportsAndFitness: {
106:    customImages: {},
107:  },
108:  lifestyleConditions: {},
109:  lifestyleCategoryImages: {},
110:  metabolicCore: {},
111:  digestiveHealth: {
112:    quote: "",
113:    description: "",
114:    data: {},
115:  },
116:  genesAndAddiction: {
117:    quote: "",
118:    description: "",
119:    data: {},
120:  },
121:  sleepAndRest: {
122:    quote: "",
123:    description: "",
124:    data: {},
125:  },
126:  allergiesAndSensitivity: {
127:    quote: "",
128:    description: "",
129:    generalAdvice: "",
130:    data: {},
131:  },
132:  geneTestResults: [],
133:  preventiveHealth: {
134:    diagnosticTests: {
135:      halfYearly: [],
136:      yearly: [],
137:    },
138:    nutritionalSupplements: [],
139:  },
140:  familyGeneticImpactSection: {
141:    description: "",
142:    impacts: [],
143:  },
144:  categories: [],

```

```

145:   summaries: {
146:     nutrigenomicsSummary: "",
147:     exerciseGenomicsSummary: "",
148:   },
149:   metabolicSummary: {
150:     strengths: [],
151:     weaknesses: [],
152:   },
153: });
154:
155: const getEmptyPatient = (): Patient => ({
156:   id: crypto.randomUUID(),
157:   info: {
158:     name: "",
159:     gender: "MALE",
160:     birthDate: "",
161:     sampleCode: "",
162:     sampleDate: "",
163:     reportDate: "",
164:     checkedBy: "",
165:     scientificContent: "",
166:     disclaimer: "",
167:     signature1: null,
168:     signature2: null,
169:   },
170:   reports: [getEmptyReport()],
171: });
172:
173: const getEmptyPatientInfo = (): PatientInfo => ({
174:   name: "",
175:   gender: "MALE",
176:   birthDate: "",
177:   sampleCode: "",
178:   sampleDate: "",
179:   reportDate: "",
180:   checkedBy: "",
181:   scientificContent: "",
182:   disclaimer: "",
183:   signature1: null,
184:   signature2: null,
185: });
186:
187: const AdminPage = () => {
188:   const [patients, setPatients] = useState<Patient[]>([]);
189:
190:   const searchParams = useSearchParams();
191:   const router = useRouter();
192:
193:   const initialPatientIndex = Number(searchParams.get("patientIndex")) || 0;
194:   const initialReportIndex = Number(searchParams.get("reportIndex")) || 0;
195:
196:   const [selectedPatientIndex, setSelectedPatientIndex] =
197:     useState(initialPatientIndex);
198:   const [selectedReportIndex, setSelectedReportIndex] =
199:     useState(initialReportIndex);
200:
201:   const [dataLoaded, setDataLoaded] = useState(false);
202:
203:   const selectedPatient = patients[selectedPatientIndex];
204:   const selectedReport = selectedPatient?.reports[selectedReportIndex];
205:   const [isLoading, setIsLoading] = useState(true);
206:   const [newSampleCode, setNewSampleCode] = useState("");
207:   const { toast } = useToast();
208:   const [newReportName, setNewReportName] = useState("");
209:   const [showAddReportModal, setShowAddReportModal] = useState(false);
210:
211:   const [activeTab, setActiveTab] = useState("");
212:
213:   // Load from localStorage on mount
214:
215:   useEffect(() => {
216:     const urlTab = searchParams.get("tab");
217:     if (urlTab) {

```

```

218:     setActiveTab(urlTab);
219:     return;
220:   }
221:   if (typeof window !== "undefined") {
222:     const saved = sessionStorage.getItem("activeTab");
223:     if (saved) setActiveTab(saved);
224:   }
225: }, [searchParams]);
226:
227: useEffect(() => {
228:   if (typeof window !== "undefined") {
229:     sessionStorage.setItem("activeTab", activeTab);
230:   }
231: }, [activeTab]);
232:
233: useEffect(() => {
234:   const params = new URLSearchParams(searchParams.toString());
235:   params.set("patientIndex", String(selectedPatientIndex));
236:   params.set("reportIndex", String(selectedReportIndex));
237:   params.set("tab", activeTab);
238:   router.replace(`${params.toString()}`, { scroll: false });
239: }, [selectedPatientIndex, selectedReportIndex, activeTab]);
240:
241: useEffect(() => {
242:   const len = selectedPatient?.reports?.length ?? 0;
243:   if (len === 0) return;
244:   if (selectedReportIndex >= len) {
245:     setSelectedReportIndex(len - 1);
246:   }
247: }, [selectedPatient?.reports?.length, selectedReportIndex]);
248:
249: // Load data from API on component mount
250: useEffect(() => {
251:   const fetchData = async () => {
252:     setIsLoading(true);
253:     try {
254:       const response = await fetch("/api/patients-data");
255:       if (response.ok) {
256:         const data = await response.json();
257:         if (Array.isArray(data) && data.length > 0) {
258:           setPatients(data);
259:           setDataLoaded(true);
260:         } else {
261:           // If the backend returns an empty list, treat as "no data"
262:           setPatients([]);
263:           setDataLoaded(true); // still loaded, but empty
264:         }
265:
266:         toast({
267:           description: "Report data has been loaded from the backend.",
268:           variant: "success",
269:           duration: 2000,
270:         });
271:       } else {
272:         // non-200 (server says nothing found or error)
273:         console.warn("No existing data found, starting with empty structure");
274:         setPatients([]);
275:         setDataLoaded(false);
276:         toast({
277:           title: "No existing data",
278:           description:
279:             "Starting with empty report structure. Add your data and save.",
280:           variant: "default",
281:           duration: 1500,
282:         });
283:       }
284:     } catch (error) {
285:       console.error("Error fetching report data:", error);
286:       toast({
287:         title: "Error fetching data",
288:         description:
289:           "Starting with empty report structure due to connection error.",
290:         variant: "destructive",

```

```

291:         duration: 1500,
292:     });
293:     } finally {
294:         setIsLoading(false);
295:     }
296: };
297: fetchData();
298: }, [toast]);
299:
300: // Add this useEffect after the existing one
301: useEffect(() => {
302:     if (!isLoading && patients.length === 0) {
303:         setPatients([getEmptyPatient()]);
304:     }
305: }, [isLoading, patients.length]);
306:
307: const addNewPatient = (sampleCode: string) => {
308:     const isDuplicate = patients.some(
309:         (patient) => patient.info.sampleCode === sampleCode
310:     );
311:
312:     if (isDuplicate) {
313:         toast({
314:             title: "Duplicate Sample Code",
315:             description: "A patient with this sample code already exists.",
316:             variant: "destructive",
317:             duration: 1500,
318:         });
319:         return;
320:     }
321:
322:     const newPatient: Patient = {
323:         id: crypto.randomUUID(),
324:         info: {
325:             name: "",
326:             gender: "MALE",
327:             birthDate: "",
328:             sampleCode,
329:             sampleDate: "",
330:             reportDate: "",
331:             checkedBy: "",
332:             scientificContent: "",
333:             disclaimer: "",
334:             signature1: null,
335:             signature2: null,
336:         },
337:         reports: [getEmptyReport()],
338:     };
339:
340:     setPatients((prev) => [...prev, newPatient]);
341:     setSelectedPatientIndex(patients.length); // switch to new
342:     setSelectedReportIndex(0);
343: };
344:
345: const findPatientBySampleCode = (sampleCode: string): Patient | undefined => {
346:     return patients.find((patient) => patient.info.sampleCode === sampleCode);
347: };
348:
349: // Function to handle the add report button click
350: const handleAddReportClick = () => {
351:     setShowAddReportModal(true);
352: };
353:
354: // Function to handle the modal form submission
355: const handleAddReportSubmit = () => {
356:     const reportName = newReportName.trim();
357:     if (!reportName) {
358:         toast({
359:             title: "Report Name Required",
360:             description: "Please enter a name for the new report.",
361:             variant: "destructive",
362:             duration: 1500,
363:         });

```

```

364:     return;
365: }
366:
367: const existingReportNames = selectedPatient.reports.map((r) =>
368:   r.name?.toLowerCase()
369: );
370: if (existingReportNames.includes(reportName.toLowerCase())) {
371:   toast({
372:     title: "Duplicate Report Name",
373:     description: "A report with this name already exists for this patient.",
374:     variant: "destructive",
375:     duration: 1500,
376:   });
377:   return;
378: }
379:
380: addReportToSelectedPatient(reportName);
381: setNewReportName("");
382: setShowAddReportModal(false);
383: };
384:
385: // Function to update report name
386: const updateReportName = (newName: string) => {
387:   setPatients((prev) => {
388:     const updated = [...prev];
389:     const patient = { ...updated[selectedPatientIndex] };
390:     const reports = [...patient.reports];
391:
392:     reports[selectedReportIndex] = {
393:       ...reports[selectedReportIndex],
394:       name: newName,
395:     };
396:
397:     updated[selectedPatientIndex] = { ...patient, reports };
398:     return updated;
399:   });
400: };
401:
402: const deleteSelectedReport = async () => {
403:   if (!selectedPatient) return;
404:
405:   const confirmed = confirm("Are you sure you want to delete this report?");
406:   if (!confirmed) return;
407:
408:   if (selectedPatient.reports.length <= 1) {
409:     alert("At least one report must remain for a patient.");
410:     return;
411:   }
412:
413:   const reportToDelete = selectedPatient.reports[selectedReportIndex];
414:   if (!reportToDelete) return;
415:
416:   const deletedReportName =
417:     reportToDelete.name || `Report ${selectedReportIndex + 1}`;
418:
419:   try {
420:     // Call API to delete in DB
421:     await fetch(`/api/patients-data?reportId=${reportToDelete.id}`, {
422:       method: "DELETE",
423:     });
424:
425:     // Update local state
426:     setPatients((prev) => {
427:       const updated = [...prev];
428:       const patient = { ...updated[selectedPatientIndex] };
429:       const newReports = patient.reports.filter(
430:         (r) => r.id !== reportToDelete.id
431:       );
432:
433:       updated[selectedPatientIndex] = { ...patient, reports: newReports };
434:       return updated;
435:     });
436:

```



```

437:     setSelectedReportIndex(
438:         selectedReportIndex < selectedPatient.reports.length - 1
439:         ? selectedReportIndex
440:         : selectedReportIndex - 1
441:     );
442:
443:     toast({
444:         title: "Report Deleted",
445:         description: `Report "${deletedReportName}" has been removed.`,
446:         variant: "destructive",
447:         duration: 2000,
448:     });
449: } catch (err) {
450:     toast({
451:         title: "Error",
452:         description: "Failed to delete the report.",
453:         variant: "destructive",
454:         duration: 2000,
455:     });
456: }
457: };
458:
459: const deleteSelectedPatient = async () => {
460:     if (!selectedPatient) return;
461:
462:     const confirmed = confirm("Are you sure you want to delete this patient?");
463:     if (!confirmed) return;
464:
465:     if (patients.length <= 1) {
466:         alert("At least one patient must remain.");
467:         return;
468:     }
469:
470:     const patientToDelete = selectedPatient;
471:     const deletedPatientName =
472:         patientToDelete.info?.name || `Patient ${selectedPatientIndex + 1}`;
473:
474:     try {
475:         // Call API to delete in DB
476:         await fetch(`/api/patients-data?patientId=${patientToDelete.id}`, {
477:             method: "DELETE",
478:         });
479:
480:         // Update local state
481:         setPatients((prev) => {
482:             const updated = [...prev];
483:             updated.splice(selectedPatientIndex, 1);
484:
485:             // Adjust selection
486:             const newIndex = Math.max(0, selectedPatientIndex - 1);
487:             setSelectedPatientIndex(newIndex);
488:             setSelectedReportIndex(0);
489:
490:             return updated;
491:         });
492:
493:         toast({
494:             title: "Patient Deleted",
495:             description: `Patient "${deletedPatientName}" has been removed.`,
496:             variant: "destructive",
497:             duration: 2000,
498:         });
499:     } catch (err) {
500:         toast({
501:             title: "Error",
502:             description: "Failed to delete the patient.",
503:             variant: "destructive",
504:             duration: 2000,
505:         });
506:     }
507: };
508:
509: const addReportToSelectedPatient = (reportName: string = "") => {

```

```

510:     let newReportIndex = 0;
511:     let updatedPatientName = "";
512:
513:     setPatients((prev) => {
514:         const updated = [...prev];
515:         const patient = { ...updated[selectedPatientIndex] };
516:
517:         // Create a completely new report with the specified name
518:         const newReport = getEmptyReport(reportName);
519:
520:         // Create a new reports array and add the new report at the end
521:         const newReports = [...patient.reports, newReport];
522:
523:         // Update the patient with the new reports array
524:         const updatedPatient = {
525:             ...patient,
526:             reports: newReports,
527:         };
528:
529:         updated[selectedPatientIndex] = updatedPatient;
530:
531:         // Calculate the index of the newly added report
532:         newReportIndex = newReports.length - 1;
533:         updatedPatientName = patient.info.name || "the patient";
534:
535:         return updated;
536:     });
537:
538:     // Set the selected report index to the newly created report
539:     setSelectedReportIndex(newReportIndex);
540:
541:     toast({
542:         title: "New Report Added",
543:         description: `Report "${
544:             reportName || `Report ${newReportIndex + 1}`
545:         }" was added to ${updatedPatientName}.`,
546:         variant: "success",
547:         duration: 1500,
548:     });
549: };
550:
551: const saveReportData = async () => {
552:     try {
553:         const response = await fetch("/api/patients-data", {
554:             method: "POST",
555:             headers: {
556:                 "Content-Type": "application/json",
557:             },
558:             body: JSON.stringify(patients),
559:         });
560:
561:         if (response.ok) {
562:             toast({
563:                 title: "Report data saved!",
564:                 description: "Your progress has been saved to the backend.",
565:                 variant: "success",
566:                 duration: 1500,
567:             });
568:         } else {
569:             const errorData = await response.json();
570:             toast({
571:                 title: "Failed to save data",
572:                 description:
573:                     errorData.error || "An error occurred while saving data.",
574:                 variant: "destructive",
575:                 duration: 1500,
576:             });
577:         }
578:     } catch (error) {
579:         console.error("Error saving report data:", error);
580:         toast({
581:             title: "Error saving data",
582:             description: "An unexpected error occurred while saving data.",

```

```

583:         variant: "destructive",
584:         duration: 1500,
585:     });
586: }
587: };
588:
589: const resetSection = (sectionKey: keyof Report) => {
590:     setPatients((prev) => {
591:         const updated = [...prev];
592:         const patient = { ...updated[selectedPatientIndex] };
593:         const reports = [...patient.reports];
594:         const report = { ...reports[selectedReportIndex] };
595:
596:         const emptyReport = getEmptyReport();
597:
598:         // report[sectionKey] = emptyReport[sectionKey] as Report[typeof sectionKey];
599:
600:         reports[selectedReportIndex] = report;
601:         updated[selectedPatientIndex] = { ...patient, reports };
602:         return updated;
603:     });
604: };
605:
606: const resetPatientInfo = () => {
607:     setPatients((prev) => {
608:         const updated = [...prev];
609:         updated[selectedPatientIndex] = {
610:             ...updated[selectedPatientIndex],
611:             info: getEmptyPatientInfo(),
612:         };
613:         return updated;
614:     });
615: };
616:
617: // Patient Info Update Functions
618: const updatePatientInfo = (field: keyof PatientInfo, value: string) => {
619:     setPatients((prevPatients) => {
620:         const newPatients = [...prevPatients];
621:         newPatients[selectedPatientIndex] = {
622:             ...newPatients[selectedPatientIndex],
623:             info: {
624:                 ...newPatients[selectedPatientIndex].info,
625:                 [field]: value,
626:             },
627:         };
628:         return newPatients;
629:     });
630: };
631:
632: // Report Content Update Functions
633: const updateReportContent = (field: keyof ReportContent, value: string) => {
634:     setPatients((prev) => {
635:         const updated = [...prev];
636:         const patient = { ...updated[selectedPatientIndex] };
637:         const reports = [...patient.reports];
638:         const report = {
639:             ...reports[selectedReportIndex],
640:             content: {
641:                 ...reports[selectedReportIndex].content,
642:                 [field]: value,
643:             },
644:         };
645:         reports[selectedReportIndex] = report;
646:         updated[selectedPatientIndex] = { ...patient, reports };
647:         return updated;
648:     });
649: };
650:
651: // Report Settings Update Functions
652: const updateReportSettings = <K extends keyof ReportSettings>(
653:     field: K,
654:     value: ReportSettings[K]
655: ) => {

```

```

656:     setPatients((prev) => {
657:         const updated = [...prev];
658:         const patient = { ...updated[selectedPatientIndex] };
659:         const reports = [...patient.reports];
660:         const currentSettings = reports[selectedReportIndex].settings;
661:
662:         const report = {
663:             ...reports[selectedReportIndex],
664:             settings: {
665:                 ...currentSettings,
666:                 [field]: value,
667:             },
668:         };
669:
670:         reports[selectedReportIndex] = report;
671:         updated[selectedPatientIndex] = { ...patient, reports };
672:         return updated;
673:     });
674: };
675:
676: // ? Helper functions to add to your parent component
677:
678: const updateHealthSummaryDescription = (
679:     patientIndex: number,
680:     reportIndex: number,
681:     value: string
682: ) => {
683:     setPatients((prev) => {
684:         const updatedPatients = [...prev];
685:         const updatedReports = [...updatedPatients[patientIndex].reports];
686:         const report = { ...updatedReports[reportIndex] };
687:
688:         if (!report.healthSummary) {
689:             report.healthSummary = { description: "", data: [] };
690:         }
691:
692:         report.healthSummary = {
693:             ...report.healthSummary,
694:             description: value,
695:         };
696:
697:         updatedReports[reportIndex] = report;
698:         updatedPatients[patientIndex] = {
699:             ...updatedPatients[patientIndex],
700:             reports: updatedReports,
701:         };
702:
703:         return updatedPatients;
704:     });
705: };
706:
707: const updateHealthSummaryEntry = (
708:     patientIndex: number,
709:     reportIndex: number,
710:     entryIndex: number,
711:     field: keyof HealthSummaryEntry,
712:     value: string
713: ) => {
714:     setPatients((prev) => {
715:         const updatedPatients = [...prev];
716:         const updatedReports = [...updatedPatients[patientIndex].reports];
717:         const report = { ...updatedReports[reportIndex] };
718:
719:         if (!report.healthSummary) {
720:             report.healthSummary = { description: "", data: [] };
721:         }
722:
723:         const updatedData = [...report.healthSummary.data];
724:         if (!updatedData[entryIndex]) {
725:             updatedData[entryIndex] = { title: "", description: "" };
726:         }
727:         updatedData[entryIndex] = {
728:             ...updatedData[entryIndex],

```

```

729:         [field]: value,
730:     };
731:
732:     report.healthSummary = {
733:         ...report.healthSummary,
734:         data: updatedData,
735:     };
736:
737:     updatedReports[reportIndex] = report;
738:     updatedPatients[patientIndex] = {
739:         ...updatedPatients[patientIndex],
740:         reports: updatedReports,
741:     };
742:
743:     return updatedPatients;
744: });
745: };
746:
747: const addHealthSummaryEntry = (patientIndex: number, reportIndex: number) => {
748:     setPatients((prev) => {
749:         const updatedPatients = [...prev];
750:         const updatedReports = [...updatedPatients[patientIndex].reports];
751:         const report = { ...updatedReports[reportIndex] };
752:
753:         if (!report.healthSummary) {
754:             report.healthSummary = { description: "", data: [] };
755:         }
756:
757:         const updatedData = [
758:             ...report.healthSummary.data,
759:             { title: "", description: "", quote: "" },
760:         ];
761:
762:         report.healthSummary = {
763:             ...report.healthSummary,
764:             data: updatedData,
765:         };
766:
767:         updatedReports[reportIndex] = report;
768:         updatedPatients[patientIndex] = {
769:             ...updatedPatients[patientIndex],
770:             reports: updatedReports,
771:         };
772:
773:         return updatedPatients;
774:     });
775: };
776:
777: const removeHealthSummaryEntry = (
778:     patientIndex: number,
779:     reportIndex: number,
780:     entryIndex: number
781: ) => {
782:     setPatients((prev) => {
783:         const updatedPatients = [...prev];
784:         const updatedReports = [...updatedPatients[patientIndex].reports];
785:         const report = { ...updatedReports[reportIndex] };
786:
787:         if (report.healthSummary) {
788:             const updatedData = report.healthSummary.data.filter(
789:                 (_, idx) => idx !== entryIndex
790:             );
791:             report.healthSummary = {
792:                 ...report.healthSummary,
793:                 data: updatedData,
794:             };
795:         }
796:
797:         updatedReports[reportIndex] = report;
798:         updatedPatients[patientIndex] = {
799:             ...updatedPatients[patientIndex],
800:             reports: updatedReports,
801:         };

```

```

802:
803:     return updatedPatients;
804:   });
805: };
806:
807: // Dynamic Diet Field Functions
808: const updateDynamicDietFieldDefinitions = (
809:   definitions: DynamicDietFieldDefinition[]
810: ) => {
811:   setPatients((prev) => {
812:     const updated = [...prev];
813:     const patient = { ...updated[selectedPatientIndex] };
814:     const reports = [...patient.reports];
815:
816:     // Replace the entire array
817:     reports[selectedReportIndex] = {
818:       ...reports[selectedReportIndex],
819:       dynamicDietFieldDefinitions: definitions,
820:     };
821:
822:     updated[selectedPatientIndex] = { ...patient, reports };
823:     return updated;
824:   });
825: };
826:
827: const updatePatientDietAnalysisResult = (
828:   fieldId: string,
829:   data: Partial<PatientDietAnalysisResult>
830: ) => {
831:   setPatients((prev) => {
832:     const updated = [...prev];
833:     const patient = { ...updated[selectedPatientIndex] };
834:     const reports = [...patient.reports];
835:     const currentReport = { ...reports[selectedReportIndex] };
836:     const existingIndex = currentReport.patientDietAnalysisResults.findIndex(
837:       (r) => r.fieldId === fieldId
838:     );
839:
840:     const newResults = [...currentReport.patientDietAnalysisResults];
841:
842:     if (existingIndex > -1) {
843:       newResults[existingIndex] = {
844:         ...newResults[existingIndex],
845:         ...data,
846:       };
847:     } else {
848:       newResults.push({
849:         fieldId,
850:         score: 0,
851:         level: "NORMAL",
852:         recommendation: "",
853:         recommendations: {
854:           LOW: "",
855:           NORMAL: "",
856:           HIGH: "",
857:         },
858:         selectedLevel: "NORMAL",
859:         ...data,
860:       });
861:     }
862:
863:     reports[selectedReportIndex] = {
864:       ...currentReport,
865:       patientDietAnalysisResults: newResults,
866:     };
867:
868:     updated[selectedPatientIndex] = { ...patient, reports };
869:     return updated;
870:   });
871: };
872:
873: // Nutrition Data Update Functions
874: const updateNutritionData = (

```

```

875:     section: string,
876:     field: string,
877:     data: Partial<NutrientData> | string | Record<string, NutrientData>
878: ) => {
879:     setPatients((prev) => {
880:         const updated = [...prev];
881:         const patient = { ...updated[selectedPatientIndex] };
882:         const reports = [...patient.reports];
883:         const report = { ...reports[selectedReportIndex] };
884:
885:         const nutritionData = report.nutritionData || {
886:             quote: "",
887:             description: "",
888:             data: {},
889:         };
890:
891:         // ? FIXED: Handle quote/description updates
892:         if (section === "quote" || section === "description") {
893:             report.nutritionData = {
894:                 ...nutritionData,
895:                 [section]: data as string,
896:             };
897:         }
898:         // ? FIXED: Handle section deletion
899:         else if (field === "" && data === "__delete__") {
900:             const updatedData = { ...nutritionData.data };
901:             delete updatedData[section];
902:             report.nutritionData = {
903:                 ...nutritionData,
904:                 data: updatedData,
905:             };
906:         }
907:         // ? FIXED: Handle field updates and section replacement
908:         else {
909:             const currentSection = nutritionData.data?.[section] || {};
910:             let updatedSection;
911:
912:             // Replace entire section
913:             if (field === "") {
914:                 updatedSection = data as Record<string, NutrientData>;
915:             }
916:             // Update specific field
917:             else {
918:                 const prevField = currentSection[field] || {};
919:                 updatedSection = {
920:                     ...currentSection,
921:                     [field]: {
922:                         ...prevField,
923:                         ...(data as Partial<NutrientData>),
924:                     },
925:                 };
926:             }
927:
928:             report.nutritionData = {
929:                 ...nutritionData,
930:                 data: {
931:                     ...nutritionData.data,
932:                     [section]: updatedSection,
933:                 },
934:             };
935:         }
936:
937:         reports[selectedReportIndex] = report;
938:         updated[selectedPatientIndex] = { ...patient, reports };
939:         return updated;
940:     });
941: };
942:
943: // Sports and Fitness Update Functions
944: const updateSportsAndFitness = (
945:     section: keyof SportsAndFitness | "customImages" | "quote" | "description",
946:     field: string,
947:     data:

```

```

948:         | Partial<{ level: string; description: string }>
949:         | { label: string; url: string }
950:         | string
951:         | null
952:     ) => {
953:         setPatients((prev) => {
954:             const updated = [...prev];
955:             const patient = { ...updated[selectedPatientIndex] };
956:             const reports = [...patient.reports];
957:             const report = { ...reports[selectedReportIndex] };
958:             const current = { ...(report.sportsAndFitness || {}) };
959:
960:             // Handle quote or description update
961:             if (section === "quote" || section === "description") {
962:                 report.sportsAndFitness = {
963:                     ...current,
964:                     [section]: data as string,
965:                 };
966:             }
967:             // Handle custom image update
968:             else if (section === "customImages") {
969:                 const { label, url } = data as { label: string; url: string };
970:                 report.sportsAndFitness = {
971:                     ...current,
972:                     customImages: {
973:                         ...(current.customImages || {}),
974:                         [label]: url,
975:                     },
976:                 };
977:             }
978:             // Handle category deletion
979:             else if (field === "__delete_category__") {
980:                 const { [section]: _, ...rest } = current;
981:                 report.sportsAndFitness = rest;
982:             }
983:             // Handle new category creation
984:             else if (field === "__init__") {
985:                 report.sportsAndFitness = {
986:                     ...current,
987:                     [section]: {},
988:                 };
989:             }
990:             // Handle field deletion (NEW: check for null data)
991:             else if (data === null) {
992:                 const category = current[section] as Record<
993:                     string,
994:                     { level: string; description: string }
995:                 >;
996:                 if (
997:                     category &&
998:                     typeof category === "object" &&
999:                     !Array.isArray(category)
1000:                 ) {
1001:                     const { [field]: _, ...rest } = category;
1002:                     report.sportsAndFitness = {
1003:                         ...current,
1004:                         [section]: rest,
1005:                     };
1006:                 }
1007:             }
1008:             // Handle field deletion (OLD: for backward compatibility)
1009:             else if (field.startsWith("__delete_field_")) {
1010:                 const fieldKey = field.replace("__delete_field_", "");
1011:                 const category = current[section] as Record<
1012:                     string,
1013:                     { level: string; description: string }
1014:                 >;
1015:                 if (
1016:                     category &&
1017:                     typeof category === "object" &&
1018:                     !Array.isArray(category)
1019:                 ) {
1020:                     const { [fieldKey]: _, ...rest } = category;

```



```

1021:         report.sportsAndFitness = {
1022:             ...current,
1023:             [section]: rest,
1024:         };
1025:     }
1026: }
1027: // Handle field addition or update
1028: else {
1029:     const sectionData = current[section] as
1030:     | Record<string, { level: string; description: string }>
1031:     | undefined;
1032:     report.sportsAndFitness = {
1033:         ...current,
1034:         [section]: {
1035:             ...(sectionData || {}),
1036:             [field]: {
1037:                 ...(sectionData?.[field] || {}),
1038:                 ...(data as { level: string; description: string }),
1039:             },
1040:         },
1041:     };
1042: }
1043:
1044: reports[selectedReportIndex] = report;
1045: updated[selectedPatientIndex] = { ...patient, reports };
1046: return updated;
1047: });
1048: };
1049:
1050: // Lifestyle Conditions Update Functions
1051: const updateFieldStatus = (
1052:     categoryId: string,
1053:     fieldName: string,
1054:     updated: Partial<HealthConditionStatus>
1055: ) => {
1056:     setPatients((prev) => {
1057:         const updatedPatients = [...prev];
1058:         const patient = { ...updatedPatients[selectedPatientIndex] };
1059:         const reports = [...patient.reports];
1060:         const report = { ...reports[selectedReportIndex] };
1061:
1062:         const prevConditions = report.lifestyleConditions || {};
1063:
1064:         const prevCategory = prevConditions[categoryId];
1065:         const safeCategory =
1066:             typeof prevCategory === "object" && prevCategory !== null
1067:             ? prevCategory
1068:             : {};
1069:
1070:         const existingField = safeCategory[fieldName] || {
1071:             status: "strength",
1072:             description: "",
1073:             sensitivity: "low",
1074:             avoid: [],
1075:             follow: [],
1076:             consume: [],
1077:             monitor: [],
1078:             avoidLabel: "AVOID",
1079:             followLabel: "FOLLOW",
1080:             consumeLabel: "CONSUME",
1081:             monitorLabel: "MONITOR",
1082:         };
1083:
1084:         const updatedField = {
1085:             ...existingField,
1086:             ...updated,
1087:         };
1088:
1089:         const updatedCategory = {
1090:             ...safeCategory,
1091:             [fieldName]: updatedField,
1092:         };
1093:

```

```

1094:     const updatedConditions = {
1095:         ...prevConditions,
1096:         [categoryId]: updatedCategory,
1097:     };
1098:
1099:     report.lifestyleConditions = updatedConditions;
1100:     reports[selectedReportIndex] = report;
1101:     updatedPatients[selectedPatientIndex] = { ...patient, reports };
1102:     return updatedPatients;
1103: });
1104: };
1105:
1106: // Metabolic Core Update Functions
1107: const updateMetabolicCore = (area: string, data: any) => {
1108:     setPatients((prev) => {
1109:         const updated = [...prev];
1110:         const patient = { ...updated[selectedPatientIndex] };
1111:         const reports = [...patient.reports];
1112:         const currentReport = { ...reports[selectedReportIndex] };
1113:
1114:         if (area === "quote" || area === "description") {
1115:             currentReport.metabolicCore = {
1116:                 ...currentReport.metabolicCore,
1117:                 [area]: data,
1118:             };
1119:         } else {
1120:             currentReport.metabolicCore = {
1121:                 ...currentReport.metabolicCore,
1122:                 [area]: {
1123:                     ...(currentReport.metabolicCore[area] || {}),
1124:                     ...data,
1125:                 },
1126:             };
1127:         }
1128:
1129:         reports[selectedReportIndex] = currentReport;
1130:         updated[selectedPatientIndex] = { ...patient, reports };
1131:         return updated;
1132:     });
1133: };
1134:
1135: // Digestive Health Update Functions
1136: const updateDigestiveHealth = (
1137:     key: string,
1138:     data: Partial<DigestiveHealthEntry>
1139: ) => {
1140:     setPatients((prev) => {
1141:         const updated = [...prev];
1142:         const patient = { ...updated[selectedPatientIndex] };
1143:         const reports = [...patient.reports];
1144:         const current = reports[selectedReportIndex];
1145:
1146:         const existingField = current.digestiveHealth.data[key] || {
1147:             title: "",
1148:             icon: "",
1149:             sensitivity: "",
1150:             good: "",
1151:             bad: "",
1152:         };
1153:
1154:         const newDigestiveHealth = {
1155:             ...current.digestiveHealth,
1156:             data: {
1157:                 ...current.digestiveHealth.data,
1158:                 [key]: {
1159:                     ...existingField,
1160:                     ...data,
1161:                 },
1162:             },
1163:         };
1164:
1165:         reports[selectedReportIndex] = {
1166:             ...current,

```

```

1167:         digestiveHealth: newDigestiveHealth,
1168:     };
1169:     updated[selectedPatientIndex] = { ...patient, reports };
1170:     return updated;
1171: });
1172: };
1173:
1174: const addDigestiveField = (key: string) => {
1175:     setPatients((prev) => {
1176:         const updated = [...prev];
1177:         const patient = { ...updated[selectedPatientIndex] };
1178:         const reports = [...patient.reports];
1179:
1180:         const report = { ...reports[selectedReportIndex] };
1181:         report.digestiveHealth = {
1182:             ...report.digestiveHealth,
1183:             data: {
1184:                 ...report.digestiveHealth.data,
1185:                 [key]: {
1186:                     title: key,
1187:                     icon: "",
1188:                     sensitivity: "",
1189:                     good: "",
1190:                     bad: "",
1191:                 },
1192:             },
1193:         };
1194:
1195:         reports[selectedReportIndex] = report;
1196:         updated[selectedPatientIndex] = { ...patient, reports };
1197:         return updated;
1198:     });
1199: };
1200:
1201: const deleteDigestiveField = (key: string) => {
1202:     setPatients((prev) => {
1203:         const updated = [...prev];
1204:         const patient = { ...updated[selectedPatientIndex] };
1205:         const reports = [...patient.reports];
1206:
1207:         const report = { ...reports[selectedReportIndex] };
1208:         const updatedData = { ...report.digestiveHealth.data };
1209:         delete updatedData[key];
1210:
1211:         report.digestiveHealth = {
1212:             ...report.digestiveHealth,
1213:             data: updatedData,
1214:         };
1215:
1216:         reports[selectedReportIndex] = report;
1217:         updated[selectedPatientIndex] = { ...patient, reports };
1218:         return updated;
1219:     });
1220: };
1221:
1222: const updateDigestiveQuoteAndDescription = (
1223:     quote: string,
1224:     description: string
1225: ) => {
1226:     setPatients((prev) => {
1227:         const updated = [...prev];
1228:         const patient = { ...updated[selectedPatientIndex] };
1229:         const reports = [...patient.reports];
1230:
1231:         const report = {
1232:             ...reports[selectedReportIndex],
1233:             digestiveHealth: {
1234:                 ...reports[selectedReportIndex].digestiveHealth,
1235:                 quote,
1236:                 description,
1237:             },
1238:         };
1239:

```

```

1240:     reports[selectedReportIndex] = report;
1241:     updated[selectedPatientIndex] = { ...patient, reports };
1242:     return updated;
1243:   });
1244: };
1245:
1246: // Genes and Addiction Update Functions
1247: const updateGenesAndAddiction = (
1248:   field: keyof GenesAndAddiction["data"],
1249:   data: Partial<GenesAndAddiction>
1250: ) => {
1251:   setPatients((prev) => {
1252:     const updated = [...prev];
1253:     const patient = { ...updated[selectedPatientIndex] };
1254:     const reports = [...patient.reports];
1255:     const currentReport = { ...reports[selectedReportIndex] };
1256:
1257:     const addictionData = {
1258:       ...currentReport.genesAndAddiction.data,
1259:       [field]: {
1260:         ...(currentReport.genesAndAddiction.data?.[field] || {}),
1261:         ...data,
1262:       },
1263:     };
1264:
1265:     currentReport.genesAndAddiction = {
1266:       ...currentReport.genesAndAddiction,
1267:       data: addictionData,
1268:     };
1269:
1270:     reports[selectedReportIndex] = currentReport;
1271:     updated[selectedPatientIndex] = { ...patient, reports };
1272:     return updated;
1273:   });
1274: };
1275:
1276: // Sleep and Rest Update Functions
1277: const updateSleepAndRest = (
1278:   field: keyof SleepAndRest["data"],
1279:   data: Partial<SleepData>
1280: ) => {
1281:   setPatients((prev) => {
1282:     const updated = [...prev];
1283:     const patient = { ...updated[selectedPatientIndex] };
1284:     const reports = [...patient.reports];
1285:     const report = {
1286:       ...reports[selectedReportIndex],
1287:       sleepAndRest: {
1288:         ...reports[selectedReportIndex].sleepAndRest,
1289:         data: {
1290:           ...reports[selectedReportIndex].sleepAndRest.data,
1291:           [field]: {
1292:             ...(reports[selectedReportIndex].sleepAndRest.data?.[field] || {}),
1293:             ...data,
1294:           },
1295:         },
1296:       },
1297:     };
1298:
1299:     reports[selectedReportIndex] = report;
1300:     updated[selectedPatientIndex] = { ...patient, reports };
1301:     return updated;
1302:   });
1303: };
1304:
1305: // Allergies and Sensitivity Update Functions
1306: const updateAllergiesAndSensitivity = (
1307:   data: Partial<AllergiesAndSensitivity>
1308: ) => {
1309:   setPatients((prev) => {
1310:     const updated = [...prev];
1311:     const patient = { ...updated[selectedPatientIndex] };
1312:     const reports = [...patient.reports];

```

```

1313:     const report = {
1314:       ...reports[selectedReportIndex],
1315:       allergiesAndSensitivity: {
1316:         ...reports[selectedReportIndex].allergiesAndSensitivity,
1317:         ...data,
1318:       },
1319:     };
1320:     reports[selectedReportIndex] = report;
1321:     updated[selectedPatientIndex] = { ...patient, reports };
1322:     return updated;
1323:   });
1324: };
1325:
1326: // Preventive Health Update Functions
1327: const updatePreventiveHealth = (
1328:   section: keyof PreventiveHealth,
1329:   data: any
1330: ) => {
1331:   setPatients((prev) => {
1332:     const updated = [...prev];
1333:     const patient = { ...updated[selectedPatientIndex] };
1334:     const reports = [...patient.reports];
1335:
1336:     // Get the current preventiveHealth, or use defaults
1337:     const currentPreventiveHealth = reports[selectedReportIndex]
1338:       .preventiveHealth ?? {
1339:         description: "",
1340:         diagnosticTests: { halfYearly: [], yearly: [] },
1341:         nutritionalSupplements: [],
1342:       };
1343:
1344:     const report = {
1345:       ...reports[selectedReportIndex],
1346:       preventiveHealth: {
1347:         ...currentPreventiveHealth,
1348:         [section]: data,
1349:       },
1350:     };
1351:
1352:     reports[selectedReportIndex] = report;
1353:     updated[selectedPatientIndex] = { ...patient, reports };
1354:     return updated;
1355:   });
1356: };
1357:
1358: // Family Genetic Impact Functions
1359: const addFamilyGeneticImpact = () => {
1360:   setPatients((prev) => {
1361:     const updated = [...prev];
1362:     const patient = { ...updated[selectedPatientIndex] };
1363:     const reports = [...patient.reports];
1364:     const currentReport = reports[selectedReportIndex];
1365:
1366:     // Get current section or create default
1367:     const currentSection = currentReport.familyGeneticImpactSection || {
1368:       description: "",
1369:       impacts: [],
1370:     };
1371:
1372:     const impacts = [...currentSection.impacts];
1373:
1374:     impacts.push({
1375:       gene: "",
1376:       normalAlleles: "",
1377:       yourResult: "",
1378:       healthImpact: "",
1379:     });
1380:
1381:     reports[selectedReportIndex] = {
1382:       ...currentReport,
1383:       familyGeneticImpactSection: {
1384:         ...currentSection,
1385:         impacts: impacts,

```

```

1386:     },
1387:   };
1388:
1389:   updated[selectedPatientIndex] = { ...patient, reports };
1390:   return updated;
1391: });
1392: };
1393:
1394: const updateFamilyGeneticImpact = (
1395:   index: number,
1396:   field: keyof FamilyGeneticImpact,
1397:   value: string
1398: ) => {
1399:   setPatients((prev) => {
1400:     const updated = [...prev];
1401:     const patient = { ...updated[selectedPatientIndex] };
1402:     const reports = [...patient.reports];
1403:     const currentReport = reports[selectedReportIndex];
1404:
1405:     // Get current section or create default
1406:     const currentSection = currentReport.familyGeneticImpactSection || {
1407:       description: "",
1408:       impacts: [],
1409:     };
1410:
1411:     const impacts = [...currentSection.impacts];
1412:     impacts[index] = { ...impacts[index], [field]: value };
1413:
1414:     reports[selectedReportIndex] = {
1415:       ...currentReport,
1416:       familyGeneticImpactSection: {
1417:         ...currentSection,
1418:         impacts: impacts,
1419:       },
1420:     };
1421:
1422:     updated[selectedPatientIndex] = { ...patient, reports };
1423:     return updated;
1424:   });
1425: };
1426:
1427: const removeFamilyGeneticImpact = (index: number) => {
1428:   setPatients((prev) => {
1429:     const updated = [...prev];
1430:     const patient = { ...updated[selectedPatientIndex] };
1431:     const reports = [...patient.reports];
1432:     const currentReport = reports[selectedReportIndex];
1433:
1434:     // Get current section or create default
1435:     const currentSection = currentReport.familyGeneticImpactSection || {
1436:       description: "",
1437:       impacts: [],
1438:     };
1439:
1440:     reports[selectedReportIndex] = {
1441:       ...currentReport,
1442:       familyGeneticImpactSection: {
1443:         ...currentSection,
1444:         impacts: currentSection.impacts.filter((_, i) => i !== index),
1445:       },
1446:     };
1447:
1448:     updated[selectedPatientIndex] = { ...patient, reports };
1449:     return updated;
1450:   });
1451: };
1452:
1453: // You'll also need to add a function to update the description
1454: const updateFamilyGeneticImpactDescription = (description: string) => {
1455:   setPatients((prev) => {
1456:     const updated = [...prev];
1457:     const patient = { ...updated[selectedPatientIndex] };
1458:     const reports = [...patient.reports];

```

```

1459:     const currentReport = reports[selectedReportIndex];
1460:
1461:     // Get current section or create default
1462:     const currentSection = currentReport.familyGeneticImpactSection || {
1463:       description: "",
1464:       impacts: [],
1465:     };
1466:
1467:     reports[selectedReportIndex] = {
1468:       ...currentReport,
1469:       familyGeneticImpactSection: {
1470:         ...currentSection,
1471:         description: description,
1472:       },
1473:     };
1474:
1475:     updated[selectedPatientIndex] = { ...patient, reports };
1476:     return updated;
1477:   });
1478: };
1479:
1480: // Gene Test Results Functions
1481: const updateGeneTestResult = (
1482:   index: number,
1483:   field: keyof GeneTestResult,
1484:   value: string
1485: ) => {
1486:   setPatients((prev) => {
1487:     const updated = [...prev];
1488:     const patient = { ...updated[selectedPatientIndex] };
1489:     const reports = [...patient.reports];
1490:     const geneTests = [...reports[selectedReportIndex].geneTestResults];
1491:
1492:     geneTests[index] = { ...geneTests[index], [field]: value };
1493:
1494:     reports[selectedReportIndex] = {
1495:       ...reports[selectedReportIndex],
1496:       geneTestResults: geneTests,
1497:     };
1498:
1499:     updated[selectedPatientIndex] = { ...patient, reports };
1500:     return updated;
1501:   });
1502: };
1503:
1504: const addGeneTestResult = () => {
1505:   setPatients((prev) => {
1506:     const updated = [...prev];
1507:     const patient = { ...updated[selectedPatientIndex] };
1508:     const reports = [...patient.reports];
1509:     const geneTests = [...reports[selectedReportIndex].geneTestResults];
1510:
1511:     geneTests.push({ genecode: "", geneName: "", variation: "", result: "" });
1512:
1513:     reports[selectedReportIndex] = {
1514:       ...reports[selectedReportIndex],
1515:       geneTestResults: geneTests,
1516:     };
1517:
1518:     updated[selectedPatientIndex] = { ...patient, reports };
1519:     return updated;
1520:   });
1521: };
1522:
1523: const removeGeneTestResult = (index: number) => {
1524:   setPatients((prev) => {
1525:     const updated = [...prev];
1526:     const patient = { ...updated[selectedPatientIndex] };
1527:     const reports = [...patient.reports];
1528:
1529:     reports[selectedReportIndex] = {
1530:       ...reports[selectedReportIndex],
1531:       geneTestResults: reports[selectedReportIndex].geneTestResults.filter(

```

```

1532:         (_, i) => i !== index
1533:     ),
1534: };
1535:
1536:     updated[selectedPatientIndex] = { ...patient, reports };
1537:     return updated;
1538: });
1539: };
1540:
1541: // Genomic Analysis Table Update Functions
1542: const updateGenomicAnalysisTable = (newTable: GenomicAnalysisTable) => {
1543:     setPatients((prev) => {
1544:         const updated = [...prev];
1545:         const patient = { ...updated[selectedPatientIndex] };
1546:         const reports = [...patient.reports];
1547:
1548:         reports[selectedReportIndex] = {
1549:             ...reports[selectedReportIndex],
1550:             genomicAnalysisTable: newTable,
1551:         };
1552:
1553:         updated[selectedPatientIndex] = { ...patient, reports };
1554:         return updated;
1555:     });
1556: };
1557:
1558: const updateGenomicCategoryGroup = (
1559:     index: number,
1560:     newGroup: GenomicCategoryGroup
1561: ) => {
1562:     setPatients((prev) => {
1563:         const updated = [...prev];
1564:         const report = updated[selectedPatientIndex].reports[selectedReportIndex];
1565:         const genomicAnalysisTable = report.genomicAnalysisTable ?? {
1566:             description: "",
1567:             categories: [],
1568:         };
1569:         const newCategories = [...genomicAnalysisTable.categories];
1570:         newCategories[index] = newGroup;
1571:         updated[selectedPatientIndex].reports[selectedReportIndex] = {
1572:             ...report,
1573:             genomicAnalysisTable: {
1574:                 ...genomicAnalysisTable,
1575:                 categories: newCategories,
1576:             },
1577:         };
1578:         return updated;
1579:     });
1580: };
1581:
1582: // Show loading state
1583: if (isLoading) {
1584:     return (
1585:         <div className="min-h-screen flex items-center justify-center bg-gray-50">
1586:             <div className="text-center">
1587:                 <div className="inline-block animate-spin rounded-full h-12 w-12 border-b-2 border-blue-600 mb-4">
1588:                     <p className="text-gray-600 font-medium">Loading...</p>
1589:                 </div>
1590:             </div>
1591:         );
1592:     }
1593:
1594: if (!selectedPatient || !selectedReport) {
1595:     return (
1596:         <div className="min-h-screen flex items-center justify-center bg-gray-50">
1597:             <div className="text-center max-w-md mx-auto px-6">
1598:                 <div className="bg-white rounded-lg shadow-lg p-8">
1599:                     <div className="mx-auto w-16 h-16 bg-gray-100 rounded-full flex items-center justify-center mb-4">
1600:                         <svg
1601:                             className="w-8 h-8 text-gray-400"
1602:                             fill="none"
1603:                             stroke="currentColor"
1604:                             viewBox="0 0 24 24"

```



```

1605:         >
1606:         <path
1607:             strokeLinecap="round"
1608:             strokeLinejoin="round"
1609:             strokeWidth={2}
1610:             d="M9 12h6m-6 4h6m2 5H7a2 2 0 01-2-2V5a2 2 0 012-2h5.586a1 1 0 01.707.29315.414 5.414a1
1611:         />
1612:     </svg>
1613: </div>
1614: <h3 className="text-lg font-semibold text-gray-900 mb-2">
1615:     No Data Available
1616: </h3>
1617: <p className="text-gray-500">
1618:     No patient or report data found. Please select a patient and
1619:     report to continue.
1620: </p>
1621: </div>
1622: </div>
1623: </div>
1624: );
1625: }
1626:
1627: return (
1628:     <div className="max-w-8xl mx-auto py-6 sm:py-8 lg:py-10 px-4 sm:px-6 lg:px-8">
1629:         <div className="mb-6 sm:mb-8 no-print space-y-4">
1630:             { /* Patient & Report Selectors */ }
1631:             <div className="flex flex-wrap gap-4 items-end">
1632:                 { /* Patient Selector */ }
1633:                 <div className="flex flex-col gap-1 min-w-[160px]">
1634:                     <label className="text-sm font-medium text-gray-700">
1635:                         Select Patient
1636:                     </label>
1637:                     <div className="relative">
1638:                         <Select
1639:                             value={String(selectedPatientIndex)}
1640:                             onChange={(value) =>
1641:                                 setSelectedPatientIndex(Number(value))
1642:                             }
1643:                         >
1644:                             <SelectTrigger className="w-[160px]">
1645:                                 <SelectValue placeholder="Select Patient" />
1646:                             </SelectTrigger>
1647:                             <SelectContent>
1648:                                 {patients.map((p, i) => (
1649:                                     <SelectItem key={p.id} value={String(i)}>
1650:                                         {p.info.name || `Patient ${i + 1}`}
1651:                                     </SelectItem>
1652:                                 ))}
1653:                             </SelectContent>
1654:                         </Select>
1655:
1656:                         <div className="absolute inset-y-0 right-0 flex items-center pr-3 pointer-events-none">
1657:                             <svg
1658:                                 className="w-4 h-4 text-gray-400"
1659:                                 fill="none"
1660:                                 stroke="currentColor"
1661:                                 viewBox="0 0 24 24"
1662:                             >
1663:                                 <path
1664:                                     strokeLinecap="round"
1665:                                     strokeLinejoin="round"
1666:                                     strokeWidth={2}
1667:                                     d="M19 9l-7 7-7-7"
1668:                                 />
1669:                             </svg>
1670:                         </div>
1671:                     </div>
1672:                 </div>
1673:
1674:                 { /* Report Selector */ }
1675:                 <div className="flex flex-col gap-1 min-w-[160px]">
1676:                     <label className="text-sm font-medium text-gray-700">
1677:                         Select Report

```

```

1678:         </label>
1679:         <div className="relative">
1680:             <Select
1681:                 value={String(selectedReportIndex)}
1682:                 onChange={(value) => setSelectedReportIndex(Number(value))}
1683:             >
1684:                 <SelectTrigger className="w-[200px]">
1685:                     <SelectValue placeholder="Select Report" />
1686:                 </SelectTrigger>
1687:                 <SelectContent>
1688:                     {selectedPatient?.reports?.map((r, i) => (
1689:                         <SelectItem key={r.id} value={String(i)}>
1690:                             {r.name || `Report ${i + 1}`}
1691:                         </SelectItem>
1692:                     ))}
1693:                 </SelectContent>
1694:             </Select>
1695:             <div className="absolute inset-y-0 right-0 flex items-center pr-3 pointer-events-none">
1696:                 <svg
1697:                     className="w-4 h-4 text-gray-400"
1698:                     fill="none"
1699:                     stroke="currentColor"
1700:                     viewBox="0 0 24 24"
1701:                 >
1702:                     <path
1703:                         strokeLinecap="round"
1704:                         strokeLinejoin="round"
1705:                         strokeWidth={2}
1706:                         d="M19 9l-7 7-7-7"
1707:                     />
1708:                 </svg>
1709:             </div>
1710:         </div>
1711:     </div>
1712:
1713:     {/ * <div className="flex flex-col gap-1 min-w-[200px]">
1714:         <label className="text-sm font-medium text-gray-700">
1715:             Report Name
1716:         </label>
1717:         <Input
1718:             placeholder="Enter report name"
1719:             value={selectedReport?.name || ""}
1720:             onChange={(e) => updateReportName(e.target.value)}
1721:             className="w-[200px]"
1722:             title="Edit the name of the currently selected report"
1723:         />
1724:     </div> */}
1725:
1726:     {/ * Save Button */}
1727:     <div className="ml-auto">
1728:         <Button
1729:             onClick={saveReportData}
1730:             className="px-7 py-2.5"
1731:             title="Save all current patient and report data to storage"
1732:         >
1733:             Save Data
1734:         </Button>
1735:     </div>
1736: </div>
1737:
1738: {/ * Actions Section */}
1739: <div className="flex flex-wrap gap-3 items-center">
1740:     {/ * New Patient */}
1741:     <div className="flex gap-2 flex-1 min-w-[250px]">
1742:         <Input
1743:             placeholder="Sample Code (e.g. DNL0000000001)"
1744:             value={newSampleCode}
1745:             onChange={(e) => setNewSampleCode(e.target.value)}
1746:             className="flex-1 border-1 border-red-400"
1747:             title="Enter a unique sample code for the new patient"
1748:         />
1749:         <Button
1750:             onClick={() => addNewPatient(newSampleCode)}

```

```

1751:         className="whitespace-nowrap"
1752:         title="Create a new patient with the entered sample code"
1753:     >
1754:         Add Patient
1755:     </Button>
1756: </div>
1757:
1758: { /* Add Report */}
1759: <Button
1760:     onClick={handleAddReportClick}
1761:     className="whitespace-nowrap"
1762:     title="Add a new report to the currently selected patient"
1763: >
1764:     Add Report
1765: </Button>
1766:
1767: { /* Delete Buttons */}
1768: <Button
1769:     variant="destructive"
1770:     onClick={deleteSelectedReport}
1771:     disabled={({selectedPatient?.reports?.length ?? 0}) <= 1}
1772:     className="whitespace-nowrap"
1773:     title="Permanently delete the currently selected report"
1774: >
1775:     Delete Report
1776: </Button>
1777:
1778: <Button
1779:     variant="destructive"
1780:     onClick={deleteSelectedPatient}
1781:     className="whitespace-nowrap"
1782:     title="Permanently delete the currently selected patient and all their reports"
1783: >
1784:     Delete Patient
1785: </Button>
1786: </div>
1787:
1788: {showAddReportModal && (
1789:     <div className="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center z-50">
1790:         <div className="bg-white rounded-lg p-6 w-full max-w-md mx-4">
1791:             <h3 className="text-lg font-semibold mb-4">Add New Report</h3>
1792:             <div className="space-y-4">
1793:                 <div>
1794:                     <label className="block text-sm font-medium text-gray-700 mb-2">
1795:                         Report Name
1796:                     </label>
1797:                     <input
1798:                         placeholder="Enter report name (e.g., Initial Assessment, Follow-up, etc.)"
1799:                         value={newReportName}
1800:                         onChange={(e) => setNewReportName(e.target.value)}
1801:                         className="w-full"
1802:                         onKeyDown={(e) => {
1803:                             if (e.key === "Enter") {
1804:                                 handleAddReportSubmit();
1805:                             }
1806:                         }}
1807:                     />
1808:                 </div>
1809:                 <div className="flex gap-3 justify-end">
1810:                     <Button
1811:                         variant="outline"
1812:                         onClick={() => {
1813:                             setShowAddReportModal(false);
1814:                             setNewReportName("");
1815:                         }}
1816:                     >
1817:                         Cancel
1818:                     </Button>
1819:                     <Button onClick={handleAddReportSubmit}>Add Report</Button>
1820:                 </div>
1821:             </div>
1822:         </div>
1823:     </div>

```

```

1824:     })
1825: </div>
1826:
1827: <Tabs value={activeTab} onValueChange={setActiveTab} className="w-full">
1828:   <div className="w-full overflow-x-auto scrollbar-hide mb-4 sm:mb-6">
1829:     <TabList className="flex gap-1 sm:gap-2 min-w-max bg-white shadow-sm px-2 sm:px-4 py-2 rounded-t-
1830:       <TabTrigger
1831:         value="settings"
1832:         className="text-xs sm:text-sm whitespace-nowrap"
1833:       >
1834:         Front Page
1835:       </TabTrigger>
1836:       <TabTrigger
1837:         value="patient-info"
1838:         className="text-xs sm:text-sm whitespace-nowrap"
1839:       >
1840:         Patient Info
1841:       </TabTrigger>
1842:       <TabTrigger
1843:         value="content"
1844:         className="text-xs sm:text-sm whitespace-nowrap"
1845:       >
1846:         Content
1847:       </TabTrigger>
1848:       <TabTrigger
1849:         value="genetic-parameters"
1850:         className="text-xs sm:text-sm whitespace-nowrap"
1851:       >
1852:         Table Of Content
1853:       </TabTrigger>
1854:
1855:       <TabTrigger
1856:         value="health-summary"
1857:         className="text-xs sm:text-sm whitespace-nowrap"
1858:       >
1859:         Health Summary
1860:       </TabTrigger>
1861:
1862:       <TabTrigger
1863:         value="diet-fields"
1864:         className="text-xs sm:text-sm whitespace-nowrap"
1865:       >
1866:         Diet Fields
1867:       </TabTrigger>
1868:       <TabTrigger
1869:         value="nutrition"
1870:         className="text-xs sm:text-sm whitespace-nowrap"
1871:       >
1872:         Nutrition
1873:       </TabTrigger>
1874:       <TabTrigger
1875:         value="sports-fitness"
1876:         className="text-xs sm:text-sm whitespace-nowrap"
1877:       >
1878:         Sports & Fitness
1879:       </TabTrigger>
1880:       <TabTrigger
1881:         value="lifestyle-conditions"
1882:         className="text-xs sm:text-sm whitespace-nowrap"
1883:       >
1884:         Lifestyle
1885:       </TabTrigger>
1886:       <TabTrigger
1887:         value="metabolic-core"
1888:         className="text-xs sm:text-sm whitespace-nowrap"
1889:       >
1890:         Metabolic Core
1891:       </TabTrigger>
1892:       <TabTrigger
1893:         value="digestive-health"
1894:         className="text-xs sm:text-sm whitespace-nowrap"
1895:       >
1896:         Digestive Health

```

```

1897:         </TabsTrigger>
1898:     <TabsTrigger
1899:         value="genes-addiction"
1900:         className="text-xs sm:text-sm whitespace-nowrap"
1901:     >
1902:         Genes & Addiction
1903:     </TabsTrigger>
1904:     <TabsTrigger
1905:         value="sleep-rest"
1906:         className="text-xs sm:text-sm whitespace-nowrap"
1907:     >
1908:         Sleep & Rest
1909:     </TabsTrigger>
1910:     <TabsTrigger
1911:         value="allergies-sensitivity"
1912:         className="text-xs sm:text-sm whitespace-nowrap"
1913:     >
1914:         Allergies
1915:     </TabsTrigger>
1916:     <TabsTrigger
1917:         value="preventive-health"
1918:         className="text-xs sm:text-sm whitespace-nowrap"
1919:     >
1920:         Preventive Health
1921:     </TabsTrigger>
1922:     <TabsTrigger
1923:         value="family-genetic-impact"
1924:         className="text-xs sm:text-sm whitespace-nowrap"
1925:     >
1926:         Family Impact
1927:     </TabsTrigger>
1928:     <TabsTrigger
1929:         value="genomic-analysis"
1930:         className="text-xs sm:text-sm whitespace-nowrap"
1931:     >
1932:         Genomic Analysis
1933:     </TabsTrigger>
1934:     <TabsTrigger
1935:         value="genes"
1936:         className="text-xs sm:text-sm whitespace-nowrap"
1937:     >
1938:         Gene Results
1939:     </TabsTrigger>
1940:
1941:     <TabsTrigger
1942:         value="summaries"
1943:         className="text-xs sm:text-sm whitespace-nowrap"
1944:     >
1945:         Summaries
1946:     </TabsTrigger>
1947:
1948:     <TabsTrigger
1949:         value="pdf-export"
1950:         className="text-xs sm:text-sm whitespace-nowrap"
1951:     >
1952:         PDF Export
1953:     </TabsTrigger>
1954:     <TabsTrigger
1955:         value="import-export"
1956:         className="text-xs sm:text-sm whitespace-nowrap"
1957:     >
1958:         Import/Export
1959:     </TabsTrigger>
1960:
1961:     <TabsTrigger
1962:         value="share"
1963:         className="text-xs sm:text-sm whitespace-nowrap"
1964:     >
1965:         Share Links
1966:     </TabsTrigger>
1967:
1968:     <TabsTrigger value="preview">Preview</TabsTrigger>
1969: </TabsList>

```

```

1970:     </div>
1971:
1972:     <div className="w-full overflow-x-auto">
1973:       <TabsContent className="w-full min-w-0" value="patient-info">
1974:         <PatientInfoAdmin
1975:           patientInfo={selectedPatient?.info}
1976:           updatePatientInfo={updatePatientInfo}
1977:           onSave={saveReportData}
1978:           onReset={resetPatientInfo}
1979:         />
1980:       </TabsContent>
1981:
1982:       <TabsContent className="w-full min-w-0" value="content">
1983:         <ContentAdmin
1984:           content={selectedReport.content}
1985:           updateContent={updateReportContent}
1986:           onSave={saveReportData}
1987:           onReset={() => resetSection("content")}
1988:         />
1989:       </TabsContent>
1990:
1991:       <TabsContent className="w-full min-w-0" value="genetic-parameters">
1992:         <GeneticParametersAdmin
1993:           categories={selectedReport.categories}
1994:           onUpdateCategories={(updated) => {
1995:             setPatients((prev) => {
1996:               const updatedPatients = [...prev];
1997:               const patient = { ...updatedPatients[selectedPatientIndex] };
1998:               const reports = [...patient.reports];
1999:               const report = {
2000:                 ...reports[selectedReportIndex],
2001:                 categories: updated,
2002:               };
2003:               reports[selectedReportIndex] = report;
2004:               updatedPatients[selectedPatientIndex] = {
2005:                 ...patient,
2006:                 reports,
2007:               };
2008:               return updatedPatients;
2009:             });
2010:           }}
2011:           onSave={saveReportData}
2012:           onReset={() => resetSection("categories")}
2013:         />
2014:       </TabsContent>
2015:
2016:       <TabsContent className="w-full min-w-0" value="settings">
2017:         <SettingsAdmin
2018:           settings={selectedReport.settings}
2019:           updateSettings={updateReportSettings}
2020:           onSave={saveReportData}
2021:           onReset={() => resetSection("settings")}
2022:         />
2023:       </TabsContent>
2024:
2025:       <TabsContent value="health-summary">
2026:         <HealthSummaryAdmin
2027:           healthSummary={
2028:             selectedReport.healthSummary || {
2029:               description: "",
2030:               data: [],
2031:             }
2032:           }
2033:           onDescriptionUpdate={(value) =>
2034:             updateHealthSummaryDescription(
2035:               selectedPatientIndex,
2036:               selectedReportIndex,
2037:               value
2038:             )
2039:           }
2040:           onUpdate={(entryIndex, field, value) =>
2041:             updateHealthSummaryEntry(
2042:               selectedPatientIndex,

```

```

2043:         selectedReportIndex,
2044:         entryIndex,
2045:         field,
2046:         value
2047:     )
2048: }
2049: onAdd={() =>
2050:     addHealthSummaryEntry(selectedPatientIndex, selectedReportIndex)
2051: }
2052: onRemove={(entryIndex) =>
2053:     removeHealthSummaryEntry(
2054:         selectedPatientIndex,
2055:         selectedReportIndex,
2056:         entryIndex
2057:     )
2058: }
2059: onSave={saveReportData}
2060: isLoading={isLoading}
2061: />
2062: </TabsContent>
2063:
2064: <TabsContent className="w-full min-w-0" value="diet-fields">
2065:     <DynamicDietFieldAdmin
2066:         dynamicDietFieldDefinitions={
2067:             selectedReport.dynamicDietFieldDefinitions
2068:         }
2069:         patientDietAnalysisResults={
2070:             selectedReport.patientDietAnalysisResults
2071:         }
2072:         categories={selectedReport.dietFieldCategories}
2073:         onUpdateCategories={(newCats) => {
2074:             setPatients((prev) => {
2075:                 const updated = [...prev];
2076:                 const patient = { ...updated[selectedPatientIndex] };
2077:                 const reports = [...patient.reports];
2078:                 const report = {
2079:                     ...reports[selectedReportIndex],
2080:                     dietFieldCategories: newCats,
2081:                 };
2082:                 reports[selectedReportIndex] = report;
2083:                 updated[selectedPatientIndex] = { ...patient, reports };
2084:                 return updated;
2085:             });
2086:         }}
2087:         onUpdateFields={(fields) => {
2088:             setPatients((prev) => {
2089:                 const updated = [...prev];
2090:                 const patient = { ...updated[selectedPatientIndex] };
2091:                 const reports = [...patient.reports];
2092:                 const report = {
2093:                     ...reports[selectedReportIndex],
2094:                     dynamicDietFieldDefinitions: fields,
2095:                 };
2096:                 reports[selectedReportIndex] = report;
2097:                 updated[selectedPatientIndex] = { ...patient, reports };
2098:                 return updated;
2099:             });
2100:         }}
2101:         onUpdateResults={(results) => {
2102:             setPatients((prev) => {
2103:                 const updated = [...prev];
2104:                 const patient = { ...updated[selectedPatientIndex] };
2105:                 const reports = [...patient.reports];
2106:                 const report = {
2107:                     ...reports[selectedReportIndex],
2108:                     patientDietAnalysisResults: results,
2109:                 };
2110:                 reports[selectedReportIndex] = report;
2111:                 updated[selectedPatientIndex] = { ...patient, reports };
2112:                 return updated;
2113:             });
2114:         }}
2115:         onSave={saveReportData}

```

```

2116:         onReset={() => resetSection("dynamicDietFieldDefinitions")}
2117:     />
2118: </TabsContent>
2119:
2120: <TabsContent className="w-full min-w-0" value="nutrition">
2121:     <NutritionAdmin
2122:         nutritionData={selectedReport.nutritionData}
2123:         updateNutritionData={updateNutritionData}
2124:         onSave={saveReportData}
2125:     />
2126: </TabsContent>
2127:
2128: <TabsContent className="w-full min-w-0" value="sports-fitness">
2129:     <SportsFitnessAdmin
2130:         sportsAndFitness={selectedReport.sportsAndFitness}
2131:         updateSportsAndFitness={updateSportsAndFitness}
2132:         onSave={saveReportData}
2133:         onReset={() => resetSection("sportsAndFitness")}
2134:     />
2135: </TabsContent>
2136:
2137: <TabsContent className="w-full min-w-0" value="lifestyle-conditions">
2138:     <LifestyleConditionsAdmin
2139:         lifestyleConditions={selectedReport.lifestyleConditions}
2140:         lifestyleCategoryImages={
2141:             selectedReport.lifestyleCategoryImages ?? {}
2142:         }
2143:         addCategory={(categoryId: string) => {
2144:             setPatients((prev) => {
2145:                 const updated = [...prev];
2146:                 const patient = { ...updated[selectedPatientIndex] };
2147:                 const reports = [...patient.reports];
2148:                 const report = { ...reports[selectedReportIndex] };
2149:
2150:                 const existing = report.lifestyleConditions?.[categoryId];
2151:                 if (typeof existing === "object" && existing !== null)
2152:                     return prev;
2153:
2154:                 report.lifestyleConditions = {
2155:                     ...report.lifestyleConditions,
2156:                     [categoryId]: {},
2157:                 };
2158:
2159:                 reports[selectedReportIndex] = report;
2160:                 updated[selectedPatientIndex] = { ...patient, reports };
2161:                 return updated;
2162:             });
2163:         }}
2164:         addField={(categoryId, fieldName, status) => {
2165:             setPatients((prev) => {
2166:                 const updated = [...prev];
2167:                 const patient = { ...updated[selectedPatientIndex] };
2168:                 const reports = [...patient.reports];
2169:                 const report = { ...reports[selectedReportIndex] };
2170:                 const category = report.lifestyleConditions?.[categoryId];
2171:
2172:                 if (typeof category !== "object" || category === null)
2173:                     return prev;
2174:
2175:                 report.lifestyleConditions = {
2176:                     ...report.lifestyleConditions,
2177:                     [categoryId]: {
2178:                         ...category,
2179:                         [fieldName]: {
2180:                             status,
2181:                             description: "",
2182:                             sensitivity: "low",
2183:                             avoid: [],
2184:                             follow: [],
2185:                             consume: [],
2186:                             monitor: [],
2187:                             avoidLabel: "AVOID",
2188:                             followLabel: "FOLLOW",

```



```

2189:             consumeLabel: "CONSUME",
2190:             monitorLabel: "MONITOR",
2191:         },
2192:     },
2193: };
2194:
2195:     reports[selectedReportIndex] = report;
2196:     updated[selectedPatientIndex] = { ...patient, reports };
2197:     return updated;
2198: });
2199: }}
2200: updateFieldStatus={updateFieldStatus} // already refactored previously
2201: deleteField={(categoryId, fieldName) => {
2202:     setPatients((prev) => {
2203:         const updated = [...prev];
2204:         const patient = { ...updated[selectedPatientIndex] };
2205:         const reports = [...patient.reports];
2206:         const report = { ...reports[selectedReportIndex] };
2207:         const category = report.lifestyleConditions?.[categoryId];
2208:
2209:         if (typeof category !== "object" || category === null)
2210:             return prev;
2211:
2212:         const { [fieldName]: _, ...restFields } = category;
2213:
2214:         report.lifestyleConditions = {
2215:             ...report.lifestyleConditions,
2216:             [categoryId]: restFields,
2217:         };
2218:
2219:         reports[selectedReportIndex] = report;
2220:         updated[selectedPatientIndex] = { ...patient, reports };
2221:         return updated;
2222:     });
2223: }}
2224: deleteCategory={(categoryId: string) => {
2225:     setPatients((prev) => {
2226:         const updated = [...prev];
2227:         const patient = { ...updated[selectedPatientIndex] };
2228:         const reports = [...patient.reports];
2229:         const report = { ...reports[selectedReportIndex] };
2230:
2231:         const { [categoryId]: _, ...restConditions } =
2232:             report.lifestyleConditions;
2233:         const { [categoryId]: __, ...restImages } =
2234:             report.lifestyleCategoryImages ?? {};
2235:
2236:         report.lifestyleConditions = restConditions;
2237:         report.lifestyleCategoryImages = restImages;
2238:
2239:         reports[selectedReportIndex] = report;
2240:         updated[selectedPatientIndex] = { ...patient, reports };
2241:         return updated;
2242:     });
2243: }}
2244: updateQuoteAndDescription=({
2245:     quote: string,
2246:     description: string
2247: }) => {
2248:     setPatients((prev) => {
2249:         const updated = [...prev];
2250:         const patient = { ...updated[selectedPatientIndex] };
2251:         const reports = [...patient.reports];
2252:         const report = {
2253:             ...reports[selectedReportIndex],
2254:             lifestyleConditions: {
2255:                 ...reports[selectedReportIndex].lifestyleConditions,
2256:                 quote,
2257:                 description,
2258:             },
2259:         };
2260:         reports[selectedReportIndex] = report;
2261:         updated[selectedPatientIndex] = { ...patient, reports };

```

```

2262:         return updated;
2263:     });
2264: }
2265: updateCategoryImage={(categoryId: string, imageUrl: string) => {
2266:     setPatients((prev) => {
2267:         const updated = [...prev];
2268:         const patient = { ...updated[selectedPatientIndex] };
2269:         const reports = [...patient.reports];
2270:         const report = {
2271:             ...reports[selectedReportIndex],
2272:             lifestyleCategoryImages: {
2273:                 ...(reports[selectedReportIndex]
2274:                     .lifestyleCategoryImages ?? {}),
2275:                 [categoryId]: imageUrl,
2276:             },
2277:         };
2278:         reports[selectedReportIndex] = report;
2279:         updated[selectedPatientIndex] = { ...patient, reports };
2280:         return updated;
2281:     });
2282: }}
2283: onSave={saveReportData}
2284: onReset={() => resetSection("lifestyleConditions")}
2285: />
2286: </TabsContent>
2287:
2288: <TabsContent className="w-full min-w-0" value="metabolic-core">
2289:     <MetabolicCoreAdmin
2290:         metabolicCore={selectedReport.metabolicCore}
2291:         setMetabolicCore={(valueOrUpdater) => {
2292:             setPatients((prev) => {
2293:                 const updated = [...prev];
2294:                 const patient = { ...updated[selectedPatientIndex] };
2295:                 const reports = [...patient.reports];
2296:
2297:                 const prevMetabolicCore =
2298:                     reports[selectedReportIndex].metabolicCore;
2299:                 const nextMetabolicCore =
2300:                     typeof valueOrUpdater === "function"
2301:                     ? (
2302:                         valueOrUpdater as (
2303:                             prev: MetabolicCore
2304:                         ) => MetabolicCore
2305:                     )(prevMetabolicCore)
2306:                     : valueOrUpdater;
2307:
2308:                 reports[selectedReportIndex] = {
2309:                     ...reports[selectedReportIndex],
2310:                     metabolicCore: nextMetabolicCore,
2311:                 };
2312:
2313:                 updated[selectedPatientIndex] = { ...patient, reports };
2314:                 return updated;
2315:             });
2316:         }}
2317:         updateMetabolicCore={updateMetabolicCore}
2318:         onSave={saveReportData}
2319:         onReset={() => resetSection("metabolicCore")}
2320:     />
2321: </TabsContent>
2322:
2323: <TabsContent className="w-full min-w-0" value="digestive-health">
2324:     <DigestiveHealthAdmin
2325:         digestiveHealth={selectedReport.digestiveHealth}
2326:         updateField={updateDigestiveHealth}
2327:         addField={addDigestiveField}
2328:         deleteField={deleteDigestiveField}
2329:         updateQuoteAndDescription={updateDigestiveQuoteAndDescription}
2330:         onSave={saveReportData}
2331:         onReset={() => resetSection("digestiveHealth")}
2332:     />
2333: </TabsContent>
2334:

```

```

2335: <TabsContent className="w-full min-w-0" value="genes-addiction">
2336:   <GenesAddictionAdmin
2337:     data={selectedReport.genesAndAddiction}
2338:     updateData={({updated}) => {
2339:       setPatients((prev) => {
2340:         const updatedPatients = [...prev];
2341:         const patient = { ...updatedPatients[selectedPatientIndex] };
2342:         const reports = [...patient.reports];
2343:
2344:         reports[selectedReportIndex] = {
2345:           ...reports[selectedReportIndex],
2346:           genesAndAddiction: {
2347:             ...reports[selectedReportIndex].genesAndAddiction,
2348:             ...updated,
2349:           },
2350:         };
2351:
2352:         updatedPatients[selectedPatientIndex] = {
2353:           ...patient,
2354:           reports,
2355:         };
2356:         return updatedPatients;
2357:       });
2358:     }}
2359:     onSave={saveReportData}
2360:     onReset={() => resetSection("genesAndAddiction")}
2361:   />
2362: </TabsContent>
2363:
2364: <TabsContent className="w-full min-w-0" value="sleep-rest">
2365:   <SleepRestAdmin
2366:     data={selectedReport.sleepAndRest}
2367:     updateData={({update}) => {
2368:       setPatients((prev) => {
2369:         const updatedPatients = [...prev];
2370:         const patient = { ...updatedPatients[selectedPatientIndex] };
2371:         const reports = [...patient.reports];
2372:
2373:         reports[selectedReportIndex] = {
2374:           ...reports[selectedReportIndex],
2375:           sleepAndRest: {
2376:             ...reports[selectedReportIndex].sleepAndRest,
2377:             ...update,
2378:           },
2379:         };
2380:
2381:         updatedPatients[selectedPatientIndex] = {
2382:           ...patient,
2383:           reports,
2384:         };
2385:         return updatedPatients;
2386:       });
2387:     }}
2388:     onSave={saveReportData}
2389:     onReset={() => resetSection("sleepAndRest")}
2390:   />
2391: </TabsContent>
2392:
2393: <TabsContent className="w-full min-w-0" value="allergies-sensitivity">
2394:   <AllergiesSensitivityAdmin
2395:     allergiesAndSensitivity={
2396:       selectedReport.allergiesAndSensitivity || {
2397:         quote: "",
2398:         description: "",
2399:         generalAdvice: "",
2400:         data: {},
2401:       }
2402:     }
2403:     updateAllergiesAndSensitivity={updateAllergiesAndSensitivity}
2404:     onSave={saveReportData}
2405:     onReset={() => resetSection("allergiesAndSensitivity")}
2406:   />
2407: </TabsContent>

```

```

2408:
2409: <TabsContent className="w-full min-w-0" value="preventive-health">
2410:   <PreventiveHealthAdmin
2411:     preventiveHealth={selectedReport.preventiveHealth}
2412:     updatePreventiveHealth={updatePreventiveHealth}
2413:     onSave={saveReportData}
2414:     onReset={() => resetSection("preventiveHealth")}
2415:   />
2416: </TabsContent>
2417:
2418: <TabsContent className="w-full min-w-0" value="family-genetic-impact">
2419:   <FamilyGeneticImpactAdmin
2420:     familyGeneticImpactSection={
2421:       selectedReport.familyGeneticImpactSection || {
2422:         description: "",
2423:         impacts: [],
2424:       }
2425:     }
2426:     setFamilyGeneticImpactSection={(section) => {
2427:       setPatients((prev) => {
2428:         const updated = [...prev];
2429:         const patient = { ...updated[selectedPatientIndex] };
2430:         const reports = [...patient.reports];
2431:
2432:         reports[selectedReportIndex] = {
2433:           ...reports[selectedReportIndex],
2434:           familyGeneticImpactSection: section,
2435:         };
2436:
2437:         updated[selectedPatientIndex] = { ...patient, reports };
2438:         return updated;
2439:       });
2440:     }}
2441:     onSave={saveReportData}
2442:     onReset={() => resetSection("familyGeneticImpactSection")}
2443:   />
2444: </TabsContent>
2445:
2446: <TabsContent className="w-full min-w-0" value="genomic-analysis">
2447:   <GenomicAnalysisAdmin
2448:     genomicAnalysisTable={
2449:       selectedReport.genomicAnalysisTable || {
2450:         description: "",
2451:         categories: [],
2452:       }
2453:     }
2454:     setGenomicAnalysisTable={updateGenomicAnalysisTable}
2455:     onSave={saveReportData}
2456:     onReset={() => resetSection("genomicAnalysisTable")}
2457:   />
2458: </TabsContent>
2459:
2460: <TabsContent className="w-full min-w-0" value="genes">
2461:   <GeneResultsAdmin
2462:     geneTestResults={selectedReport.geneTestResults}
2463:     updateGeneTestResult={updateGeneTestResult}
2464:     addGeneTestResult={addGeneTestResult}
2465:     removeGeneTestResult={removeGeneTestResult}
2466:     onSave={saveReportData}
2467:     onReset={() => resetSection("geneTestResults")}
2468:   />
2469: </TabsContent>
2470:
2471: <TabsContent className="w-full min-w-0" value="pdf-export">
2472:   {selectedPatient && selectedReport && (
2473:     <PDFGenerator
2474:       reportData={{
2475:         ...selectedReport,
2476:         patientInfo: selectedPatient.info,
2477:       }}
2478:       reportIndex={selectedReportIndex}
2479:       patientCode={selectedPatient.info.sampleCode}
2480:     />

```

```

2481:         })
2482:     </TabsContent>
2483:
2484:     <TabsContent className="w-full min-w-0" value="import-export">
2485:         <ImportExportAdmin
2486:             reportData={{
2487:                 ...selectedReport,
2488:                 patientInfo: selectedPatient.info,
2489:             }}
2490:             setReportData={(updatedReport: ComprehensiveReportData) => {
2491:                 setPatients((prev) => {
2492:                     const updated = [...prev];
2493:                     const patient = { ...updated[selectedPatientIndex] };
2494:                     const reports = [...patient.reports];
2495:
2496:                     // Destructure into patientInfo and the rest of the report
2497:                     const { patientInfo, ...reportRest } = updatedReport;
2498:
2499:                     reports[selectedReportIndex] = reportRest; // already typed as Report
2500:
2501:                     updated[selectedPatientIndex] = {
2502:                         ...patient,
2503:                         reports,
2504:                         info: patientInfo,
2505:                     };
2506:
2507:                     return updated;
2508:                 });
2509:             }}
2510:         />
2511:     </TabsContent>
2512:
2513:     <TabsContent className="w-full min-w-0" value="preview">
2514:         {selectedReport && selectedPatient ? (
2515:             <ReportPreview
2516:                 selectedReport={selectedReport}
2517:                 selectedPatient={selectedPatient}
2518:             />
2519:         ) : (
2520:             <div className="text-center text-gray-500 italic py-10">
2521:                 No patient or report selected for preview.
2522:             </div>
2523:         )}
2524:     </TabsContent>
2525:
2526:     <TabsContent className="w-full min-w-0" value="share">
2527:         <ShareManagement
2528:             reportId={selectedReport.id}
2529:             patientId={selectedPatient.id}
2530:             reportName={selectedReport.name}
2531:         />
2532:     </TabsContent>
2533:
2534: </div>
2535: </Tabs>
2536: </div>
2537: );
2538: };
2539:
2540: export default AdminPage;

```

## ■ File: app\api\addiction-images\route.ts

```

=====
1: import { NextRequest, NextResponse } from "next/server";
2: import { writeFile, mkdir, readdir } from "fs/promises";
3: import path from "path";
4: import { existsSync } from "fs";
5:
6: // Interface for image objects
7: interface AddictionImage {
8:     label: string;

```

```

9:   url: string;
10: }
11:
12: // GET handler - Fetch all addiction images
13: export async function GET() {
14:   try {
15:     const publicPath = path.join(process.cwd(), "public");
16:     const iconsPath = path.join(publicPath, "addiction", "icons");
17:     const sensitivityPath = path.join(publicPath, "addiction", "sensitivity");
18:
19:     // Initialize arrays for images
20:     const icons: AddictionImage[] = [];
21:     const sensitivities: AddictionImage[] = [];
22:
23:     // Read icons directory
24:     if (existsSync(iconsPath)) {
25:       try {
26:         const iconFiles = await readdir(iconsPath);
27:         for (const file of iconFiles) {
28:           if (file.match(/\.(jpg|jpeg|png|gif|svg|webp)$/i)) {
29:             const label = path.parse(file).name; // Remove extension
30:             icons.push({
31:               label,
32:               url: `/addiction/icons/${file}`,
33:             });
34:           }
35:         }
36:       } catch (error) {
37:         console.warn("Could not read icons directory:", error);
38:       }
39:     }
40:
41:     // Read sensitivity directory
42:     if (existsSync(sensitivityPath)) {
43:       try {
44:         const sensitivityFiles = await readdir(sensitivityPath);
45:         for (const file of sensitivityFiles) {
46:           if (file.match(/\.(jpg|jpeg|png|gif|svg|webp)$/i)) {
47:             const label = path.parse(file).name; // Remove extension
48:             sensitivities.push({
49:               label,
50:               url: `/addiction/sensitivity/${file}`,
51:             });
52:           }
53:         }
54:       } catch (error) {
55:         console.warn("Could not read sensitivity directory:", error);
56:       }
57:     }
58:
59:     return NextResponse.json({
60:       success: true,
61:       icons,
62:       sensitivities,
63:     });
64:   } catch (error) {
65:     console.error("Error fetching addiction images:", error);
66:     return NextResponse.json(
67:       {
68:         success: false,
69:         error: "Failed to fetch images",
70:       },
71:       { status: 500 }
72:     );
73:   }
74: }
75:
76: // POST handler - Upload new addiction images
77: export async function POST(request: NextRequest) {
78:   try {
79:     const formData = await request.formData();
80:     const file = formData.get("file") as File;
81:     const label = formData.get("label") as string;

```

```

82:     const folder = formData.get("folder") as string; // "addiction/icons" or "addiction/sensitivity"
83:
84:     // Validation
85:     if (!file) {
86:         return NextResponse.json(
87:             { success: false, error: "No file provided" },
88:             { status: 400 }
89:         );
90:     }
91:
92:     if (!label) {
93:         return NextResponse.json(
94:             { success: false, error: "No label provided" },
95:             { status: 400 }
96:         );
97:     }
98:
99:     if (
100:         !folder ||
101:         (!folder.includes("addiction/icons") &&
102:         !folder.includes("addiction/sensitivity"))
103:     ) {
104:         return NextResponse.json(
105:             { success: false, error: "Invalid folder specified" },
106:             { status: 400 }
107:         );
108:     }
109:
110:     // Validate file type
111:     const allowedTypes = [
112:         "image/jpeg",
113:         "image/jpg",
114:         "image/png",
115:         "image/gif",
116:         "image/svg+xml",
117:         "image/webp",
118:     ];
119:     if (!allowedTypes.includes(file.type)) {
120:         return NextResponse.json(
121:             {
122:                 success: false,
123:                 error: "Invalid file type. Please upload an image file.",
124:             },
125:             { status: 400 }
126:         );
127:     }
128:
129:     // Create file extension based on MIME type
130:     const getFileExtension = (mimeType: string): string => {
131:         const extensions: Record<string, string> = {
132:             "image/jpeg": ".jpg",
133:             "image/jpg": ".jpg",
134:             "image/png": ".png",
135:             "image/gif": ".gif",
136:             "image/svg+xml": ".svg",
137:             "image/webp": ".webp",
138:         };
139:         return extensions[mimeType] || ".jpg";
140:     };
141:
142:     const fileExtension = getFileExtension(file.type);
143:     const cleanLabel = label.toLowerCase().replace(/[^\a-z0-9_-]/g, "_");
144:     const fileName = `${cleanLabel}${fileExtension}`;
145:
146:     // Create directory path
147:     const publicPath = path.join(process.cwd(), "public");
148:     const uploadDir = path.join(publicPath, folder);
149:     const filePath = path.join(uploadDir, fileName);
150:
151:     // Ensure directory exists
152:     try {
153:         await mkdir(uploadDir, { recursive: true });
154:     } catch (error) {

```

```

155:     console.error("Error creating directory:", error);
156: }
157:
158: // Convert file to buffer and save
159: const bytes = await file.arrayBuffer();
160: const buffer = Buffer.from(bytes);
161: await writeFile(filePath, buffer);
162:
163: // Generate the public URL
164: const publicUrl = `/${folder}/${fileName}`;
165:
166: return NextResponse.json({
167:     success: true,
168:     message: "File uploaded successfully",
169:     label: cleanLabel,
170:     url: publicUrl,
171:     filename: fileName,
172: });
173: } catch (error) {
174:     console.error("Error uploading file:", error);
175:     return NextResponse.json(
176:         {
177:             success: false,
178:             error: "Failed to upload file",
179:         },
180:         { status: 500 }
181:     );
182: }
183: }
184:
185: // DELETE handler - Optional: Delete addiction images
186: export async function DELETE(request: NextRequest) {
187:     try {
188:         const { searchParams } = new URL(request.url);
189:         const fileName = searchParams.get("file");
190:         const folder = searchParams.get("folder");
191:
192:         if (!fileName || !folder) {
193:             return NextResponse.json(
194:                 { success: false, error: "Missing file or folder parameter" },
195:                 { status: 400 }
196:             );
197:         }
198:
199:         const publicPath = path.join(process.cwd(), "public");
200:         const filePath = path.join(publicPath, folder, fileName);
201:
202:         // Check if file exists and delete it
203:         if (existsSync(filePath)) {
204:             const fs = await import("fs/promises");
205:             await fs.unlink(filePath);
206:
207:             return NextResponse.json({
208:                 success: true,
209:                 message: "File deleted successfully",
210:             });
211:         } else {
212:             return NextResponse.json(
213:                 { success: false, error: "File not found" },
214:                 { status: 404 }
215:             );
216:         }
217:     } catch (error) {
218:         console.error("Error deleting file:", error);
219:         return NextResponse.json(
220:             {
221:                 success: false,
222:                 error: "Failed to delete file",
223:             },
224:             { status: 500 }
225:         );
226:     }
227: }

```



## ■ File: app\api\allergies-image\route.ts

```
=====
1: import { promises as fs } from "fs";
2: import path from "path";
3: import { NextRequest, NextResponse } from "next/server";
4:
5: const folderName = "allergies";
6: const folderPath = path.join(process.cwd(), "public", folderName);
7:
8: // === GET: Return all allergy image URLs ===
9: export async function GET() {
10:   try {
11:     const files = await fs.readdir(folderPath);
12:     const images = files.map((file) => ({
13:       label: path.parse(file).name,
14:       url: `/${folderName}/${file}`,
15:     }));
16:     return NextResponse.json({ success: true, images });
17:   } catch (err) {
18:     console.error("? Failed to read allergy image folder:", err);
19:     return NextResponse.json(
20:       { success: false, error: "Could not read images" },
21:       { status: 500 }
22:     );
23:   }
24: }
25:
26: // === POST: Upload a new allergy image ===
27: export async function POST(req: NextRequest) {
28:   try {
29:     const formData = await req.formData();
30:     const file = formData.get("file") as File;
31:     const label = formData.get("label") as string;
32:
33:     if (!file || !label) {
34:       return NextResponse.json(
35:         { success: false, error: "Missing file or label" },
36:         { status: 400 }
37:       );
38:     }
39:
40:     const ext = file.name.split(".").pop() || "png";
41:     const safeLabel = label
42:       .toLowerCase()
43:       .replace(/\s+/g, "_")
44:       .replace(/[^\a-z0-9_]/gi, "");
45:     const fileName = `${safeLabel}.${ext}`;
46:     const filePath = path.join(folderPath, fileName);
47:
48:     await fs.mkdir(folderPath, { recursive: true });
49:     const buffer = Buffer.from(await file.arrayBuffer());
50:     await fs.writeFile(filePath, buffer);
51:
52:     return NextResponse.json({
53:       success: true,
54:       label: safeLabel,
55:       url: `/${folderName}/${fileName}`,
56:     });
57:   } catch (err) {
58:     console.error("? Upload failed:", err);
59:     return NextResponse.json(
60:       { success: false, error: "Upload failed" },
61:       { status: 500 }
62:     );
63:   }
64: }
65:
66: // === DELETE: Remove an allergy image by filename ===
67: export async function DELETE(req: NextRequest) {
68:   try {
69:     const { filename } = await req.json();
70:
71:     if (!filename) {
```

```

72:         return NextResponse.json(
73:             { success: false, error: "Missing filename" },
74:             { status: 400 }
75:         );
76:     }
77:
78:     const filePath = path.join(folderPath, filename);
79:
80:     // Double-check file exists before deleting
81:     await fs.access(filePath);
82:     await fs.unlink(filePath);
83:
84:     return NextResponse.json({ success: true, deleted: filename });
85: } catch (err: any) {
86:     console.error("? Delete failed:", err.message);
87:     return NextResponse.json(
88:         { success: false, error: "Delete failed" },
89:         { status: 500 }
90:     );
91: }
92: }

```

## ■ File: app\api\comprehensive-report-data\route.ts

```

1: import { type NextRequest, NextResponse } from "next/server"
2:
3: // In-memory storage for demo purposes
4: let reportData: any = null
5:
6: export async function GET() {
7:     if (!reportData) {
8:         return NextResponse.json({ message: "No data found" }, { status: 404 })
9:     }
10:
11:     return NextResponse.json(reportData)
12: }
13:
14: export async function POST(request: NextRequest) {
15:     try {
16:         const data = await request.json()
17:         reportData = data
18:
19:         return NextResponse.json({ message: "Data saved successfully" })
20:     } catch (error) {
21:         return NextResponse.json({ message: "Error saving data" }, { status: 500 })
22:     }
23: }

```

## ■ File: app\api\digestive-images\route.ts

```

1: import { promises as fs } from "fs";
2: import path from "path";
3: import { NextRequest, NextResponse } from "next/server";
4:
5: const folderName = "digestive";
6: const folderPath = path.join(process.cwd(), "public", folderName);
7:
8: export async function GET() {
9:     try {
10:         const files = await fs.readdir(folderPath);
11:
12:         const images = files.map((file) => ({
13:             label: file.split(".")[0], // e.g., "gut"
14:             url: `/${folderName}/${file}`, // e.g., "/digestive/gut.png"
15:         }));
16:
17:         return NextResponse.json({ success: true, images });
18:     } catch (err) {
19:         console.error("Failed to read digestive image folder:", err);

```

```

20:     return NextResponse.json(
21:       { success: false, error: "Could not read images" },
22:       { status: 500 }
23:     );
24:   }
25: }
26:
27: // DELETE handler for /api/digestive-images
28: export async function DELETE(req: NextRequest) {
29:   try {
30:     const { file } = await req.json();
31:
32:     if (!file) {
33:       return NextResponse.json(
34:         { success: false, error: "Missing file name" },
35:         { status: 400 }
36:       );
37:     }
38:
39:     const filePath = path.join(folderPath, file);
40:     await fs.unlink(filePath);
41:
42:     return NextResponse.json({ success: true });
43:   } catch (err) {
44:     console.error("Failed to delete digestive image:", err);
45:     return NextResponse.json(
46:       { success: false, error: "Could not delete image" },
47:       { status: 500 }
48:     );
49:   }
50: }
51:
52: export async function POST(req: NextRequest) {
53:   try {
54:     const formData = await req.formData();
55:     const file = formData.get("file") as File;
56:     const folder = formData.get("folder") as string;
57:     const label = formData.get("label") as string;
58:
59:     if (!file || !folder || !label) {
60:       return NextResponse.json(
61:         { success: false, error: "Missing file, label, or folder" },
62:         { status: 400 }
63:       );
64:     }
65:
66:     const arrayBuffer = await file.arrayBuffer();
67:     const buffer = Buffer.from(arrayBuffer);
68:
69:     const ext = file.name.split(".").pop() || "png";
70:     const safeName = `${label.replace(/\s+/g, "_").toLowerCase()}${ext}`;
71:     const filePath = path.join(process.cwd(), "public", folder, safeName);
72:
73:     await fs.mkdir(path.dirname(filePath), { recursive: true });
74:     await fs.writeFile(filePath, buffer);
75:
76:     return NextResponse.json({
77:       success: true,
78:       url: `/${folder}/${safeName}`,
79:     });
80:   } catch (err) {
81:     console.error("Upload failed:", err);
82:     return NextResponse.json(
83:       { success: false, error: "Upload failed" },
84:       { status: 500 }
85:     );
86:   }
87: }

```

---

## ■ File: app\api\lifestyle-images\route.ts

```
=====
1: import { promises as fs } from "fs";
2: import path from "path";
3: import { NextRequest, NextResponse } from "next/server";
4:
5: function getFolderPath(folder: string) {
6:   return path.join(process.cwd(), "public", folder);
7: }
8:
9: // GET: List images in a folder (e.g., /api/image-library?folder=lifestyle)
10: export async function GET(req: NextRequest) {
11:   const folder = req.nextUrl.searchParams.get("folder");
12:   if (!folder) {
13:     return NextResponse.json(
14:       { success: false, error: "Missing folder" },
15:       { status: 400 }
16:     );
17:   }
18:
19:   const folderPath = getFolderPath(folder);
20:
21:   try {
22:     const files = await fs.readdir(folderPath);
23:     const images = files.map((file) => ({
24:       label: file.split(".")[0],
25:       url: `/${folder}/${file}`,
26:     }));
27:
28:     return NextResponse.json({ success: true, images });
29:   } catch (err) {
30:     console.error("Failed to read image folder:", err);
31:     return NextResponse.json(
32:       { success: false, error: "Could not read images" },
33:       { status: 500 }
34:     );
35:   }
36: }
37:
38: // POST: Upload new image
39: export async function POST(req: NextRequest) {
40:   try {
41:     const formData = await req.formData();
42:     const file = formData.get("file") as File;
43:     const folder = formData.get("folder") as string;
44:     const label = formData.get("label") as string;
45:
46:     if (!file || !folder || !label) {
47:       return NextResponse.json(
48:         { success: false, error: "Missing file, label, or folder" },
49:         { status: 400 }
50:       );
51:     }
52:
53:     const arrayBuffer = await file.arrayBuffer();
54:     const buffer = Buffer.from(arrayBuffer);
55:
56:     const ext = file.name.split(".").pop() || "png";
57:     const safeName = `${label.replace(/\s+/g, "_").toLowerCase()}.${ext}`;
58:     const filePath = path.join(getFolderPath(folder), safeName);
59:
60:     await fs.mkdir(path.dirname(filePath), { recursive: true });
61:     await fs.writeFile(filePath, buffer);
62:
63:     return NextResponse.json({
64:       success: true,
65:       url: `/${folder}/${safeName}`,
66:     });
67:   } catch (err) {
68:     console.error("Upload failed:", err);
69:     return NextResponse.json(
70:       { success: false, error: "Upload failed" },
71:       { status: 500 }
72:     );
73:   }
74: }
```

```

72:     );
73:   }
74: }
75:
76: // DELETE: Remove image by name
77: export async function DELETE(req: NextRequest) {
78:   try {
79:     const { file, folder } = await req.json();
80:     if (!file || !folder) {
81:       return NextResponse.json(
82:         { success: false, error: "Missing file or folder" },
83:         { status: 400 }
84:       );
85:     }
86:
87:     const filePath = path.join(getFolderPath(folder), file);
88:     await fs.unlink(filePath);
89:
90:     return NextResponse.json({ success: true });
91:   } catch (err) {
92:     console.error("Delete failed:", err);
93:     return NextResponse.json(
94:       { success: false, error: "Delete failed" },
95:       { status: 500 }
96:     );
97:   }
98: }

```

## ■ File: app\api\lifestyle-upload\route.ts

```

1: import { promises as fs } from "fs";
2: import path from "path";
3: import { NextRequest, NextResponse } from "next/server";
4:
5: const uploadFolder = path.join(process.cwd(), "public", "life");
6:
7: export async function POST(req: NextRequest) {
8:   try {
9:     const formData = await req.formData();
10:    const file = formData.get("file") as File;
11:    const categoryId = formData.get("categoryId") as string;
12:
13:    if (!file || !categoryId) {
14:      return NextResponse.json(
15:        { error: "Missing file or categoryId" },
16:        { status: 400 }
17:      );
18:    }
19:
20:    const arrayBuffer = await file.arrayBuffer();
21:    const buffer = Buffer.from(arrayBuffer);
22:
23:    const ext = file.name.split(".").pop() || "png";
24:    const fileName = `${categoryId}.${ext}`;
25:    const filePath = path.join(uploadFolder, fileName);
26:
27:    await fs.mkdir(uploadFolder, { recursive: true });
28:    await fs.writeFile(filePath, buffer);
29:
30:    const imageUrl = `/life/${fileName}`;
31:    return NextResponse.json({ url: imageUrl });
32:  } catch (error) {
33:    console.error("Upload failed:", error);
34:    return NextResponse.json({ error: "Image upload failed" }, { status: 500 });
35:  }
36: }

```

## ■ File: app\api\nutrition\route.ts

```
=====
1: import { promises as fs } from "fs";
2: import path from "path";
3: import { NextRequest, NextResponse } from "next/server";
4:
5: const reportFilePath = path.join(process.cwd(), "data", "report-data.json");
6:
7: async function readReportData() {
8:   try {
9:     const data = await fs.readFile(reportFilePath, "utf-8");
10:    return JSON.parse(data);
11:   } catch (error) {
12:     console.error("? Failed to read report-data.json:", error);
13:     return {};
14:   }
15: }
16:
17: async function writeReportData(data: any) {
18:   try {
19:     await fs.writeFile(reportFilePath, JSON.stringify(data, null, 2), "utf-8");
20:   } catch (error) {
21:     console.error("? Failed to write report-data.json:", error);
22:     throw error;
23:   }
24: }
25:
26: export async function POST(req: NextRequest) {
27:   try {
28:     const body = await req.json();
29:     if (
30:       !body.nutritionData ||
31:       typeof body.nutritionData !== "object" ||
32:       typeof body.nutritionData.data !== "object"
33:     ) {
34:       return NextResponse.json(
35:         { error: "Invalid nutritionData format" },
36:         { status: 400 }
37:       );
38:     }
39:
40:     const reportData = await readReportData();
41:
42:     // ? Overwrite the entire nutritionData object
43:     reportData.nutritionData = {
44:       quote: body.nutritionData.quote || "",
45:       description: body.nutritionData.description || "",
46:       data: body.nutritionData.data || {},
47:     };
48:
49:     await writeReportData(reportData);
50:
51:     return NextResponse.json({ success: true });
52:   } catch (error) {
53:     console.error("? API error in POST /api/nutrition:", error);
54:     return NextResponse.json(
55:       { error: "Internal Server Error" },
56:       { status: 500 }
57:     );
58:   }
59: }
=====
```

## ■ File: app\api\patients-data\route.ts

```
=====
1: import { PrismaClient, Prisma } from "@prisma/client";
2: import { NextResponse } from "next/server";
3: import type { FamilyGeneticImpact } from "@prisma/client";
4: import { HealthConditionStatus } from "@types/report-types";
5:
6: type AllergiesAndSensitivity = {
7:   quote: string;
=====
```

```

8:   description: string;
9:   generalAdvice: string;
10:  data: Record<string, { id?: string; title: string; image: string }>;
11: };
12:
13: const prisma = new PrismaClient();
14:
15: // STEP 1 ? Include nested models
16: const reportInclude = Prisma.validator<Prisma.ReportInclude>()({
17:   dynamicDietFieldDefinitions: {
18:     include: {
19:       fields: true,
20:     },
21:   },
22:   patientDietAnalysis: true,
23:   nutritionData: true,
24:   lifestyleConditions: true,
25:   lifestyleCategoryImages: true,
26:   metabolicCoreData: true,
27:   digestiveHealthData: true,
28:   addictionData: true,
29:   sleepData: true,
30:   allergyData: true,
31:   geneTestResults: true,
32:   geneticCategories: true,
33:   preventiveTests: true,
34:   nutritionalSupplements: true,
35:   familyGeneticImpacts: true,
36:   healthSummaryEntries: true,
37:   genomicAnalysisTable: {
38:     include: {
39:       categories: {
40:         include: {
41:           subcategories: true,
42:         },
43:       },
44:     },
45:   },
46: });
47:
48: const patientInclude = Prisma.validator<Prisma.PatientDefaultArgs>()({
49:   include: {
50:     reports: {
51:       include: reportInclude,
52:     },
53:   },
54: });
55:
56: type FullPatient = Prisma.PatientGetPayload<typeof patientInclude>;
57:
58: // Utility to remove undefined fields
59: const pickDefined = (obj: Record<string, any>) =>
60:   Object.fromEntries(Object.entries(obj).filter(([_, v]) => v !== undefined));
61:
62: // ? GET ? returns structured patient + report data
63: export async function GET() {
64:   try {
65:     const rawPatients: FullPatient[] = await prisma.patient.findMany({
66:       include: patientInclude.include,
67:     });
68:
69:     const normalizedPatients = rawPatients.map((p) => ({
70:       id: p.id,
71:       info: {
72:         name: p.name ?? "",
73:         gender: p.gender ?? "",
74:         birthDate: p.birthDate ?? "",
75:         sampleCode: p.sampleCode ?? "",
76:         sampleDate: p.sampleDate ?? "",
77:         reportDate: p.reportDate ?? "",
78:         checkedBy: p.checkedBy ?? "",
79:         scientificContent: p.scientificContent ?? "",
80:         disclaimer: p.disclaimer ?? "",

```

```

81:     signature1: p.signature1 ?? "",
82:     signature2: p.signature2 ?? "",
83:   },
84:   reports: (p.reports ?? []).map((r) => {
85:     const preventiveHealthObj = {
86:       description: r.preventiveHealthDescription ?? "",
87:       diagnosticTests: {
88:         halfYearly: (r.preventiveTests ?? []).
89:           .filter((t: any) => t.frequency === "halfYearly")
90:           .map((t: any) => t.testName ?? ""),
91:         yearly: (r.preventiveTests ?? []).
92:           .filter((t: any) => t.frequency === "yearly")
93:           .map((t: any) => t.testName ?? ""),
94:       },
95:       nutritionalSupplements: (r.nutritionalSupplements ?? []).map(
96:         (s: any) => ({
97:           id: s.id || crypto.randomUUID(),
98:           supplement: s.supplement ?? "",
99:           needed: !!s.needed,
100:         })
101:       ),
102:     };
103:
104:     const lifestyleConditionsObj: any = {
105:       quote: r.lifestyleQuote ?? "",
106:       description: r.lifestyleDescription ?? "",
107:     };
108:     (r.lifestyleConditions ?? []).forEach((condition: any) => {
109:       if (!lifestyleConditionsObj[condition.categoryId]) {
110:         lifestyleConditionsObj[condition.categoryId] = {};
111:       }
112:       lifestyleConditionsObj[condition.categoryId][
113:         condition.conditionName
114:       ] = {
115:         status: condition.status ?? "",
116:         description: condition.description ?? "",
117:         sensitivity: condition.sensitivity ?? "",
118:         avoid: condition.avoid ?? [],
119:         follow: condition.follow ?? [],
120:         consume: condition.consume ?? [],
121:         monitor: condition.monitor ?? [],
122:         avoidLabel: condition.avoidLabel ?? "AVOID",
123:         followLabel: condition.followLabel ?? "FOLLOW",
124:         consumeLabel: condition.consumeLabel ?? "CONSUME",
125:         monitorLabel: condition.monitorLabel ?? "MONITOR",
126:       };
127:     });
128:
129:     // --- Transform allergyData array into object ---
130:     const allergiesAndSensitivityObj: AllergiesAndSensitivity = {
131:       quote: r.allergyQuote ?? "",
132:       description: r.allergyDescription ?? "",
133:       generalAdvice: r.allergyGeneralAdvice ?? "",
134:       data: {},
135:     };
136:
137:     (r.allergyData ?? []).forEach((item: any) => {
138:       allergiesAndSensitivityObj.data[item.key || crypto.randomUUID()] = {
139:         id: item.id || crypto.randomUUID(),
140:         title: item.title ?? "",
141:         image: item.image ?? "",
142:       };
143:     });
144:
145:     // --- Transform sleep data into expected format ---
146:     const sleepAndRestObj = {
147:       quote: r.sleepQuote ?? "",
148:       description: r.sleepDescription ?? "",
149:       data: (r.sleepData ?? []).reduce((acc, item) => {
150:         acc[item.key] = {
151:           id: item.id,
152:           title: item.title ?? "",
153:           intervention: item.intervention ?? "",

```



```

154:         image: item.image ?? "",
155:     };
156:     return acc;
157: }, {} as Record<string, { id: string; title: string; intervention: string; image: string }>),
158: };
159:
160: // --- Transform addictionData array into object ---
161: const genesAndAddictionObj: any = {
162:     quote: r.addictionQuote ?? "",
163:     description: r.addictionDescription ?? "",
164:     data: {},
165: };
166: (r.addictionData ?? []).forEach((item: any) => {
167:     genesAndAddictionObj.data[item.key] = {
168:         id: item.id,
169:         title: item.title ?? "",
170:         icon: item.icon ?? "",
171:         sensitivityIcon: item.sensitivityIcon ?? "",
172:     };
173: });
174:
175: // --- Transform lifestyleCategoryImages array into map ---
176: const lifestyleCategoryImagesObj: Record<string, string> = {};
177: (r.lifestyleCategoryImages ?? []).forEach((img: any) => {
178:     lifestyleCategoryImagesObj[img.categoryId] = img.imageUrl ?? "";
179: });
180:
181: // --- Transform metabolicCoreData array into object ---
182: const metabolicCoreObj: any = {
183:     quote: r.metabolicQuote ?? "",
184:     description: r.metabolicDescription ?? "",
185: };
186:
187: (r.metabolicCoreData ?? []).forEach((entry: any) => {
188:     if (!metabolicCoreObj[entry.area]) {
189:         metabolicCoreObj[entry.area] = {
190:             impact: entry.areaImpact ?? "",
191:             advice: entry.areaAdvice ?? "", // Area-level advice
192:             genes: [],
193:         };
194:     }
195:     metabolicCoreObj[entry.area].genes.push({
196:         id: entry.id,
197:         name: entry.geneName ?? "",
198:         genotype: entry.genotype ?? "",
199:         impact: entry.impact ?? "",
200:         // No individual advice field
201:     });
202: });
203:
204: // --- Transform digestiveHealthData array into object ---
205: const digestiveHealthObj: any = {
206:     quote: r.digestiveQuote ?? "",
207:     description: r.digestiveDescription ?? "",
208:     data: {},
209: };
210: (r.digestiveHealthData ?? []).forEach((item: any) => {
211:     digestiveHealthObj.data[item.id] = {
212:         title: item.title ?? "",
213:         icon: item.icon ?? "",
214:         sensitivity: item.sensitivity ?? "",
215:         good: item.good ?? "",
216:         bad: item.bad ?? "",
217:     };
218: });
219:
220: return {
221:     id: r.id,
222:     name: r.name,
223:     patientId: r.patientId,
224:     content: {
225:         introduction: r.introduction ?? "",
226:         genomicsExplanation: r.genomicsExplanation ?? "",

```

```

227:         genesHealthImpact: r.genesHealthImpact ?? "",
228:         fundamentalsPRS: r.fundamentalsPRS ?? "",
229:         utilityDoctors: r.utilityDoctors ?? "",
230:         microarrayExplanation: r.microarrayExplanation ?? "",
231:         microarrayData: r.microarrayData ?? "",
232:     },
233:     settings: {
234:         title: r.title ?? "",
235:         subtitle: r.subtitle ?? "",
236:         companyName: r.companyName ?? "",
237:         headerColor: r.headerColor ?? "",
238:         accentColor: r.accentColor ?? "",
239:         fonts: {
240:             primary: r.primaryFont ?? "",
241:             secondary: r.secondaryFont ?? "",
242:             mono: r.monoFont ?? "",
243:         },
244:     },
245:     nutrigenomicsSummary: r.nutrigenomicsSummary ?? "",
246:     exerciseGenomicsSummary: r.exerciseGenomicsSummary ?? "",
247:     metabolicSummary: {
248:         strengths: r.metabolicStrengths ?? [],
249:         weaknesses: r.metabolicWeaknesses ?? [],
250:     },
251:     dietFieldCategories: r.dietFieldCategories ?? [],
252:     nutritionQuote: r.nutritionQuote ?? "",
253:     nutritionDescription: r.nutritionDescription ?? "",
254:     sportsAndFitness: r.sportsAndFitness ?? {},
255:
256:     digestiveHealth: digestiveHealthObj,
257:     genesAndAddiction: genesAndAddictionObj,
258:     sleepAndRest: sleepAndRestObj,
259:     allergiesAndSensitivity: allergiesAndSensitivityObj,
260:
261:     sleepQuote: r.sleepQuote ?? "",
262:     sleepDescription: r.sleepDescription ?? "",
263:     allergyQuote: r.allergyQuote ?? "",
264:     allergyDescription: r.allergyDescription ?? "",
265:     allergyGeneralAdvice: r.allergyGeneralAdvice ?? "",
266:
267:     patientDietAnalysisResults: r.patientDietAnalysis ?? [],
268:     nutritionData: {
269:         quote: r.nutritionQuote ?? "",
270:         description: r.nutritionDescription ?? "",
271:         data: (r.nutritionData ?? []).reduce((acc: any, item) => {
272:             if (!acc[item.section]) acc[item.section] = {};
273:             acc[item.section][item.field] = {
274:                 id: item.id,
275:                 score: item.score,
276:                 healthImpact: item.healthImpact,
277:                 intakeLevel: item.intakeLevel,
278:                 source: item.source,
279:             };
280:             return acc;
281:         }, {}),
282:     },
283:
284:     // ? FIXED: Proper dynamic diet field transformation for GET
285:     dynamicDietFieldDefinitions: (
286:         r.dynamicDietFieldDefinitions ?? []
287:     ).map((definition: any) => ({
288:         meta: {
289:             quote: definition.quote ?? "",
290:             description: definition.description ?? "",
291:         },
292:         fields: (definition.fields ?? []).map((field: any) => ({
293:             id: field.fieldId ?? field.id ?? "",
294:             label: field.label ?? "",
295:             category: field.category ?? "",
296:             min: field.min ?? 0,
297:             max: field.max ?? 100,
298:             highRecommendation: field.highRecommendation ?? "",
299:             normalRecommendation: field.normalRecommendation ?? "",

```

```

300:         lowRecommendation: field.lowRecommendation ?? "",
301:     })),
302: })),
303:
304:     lifestyleConditions: lifestyleConditionsObj,
305:     lifestyleCategoryImages: lifestyleCategoryImagesObj,
306:     metabolicCore: metabolicCoreObj,
307:     preventiveHealth: preventiveHealthObj,
308:
309:     sleepData: r.sleepData ?? [],
310:     allergyData: r.allergyData ?? [],
311:     geneTestResults: r.geneTestResults ?? [],
312:     categories: r.geneticCategories ?? [],
313:     nutritionalSupplements: r.nutritionalSupplements ?? [],
314:     familyGeneticImpactSection: {
315:         description: r.familyGeneticImpactDescription ?? "",
316:         impacts: (r.familyGeneticImpacts ?? []).map((impact: any) => ({
317:             id: impact.id || crypto.randomUUID(),
318:             gene: impact.gene ?? "",
319:             normalAlleles: impact.normalAlleles ?? "",
320:             yourResult: impact.yourResult ?? "",
321:             healthImpact: impact.healthImpact ?? "",
322:             reportId: impact.reportId ?? r.id,
323:         })),
324:     },
325:     genomicAnalysisTable: r.genomicAnalysisTable
326:     ? {
327:         description: r.genomicAnalysisTable.description ?? "",
328:         categories: (r.genomicAnalysisTable.categories ?? []).map(
329:             (cat: any) => ({
330:                 category: cat.category ?? "",
331:                 subcategories: (cat.subcategories ?? []).map(
332:                     (sub: any) => ({
333:                         area: sub.area ?? "",
334:                         trait: sub.trait ?? "",
335:                         genes: sub.genes ?? [],
336:                     })
337:                 ),
338:             })
339:         ),
340:     }
341:     : undefined,
342:     healthSummary: {
343:         description: r.healthSummaryDescription ?? "",
344:         data: (r.healthSummaryEntries ?? []).map((entry: any) => ({
345:             title: entry.title ?? "",
346:             description: entry.description ?? "",
347:         })),
348:     },
349: };
350: })),
351: }));
352:
353: return NextResponse.json(normalizedPatients);
354: } catch (error: any) {
355:     console.error("GET error:", error);
356:     return NextResponse.json({ error: error.message }, { status: 500 });
357: } finally {
358:     await prisma.$disconnect();
359: }
360: }
361:
362: // ? POST ? supports partial save/update
363: export async function POST(request: Request) {
364:     try {
365:         const patients = await request.json();
366:
367:         if (!Array.isArray(patients)) {
368:             return NextResponse.json(
369:                 { error: "Patients data must be an array" },
370:                 { status: 400 }
371:             );
372:         }

```

```

373:
374:   for (const patient of patients) {
375:     const { id, info, reports } = patient;
376:
377:     if (!id || typeof id !== "string") {
378:       return NextResponse.json(
379:         { error: "Each patient must have an ID" },
380:         { status: 400 }
381:       );
382:     }
383:
384:     const patientData = pickDefined({
385:       id,
386:       name: info?.name,
387:       gender: info?.gender,
388:       birthDate: info?.birthDate,
389:       sampleCode: info?.sampleCode,
390:       sampleDate: info?.sampleDate,
391:       reportDate: info?.reportDate,
392:       checkedBy: info?.checkedBy,
393:       scientificContent: info?.scientificContent,
394:       disclaimer: info?.disclaimer,
395:       signature1: info?.signature1,
396:       signature2: info?.signature2,
397:     });
398:
399:     await prisma.patient.upsert({
400:       where: { id },
401:       update: patientData as Prisma.PatientUpdateInput,
402:       create: patientData as Prisma.PatientCreateInput,
403:     });
404:
405:     if (Array.isArray(reports)) {
406:       for (const report of reports) {
407:         const content = report.content || {};
408:         const settings = report.settings || {};
409:         const fonts = settings.fonts || {};
410:         const metabolicSummary = report.metabolicSummary || {};
411:
412:         const reportData = pickDefined({
413:           id: report.id,
414:           name: report.name,
415:           introduction: content.introduction,
416:           genomicsExplanation: content.genomicsExplanation,
417:           genesHealthImpact: content.genesHealthImpact,
418:           fundamentalsPRS: content.fundamentalsPRS,
419:           utilityDoctors: content.utilityDoctors,
420:           microarrayExplanation: content.microarrayExplanation,
421:           microarrayData: content.microarrayData,
422:           title: settings.title,
423:           subtitle: settings.subtitle,
424:           companyName: settings.companyName,
425:           headerColor: settings.headerColor,
426:           accentColor: settings.accentColor,
427:           primaryFont: fonts.primary,
428:           secondaryFont: fonts.secondary,
429:           monoFont: fonts.mono,
430:           nutrigenomicsSummary: report.nutrigenomicsSummary || "",
431:           exerciseGenomicsSummary: report.exerciseGenomicsSummary || "",
432:           metabolicStrengths: metabolicSummary.strengths || [],
433:           metabolicWeaknesses: metabolicSummary.weaknesses || [],
434:           dietFieldCategories: report.dietFieldCategories || [],
435:           nutritionQuote: report.nutritionQuote || "",
436:           nutritionDescription: report.nutritionDescription || "",
437:           lifestyleQuote: report.lifestyleQuote || "",
438:           lifestyleDescription: report.lifestyleDescription || "",
439:           metabolicQuote: report.metabolicQuote || "",
440:           metabolicDescription: report.metabolicDescription || "",
441:           digestiveQuote: report.digestiveQuote || "",
442:           digestiveDescription: report.digestiveDescription || "",
443:           addictionQuote: report.addictionQuote || "",
444:           addictionDescription: report.addictionDescription || "",
445:           sleepQuote: report.sleepQuote || "",

```

```

446:         sleepDescription: report.sleepDescription || "",
447:         allergyQuote: report.allergyQuote || "",
448:         allergyDescription: report.allergyDescription || "",
449:         allergyGeneralAdvice: report.allergyGeneralAdvice || "",
450:         healthSummaryDescription: report.healthSummary?.description || "",
451:     });
452:
453:     await prisma.report.upsert({
454:       where: { id: report.id },
455:       update: {
456:         ...reportData,
457:         sportsAndFitness: report.sportsAndFitness || {},
458:       },
459:       create: {
460:         ...reportData,
461:         patient: { connect: { id } },
462:         sportsAndFitness: report.sportsAndFitness || {},
463:       },
464:     });
465:
466:     const handleNested = async (
467:       model: any,
468:       data: any[],
469:       deleteWhere: object,
470:       injectFields: object = {}
471:     ) => {
472:       if (!Array.isArray(data)) {
473:         console.warn(
474:           "? handleNested: Provided data is not an array:",
475:           data
476:         );
477:         return;
478:       }
479:
480:       try {
481:         await model.deleteMany({ where: deleteWhere });
482:
483:         if (data.length === 0) {
484:           console.info("? No data to insert for", model.name || "model");
485:           return;
486:         }
487:
488:         const modelName =
489:           model?.name || model?.toString?.() || "Unknown Model";
490:         console.log(`Inserting into ${modelName}:`, data);
491:
492:         await model.createMany({
493:           data: data.map((item) => ({ ...item, ...injectFields })),
494:           skipDuplicates: false,
495:         });
496:       } catch (err) {
497:         const modelName =
498:           model?.name || model?.toString?.() || "Unknown Model";
499:         console.error(`? handleNested failed for ${modelName}:`, err);
500:         throw err;
501:       }
502:     };
503:
504:     if (
505:       report.healthSummary?.data &&
506:       Array.isArray(report.healthSummary.data)
507:     ) {
508:       const healthSummaryWithIds = report.healthSummary.data.map(
509:         (entry: any) => ({
510:           id: entry.id || crypto.randomUUID(),
511:           reportId: report.id,
512:           title: entry.title || "",
513:           description: entry.description || "",
514:         })
515:       );
516:
517:       await handleNested(
518:         prisma.healthSummaryEntry,

```

```

519:         healthSummaryWithIds,
520:         {
521:             reportId: report.id,
522:         }
523:     );
524: }
525:
526: // ? COMPLETELY FIXED: Dynamic Diet Fields Processing
527: if (
528:     report.dynamicDietFieldDefinitions &&
529:     Array.isArray(report.dynamicDietFieldDefinitions)
530: ) {
531:     console.log(
532:         "? Processing dynamic diet fields for report:",
533:         report.id
534:     );
535:     console.log(
536:         "? Incoming definitions:",
537:         JSON.stringify(report.dynamicDietFieldDefinitions, null, 2)
538:     );
539:
540:     // First, delete ALL existing dynamic diet field definitions and their fields
541:     const deleteResult =
542:         await prisma.dynamicDietFieldDefinition.deleteMany({
543:             where: { reportId: report.id },
544:         });
545:     console.log("?? Deleted existing definitions:", deleteResult.count);
546:
547:     // Process each definition
548:     for (const [
549:         defIndex,
550:         definition,
551:     ] of report.dynamicDietFieldDefinitions.entries()) {
552:         console.log(`? Processing definition ${defIndex}:`, {
553:             metaQuote: definition.meta?.quote?.substring(0, 50) + "...",
554:             metaDescription:
555:                 definition.meta?.description?.substring(0, 50) + "...",
556:             fieldsCount: definition.fields?.length || 0,
557:         });
558:
559:         // Create the definition with meta information
560:         const createdDefinition =
561:             await prisma.dynamicDietFieldDefinition.create({
562:                 data: {
563:                     reportId: report.id,
564:                     quote: definition.meta?.quote || "",
565:                     description: definition.meta?.description || "",
566:                 },
567:             });
568:         console.log("? Created definition ID:", createdDefinition.id);
569:
570:         // Create fields if they exist
571:         if (
572:             definition.fields &&
573:             Array.isArray(definition.fields) &&
574:             definition.fields.length > 0
575:         ) {
576:             const fieldsData = definition.fields.map(
577:                 (field: any, fieldIndex: number) => {
578:                     console.log(` ? Field ${fieldIndex}:`, {
579:                         id: field.id,
580:                         label: field.label,
581:                         category: field.category,
582:                         min: field.min,
583:                         max: field.max,
584:                     });
585:
586:                     return {
587:                         definitionId: createdDefinition.id,
588:                         fieldId: field.id || crypto.randomUUID(), // Ensure we have a stable ID
589:                         label: field.label || "",
590:                         category: field.category || "",
591:                         min: typeof field.min === "number" ? field.min : 1,

```

```

592:         max: typeof field.max === "number" ? field.max : 10,
593:         highRecommendation: field.highRecommendation || "",
594:         normalRecommendation: field.normalRecommendation || "",
595:         lowRecommendation: field.lowRecommendation || "",
596:     };
597: }
598: );
599:
600: const fieldsResult = await prisma.dynamicDietField.createMany({
601:   data: fieldsData,
602:   skipDuplicates: false,
603: });
604: console.log(
605:   `? Created ${fieldsResult.count} fields for definition ${createdDefinition.id}`
606: );
607: }
608: }
609:
610: // Verify the save
611: const verifyDefinitions =
612:   await prisma.dynamicDietFieldDefinition.findMany({
613:     where: { reportId: report.id },
614:     include: { fields: true },
615:   });
616: console.log(`? VERIFICATION:`);
617: console.log(` - Definitions saved: ${verifyDefinitions.length}`);
618: console.log(
619:   ` - Fields saved: ${verifyDefinitions.reduce(
620:     (sum, def) => sum + def.fields.length,
621:     0
622:   )}`
623: );
624:
625: verifyDefinitions.forEach((def, i) => {
626:   console.log(` - Definition ${i}: ${def.fields.length} fields`);
627:   def.fields.forEach((field, j) => {
628:     console.log(
629:       ` - Field ${j}: "${field.label}" (category: "${field.category})"`
630:     );
631:   });
632: });
633: }
634:
635: // ? Patient Diet Analysis with UUID
636: if (report.patientDietAnalysisResults) {
637:   // Get all existing analysis for this report
638:   const existingAnalysis = await prisma.patientDietAnalysis.findMany({
639:     where: { reportId: report.id },
640:     select: { id: true, fieldId: true },
641:   });
642:
643:   const dietAnalysisWithIds = report.patientDietAnalysisResults.map(
644:     (field: any) => {
645:       const existing = existingAnalysis.find(
646:         (e) => e.fieldId === field.fieldId
647:       );
648:
649:       return {
650:         id: existing?.id || field.id || crypto.randomUUID(),
651:         reportId: report.id,
652:         fieldId: field.fieldId || crypto.randomUUID(),
653:         score: field.score ?? 0,
654:         level: field.level || "NORMAL",
655:         selectedLevel: field.selectedLevel || "NORMAL",
656:         recommendation: field.recommendation || "",
657:         lowRecommendation:
658:           field.lowRecommendation || field.recommendations?.LOW || "",
659:         normalRecommendation:
660:           field.normalRecommendation ||
661:           field.recommendations?.NORMAL ||
662:           "",
663:         highRecommendation:
664:           field.highRecommendation ||

```

```

665:             field.recommendations?.HIGH ||
666:             "",
667:         };
668:     }
669: );
670:
671: await handleNested(
672:     prisma.patientDietAnalysis,
673:     dietAnalysisWithIds,
674:     { reportId: report.id }
675: );
676: }
677:
678: // In patient-data/route.ts POST section:
679: if (
680:     report.nutritionData &&
681:     typeof report.nutritionData === "object"
682: ) {
683:     const { quote, description, data } = report.nutritionData;
684:
685:     const nutritionArray: any[] = [];
686:
687:     if (data && typeof data === "object") {
688:         for (const section of Object.keys(data)) {
689:             const fields = data[section];
690:             if (fields && typeof fields === "object") {
691:                 // ? Add this check
692:                 for (const field of Object.keys(fields)) {
693:                     const nutrient = fields[field];
694:                     if (nutrient && typeof nutrient === "object") {
695:                         // ? Add this check
696:                         nutritionArray.push({
697:                             id: nutrient.id || crypto.randomUUID(),
698:                             reportId: report.id,
699:                             section,
700:                             field,
701:                             score: Number(nutrient.score) || 0,
702:                             healthImpact: nutrient.healthImpact || "",
703:                             intakeLevel: nutrient.intakeLevel || "",
704:                             source: nutrient.source || "",
705:                         });
706:                     }
707:                 }
708:             }
709:         }
710:     }
711:
712:     // Save quote and description directly to report
713:     await prisma.report.update({
714:         where: { id: report.id },
715:         data: {
716:             nutritionQuote: quote || "",
717:             nutritionDescription: description || "",
718:         },
719:     });
720:
721:     // ? IMPORTANT: Clear existing nutrition data before saving new data
722:     await prisma.nutritionData.deleteMany({
723:         where: { reportId: report.id },
724:     });
725:
726:     // Save nutrition data
727:     if (nutritionArray.length > 0) {
728:         await prisma.nutritionData.createMany({
729:             data: nutritionArray,
730:         });
731:     }
732: }
733:
734: // ? Lifestyle Conditions ? normalize object to array before saving
735: if (
736:     report.lifestyleConditions &&
737:     !Array.isArray(report.lifestyleConditions)

```



```

738:     ) {
739:         const { quote, description, ...categories } =
740:             report.lifestyleConditions;
741:
742:         // Save quote + description to report table
743:         await prisma.report.update({
744:             where: { id: report.id },
745:             data: {
746:                 lifestyleQuote: quote || "",
747:                 lifestyleDescription: description || "",
748:             },
749:         });
750:
751:         // Convert object format into array for Prisma
752:         const lifestyleConditionsArray = Object.entries(categories).flatMap(
753:             ([categoryId, conditions]) => {
754:                 if (
755:                     !conditions ||
756:                     typeof conditions !== "object" ||
757:                     Array.isArray(conditions)
758:                 ) {
759:                     return [];
760:                 }
761:                 return Object.entries(
762:                     conditions as Record<string, HealthConditionStatus>
763:                 ).map(([conditionName, conditionData]) => ({
764:                     id: (conditionData as any).id || crypto.randomUUID(),
765:                     reportId: report.id,
766:                     categoryId,
767:                     conditionName,
768:                     status: conditionData.status || "",
769:                     description: conditionData.description || "",
770:                     sensitivity: conditionData.sensitivity || "low",
771:                     avoid: conditionData.avoid || [],
772:                     follow: conditionData.follow || [],
773:                     consume: conditionData.consume || [],
774:                     monitor: conditionData.monitor || [],
775:                     avoidLabel: conditionData.avoidLabel || "AVOID",
776:                     followLabel: conditionData.followLabel || "FOLLOW",
777:                     consumeLabel: conditionData.consumeLabel || "CONSUME",
778:                     monitorLabel: conditionData.monitorLabel || "MONITOR",
779:                 }));
780:             }
781:         );
782:
783:         await handleNested(
784:             prisma.lifestyleCondition,
785:             lifestyleConditionsArray,
786:             { reportId: report.id }
787:         );
788:     } else if (Array.isArray(report.lifestyleConditions)) {
789:         // If already array, just save directly
790:         const lifestyleConditionsWithIds = report.lifestyleConditions.map(
791:             (condition: any) => ({
792:                 ...condition,
793:                 id: condition.id || crypto.randomUUID(),
794:                 reportId: report.id,
795:             })
796:         );
797:
798:         await handleNested(
799:             prisma.lifestyleCondition,
800:             lifestyleConditionsWithIds,
801:             { reportId: report.id }
802:         );
803:     }
804:
805:     // ? Lifestyle Category Images ? normalize object to array before saving
806:     if (
807:         report.lifestyleCategoryImages &&
808:         !Array.isArray(report.lifestyleCategoryImages)
809:     ) {
810:         const imagesArray = Object.entries(

```

```

811:         report.lifestyleCategoryImages as Record<string, string>
812:     ).map(([categoryId, imageUrl]) => ({
813:         id: crypto.randomUUID(),
814:         reportId: report.id,
815:         categoryId,
816:         imageUrl,
817:     }));
818:
819:     await handleNested(prisma.lifestyleCategoryImage, imagesArray, {
820:         reportId: report.id,
821:     });
822: } else if (Array.isArray(report.lifestyleCategoryImages)) {
823:     const lifestyleCategoryImagesWithIds =
824:         report.lifestyleCategoryImages.map((image: any) => ({
825:             ...image,
826:             id: image.id || crypto.randomUUID(),
827:             reportId: report.id,
828:         }));
829:
830:     await handleNested(
831:         prisma.lifestyleCategoryImage,
832:         lifestyleCategoryImagesWithIds,
833:         { reportId: report.id }
834:     );
835: }
836:
837: // ? Metabolic Core Data with UUID
838: if (report.metabolicCore) {
839:     const metabolicCoreArray: Prisma.MetabolicCoreDataUncheckedCreateInput[] =
840:         [];
841:
842:     Object.entries(report.metabolicCore).forEach(
843:         ([areaName, data]: [string, any]) => {
844:             if (areaName === "quote" || areaName === "description") return;
845:
846:             if (Array.isArray(data.genes)) {
847:                 data.genes.forEach((gene: any) => {
848:                     metabolicCoreArray.push({
849:                         id: gene.id || crypto.randomUUID(),
850:                         reportId: report.id,
851:                         area: areaName,
852:                         geneName: gene.name || "",
853:                         genotype: gene.genotype || "",
854:                         impact: gene.impact || "",
855:                         areaImpact: data.impact || null,
856:                         areaAdvice: data.advice || null,
857:                     });
858:                 });
859:             }
860:         }
861:     );
862:
863:     await handleNested(prisma.metabolicCoreData, metabolicCoreArray, {
864:         reportId: report.id,
865:     });
866:
867:     await prisma.report.update({
868:         where: { id: report.id },
869:         data: {
870:             metabolicQuote: report.metabolicCore.quote || null,
871:             metabolicDescription: report.metabolicCore.description || null,
872:         },
873:     });
874: }
875:
876: // ? Digestive Health Data with UUID
877: if (report.digestiveHealth) {
878:     const { quote, description, data } = report.digestiveHealth;
879:
880:     // Save quote & description into Report table first
881:     await prisma.report.update({
882:         where: { id: report.id },
883:         data: {

```

```

884:         digestiveQuote: quote || "",
885:         digestiveDescription: description || "",
886:     },
887: });
888:
889: // Normalize data object to array for DB
890: const digestiveArray: any[] = Object.entries(data || {}).map(
891:     ([key, entry]: [string, any]) => ({
892:         id: entry.id || crypto.randomUUID(),
893:         reportId: report.id,
894:         key, // this is the logical key used in frontend
895:         title: entry.title || "",
896:         icon: entry.icon || "",
897:         sensitivity: entry.sensitivity || "",
898:         good: entry.good || "",
899:         bad: entry.bad || "",
900:     })
901: );
902:
903: await handleNested(prisma.digestiveHealthData, digestiveArray, {
904:     reportId: report.id,
905: });
906: }
907:
908: // ? Genes & Addiction Data with UUID
909: if (report.genesAndAddiction) {
910:     const { quote, description, data } = report.genesAndAddiction;
911:
912:     // Save quote & description into Report table
913:     await prisma.report.update({
914:         where: { id: report.id },
915:         data: {
916:             additionQuote: quote || "",
917:             additionDescription: description || "",
918:         },
919:     });
920:
921:     // Normalize object to array for DB
922:     const additionArray: any[] = Object.entries(data || {}).map(
923:         ([key, entry]: [string, any]) => ({
924:             id: entry.id || crypto.randomUUID(),
925:             reportId: report.id,
926:             key, // ? Save the frontend object key back to DB
927:             title: entry.title || "",
928:             icon: entry.icon || "",
929:             sensitivityIcon: entry.sensitivityIcon || "",
930:         })
931:     );
932:
933:     await handleNested(prisma.addictionData, additionArray, {
934:         reportId: report.id,
935:     });
936: }
937:
938: // ? Sleep Data with UUID
939: if (report.sleepAndRest) {
940:     const { quote, description, data } = report.sleepAndRest;
941:
942:     // 1?? Save quote/description in Report table
943:     await prisma.report.update({
944:         where: { id: report.id },
945:         data: {
946:             sleepQuote: quote || "",
947:             sleepDescription: description || "",
948:         },
949:     });
950:
951:     // 2?? Prepare array for SleepData
952:     const sleepDataArray = Object.entries(data || {}).map(
953:         ([key, entry]: [string, any]) => ({
954:             id: entry.id || crypto.randomUUID(),
955:             reportId: report.id,
956:             key,

```

```

957:         title: entry.title || "", // <-- Added
958:         intervention: entry.intervention || "",
959:         image: entry.image || "",
960:     })
961: };
962:
963: // 3?? Save with handleNested
964: await handleNested(prisma.sleepData, sleepDataArray, {
965:     reportId: report.id,
966: });
967: }
968:
969: // ? Allergies & Sensitivity ? normalize object to array before saving
970: if (
971:     report.allergiesAndSensitivity &&
972:     typeof report.allergiesAndSensitivity === "object"
973: ) {
974:     const { quote, description, generalAdvice, data } =
975:         report.allergiesAndSensitivity;
976:
977:     // 1?? Save quote/description/generalAdvice directly to Report
978:     await prisma.report.update({
979:         where: { id: report.id },
980:         data: {
981:             allergyQuote: quote || "",
982:             allergyDescription: description || "",
983:             allergyGeneralAdvice: generalAdvice || "",
984:         },
985:     });
986:
987:     // 2?? Convert object map -> array for DB
988:     const allergyDataArray = Object.entries(data || {}).map(
989:         ([key, value]) => {
990:             const val = value as {
991:                 id?: string;
992:                 title?: string;
993:                 image?: string;
994:             };
995:             return {
996:                 id: val.id || crypto.randomUUID(),
997:                 reportId: report.id,
998:                 key,
999:                 title: val.title || "",
1000:                 image: val.image || "",
1001:             };
1002:         }
1003:     );
1004:
1005:     // 3?? Save in DB, replacing old ones
1006:     await handleNested(prisma.allergyData, allergyDataArray, {
1007:         reportId: report.id,
1008:     });
1009: }
1010:
1011: // ? Gene Test Results with UUID
1012: if (report.geneTestResults) {
1013:     const geneTestResultsWithIds = report.geneTestResults.map(
1014:         (gene: any) => ({
1015:             ...gene,
1016:             id: gene.id || crypto.randomUUID(),
1017:             reportId: report.id,
1018:         })
1019:     );
1020:
1021:     await handleNested(prisma.geneTestResult, geneTestResultsWithIds, {
1022:         reportId: report.id,
1023:     });
1024: }
1025:
1026: // ? Genetic Categories with UUID (existing implementation)
1027: if (report.categories) {
1028:     const categoriesWithIds = report.categories.map((cat: any) => ({
1029:         ...cat,

```

```

1030:         reportId: report.id,
1031:         categoryId: cat.categoryId || crypto.randomUUID(),
1032:     ));
1033:
1034:     await handleNested(prisma.geneticCategory, categoriesWithIds, {
1035:         reportId: report.id,
1036:     });
1037: }
1038:
1039: // ? Preventive Health ? tests + supplements
1040: const hasPreventivePayload =
1041:     report.preventiveHealth !== undefined ||
1042:     report.nutritionalSupplements !== undefined;
1043:
1044: if (hasPreventivePayload) {
1045:     // Update the report with preventive health description
1046:     if (report.preventiveHealth?.description) {
1047:         await prisma.report.update({
1048:             where: { id: report.id },
1049:             data: {
1050:                 preventiveHealthDescription:
1051:                     report.preventiveHealth.description,
1052:             },
1053:         });
1054:     }
1055:
1056:     // Normalize diagnostic tests
1057:     const diagnosticTests = (report.preventiveHealth
1058:         ?.diagnosticTests ?? {
1059:             halfYearly: [],
1060:             yearly: [],
1061:         }) as { halfYearly?: string[]; yearly?: string[] };
1062:
1063:     const testsArray = [
1064:         ...(diagnosticTests.halfYearly || []).map((name: string) => ({
1065:             id: crypto.randomUUID(),
1066:             testName: name ?? "",
1067:             frequency: "halfYearly",
1068:             reportId: report.id,
1069:         })),
1070:         ...(diagnosticTests.yearly || []).map((name: string) => ({
1071:             id: crypto.randomUUID(),
1072:             testName: name ?? "",
1073:             frequency: "yearly",
1074:             reportId: report.id,
1075:         })),
1076:     ];
1077:
1078:     await handleNested(prisma.preventiveTest, testsArray, {
1079:         reportId: report.id,
1080:     });
1081:
1082:     // Normalize supplements ? prefer preventiveHealth, fallback to legacy
1083:     const incomingSupplements =
1084:         report.preventiveHealth?.nutritionalSupplements ??
1085:         report.nutritionalSupplements ??
1086:         [];
1087:
1088:     const supplementsNormalized = Array.isArray(incomingSupplements)
1089:         ? incomingSupplements.map((s: any) => ({
1090:             id: s?.id || crypto.randomUUID(),
1091:             supplement:
1092:                 typeof s?.supplement === "string"
1093:                 ? s.supplement
1094:                 : String(s?.title ?? s ?? ""),
1095:             needed: typeof s?.needed === "boolean" ? s.needed : true,
1096:             reportId: report.id,
1097:         }))
1098:         : [];
1099:
1100:     await handleNested(
1101:         prisma.nutritionalSupplement,
1102:         supplementsNormalized,

```

```

1103:         {
1104:             reportId: report.id,
1105:         }
1106:     );
1107: }
1108:
1109: // ? Genomic Analysis Table with nested structure
1110: if (report.genomicAnalysisTable) {
1111:     const { description, categories } = report.genomicAnalysisTable;
1112:
1113:     // First, create or update the genomic analysis table
1114:     await prisma.genomicAnalysisTable.upsert({
1115:         where: { reportId: report.id },
1116:         update: { description: description || "" },
1117:         create: {
1118:             reportId: report.id,
1119:             description: description || "",
1120:         },
1121:     });
1122:
1123:     // Get the genomic analysis table ID
1124:     const genomicTable = await prisma.genomicAnalysisTable.findUnique({
1125:         where: { reportId: report.id },
1126:     });
1127:
1128:     if (genomicTable && Array.isArray(categories)) {
1129:         // Clear existing categories and subcategories
1130:         await prisma.genomicSubcategory.deleteMany({
1131:             where: {
1132:                 genomicCategoryGroup: {
1133:                     genomicAnalysisTableId: genomicTable.id,
1134:                 },
1135:             },
1136:         });
1137:         await prisma.genomicCategoryGroup.deleteMany({
1138:             where: { genomicAnalysisTableId: genomicTable.id },
1139:         });
1140:
1141:         // Create new categories and subcategories
1142:         for (const category of categories) {
1143:             const categoryGroup = await prisma.genomicCategoryGroup.create({
1144:                 data: {
1145:                     genomicAnalysisTableId: genomicTable.id,
1146:                     category: category.category || "",
1147:                 },
1148:             });
1149:
1150:             if (Array.isArray(category.subcategories)) {
1151:                 const subcategoriesData = category.subcategories.map(
1152:                     (sub: any) => ({
1153:                         genomicCategoryGroupId: categoryGroup.id,
1154:                         area: sub.area || "",
1155:                         trait: sub.trait || "",
1156:                         genes: sub.genes || [],
1157:                     })
1158:                 );
1159:
1160:                 await prisma.genomicSubcategory.createMany({
1161:                     data: subcategoriesData,
1162:                 });
1163:             }
1164:         }
1165:     }
1166: }
1167:
1168: // Handle family genetic impact section
1169: const familyGeneticImpactData = {
1170:     description:
1171:         report.familyGeneticImpactDescription ||
1172:         report.familyGeneticImpactSection?.description ||
1173:         "",
1174:     impacts:
1175:         report.familyGeneticImpacts ||

```

```

1176:         report.familyGeneticImpact ||
1177:         report.familyGeneticImpactSection?.impacts ||
1178:         [],
1179:     };
1180:
1181:     // Handle Family Genetic Impact Section in a transaction
1182:     await prisma.$transaction(async (tx) => {
1183:         // Update the description
1184:         await tx.report.update({
1185:             where: { id: report.id },
1186:             data: {
1187:                 familyGeneticImpactDescription:
1188:                     familyGeneticImpactData.description.trim(),
1189:             },
1190:         });
1191:
1192:         // Handle impacts if they exist
1193:         if (Array.isArray(familyGeneticImpactData.impacts)) {
1194:             const validImpacts: FamilyGeneticImpact[] =
1195:                 familyGeneticImpactData.impacts
1196:                     .filter(
1197:                         (imp: Partial<FamilyGeneticImpact>) =>
1198:                             typeof imp.gene === "string" &&
1199:                             imp.gene.trim() !== "" &&
1200:                             typeof imp.normalAlleles === "string" &&
1201:                             imp.normalAlleles.trim() !== "" &&
1202:                             typeof imp.yourResult === "string" &&
1203:                             imp.yourResult.trim() !== ""
1204:                     )
1205:                     .map((imp: Partial<FamilyGeneticImpact>) => ({
1206:                         id: imp.id || crypto.randomUUID(),
1207:                         reportId: report.id,
1208:                         gene: imp.gene!.trim(),
1209:                         normalAlleles: imp.normalAlleles!.trim(),
1210:                         yourResult: imp.yourResult!.trim(),
1211:                         healthImpact: imp.healthImpact?.trim() || "",
1212:                     }));
1213:
1214:             // Delete existing impacts first
1215:             await tx.familyGeneticImpact.deleteMany({
1216:                 where: { reportId: report.id },
1217:             });
1218:
1219:             // Create new impacts
1220:             if (validImpacts.length > 0) {
1221:                 await tx.familyGeneticImpact.createMany({
1222:                     data: validImpacts,
1223:                 });
1224:             }
1225:         }
1226:     });
1227: }
1228: }
1229: }
1230:
1231: return NextResponse.json({ message: "Patients saved successfully." });
1232: } catch (error: any) {
1233:     console.error("? POST error:", error);
1234:
1235:     return NextResponse.json(
1236:         {
1237:             error: error?.message || "Unknown error",
1238:             code: error?.code || null,
1239:             cause: error?.cause || null,
1240:             meta: error?.meta || null,
1241:             stack: error?.stack || null,
1242:             timestamp: new Date().toISOString(),
1243:         },
1244:         { status: 500 }
1245:     );
1246: } finally {
1247:     await prisma.$disconnect();
1248: }

```

```

1249: }
1250:
1251: export async function DELETE(req: Request) {
1252:   try {
1253:     const { searchParams } = new URL(req.url);
1254:     const reportId = searchParams.get("reportId");
1255:     const patientId = searchParams.get("patientId");
1256:
1257:     // Case 1: Delete a report
1258:     if (reportId) {
1259:       await prisma.report.delete({
1260:         where: { id: reportId },
1261:       });
1262:
1263:       return NextResponse.json({
1264:         success: true,
1265:         type: "report",
1266:         id: reportId,
1267:       });
1268:     }
1269:
1270:     // Case 2: Delete a patient (cascade removes reports)
1271:     if (patientId) {
1272:       await prisma.patient.delete({
1273:         where: { id: patientId },
1274:       });
1275:
1276:       return NextResponse.json({
1277:         success: true,
1278:         type: "patient",
1279:         id: patientId,
1280:       });
1281:     }
1282:
1283:     // No valid parameter found
1284:     return NextResponse.json(
1285:       { error: "Missing reportId or patientId" },
1286:       { status: 400 }
1287:     );
1288:   } catch (error) {
1289:     console.error("Error deleting:", error);
1290:     return NextResponse.json({ error: "Failed to delete" }, { status: 500 });
1291:   }
1292: }

```

## File: app\api\share-report\route.ts

```

1: import { NextRequest, NextResponse } from "next/server";
2: import { PrismaClient } from "@prisma/client";
3: import {
4:   createSharePayload,
5:   hashPassword,
6:   generateSecureToken,
7: } from "@/lib/encryption";
8:
9: const prisma = new PrismaClient();
10:
11: /**
12:  * POST - Create a new share token
13:  */
14: export async function POST(request: NextRequest) {
15:   try {
16:     const body = await request.json();
17:     const {
18:       reportId,
19:       patientId,
20:       password,
21:       expiresInDays,
22:       maxViews,
23:       createdBy,
24:     } = body;

```



```

25:
26: // Validate required fields
27: if (!reportId || !patientId) {
28:   return NextResponse.json(
29:     { error: "Missing reportId or patientId" },
30:     { status: 400 }
31:   );
32: }
33:
34: // Verify report and patient exist
35: const report = await prisma.report.findUnique({
36:   where: { id: reportId },
37:   include: { patient: true },
38: });
39:
40: if (!report || report.patientId !== patientId) {
41:   return NextResponse.json(
42:     { error: "Report not found or does not belong to patient" },
43:     { status: 404 }
44:   );
45: }
46:
47: // Create encrypted token
48: const token = createSharePayload(reportId, patientId);
49:
50: // Calculate expiration
51: let expiresAt: Date | null = null;
52: if (expiresInDays && expiresInDays > 0) {
53:   expiresAt = new Date();
54:   expiresAt.setDate(expiresAt.getDate() + expiresInDays);
55: }
56:
57: // Hash password if provided
58: let hashedPassword: string | null = null;
59: if (password && password.trim()) {
60:   hashedPassword = await hashPassword(password);
61: }
62:
63: // Create share token in database
64: const shareToken = await prisma.shareToken.create({
65:   data: {
66:     token,
67:     reportId,
68:     patientId,
69:     password: hashedPassword,
70:     expiresAt,
71:     maxViews: maxViews || null,
72:     createdBy: createdBy || null,
73:     isActive: true,
74:   },
75: });
76:
77: // Generate shareable URL
78: const baseUrl = process.env.NEXT_PUBLIC_APP_URL || "http://localhost:3000";
79: const shareUrl = `${baseUrl}/shared/${token}`;
80:
81: return NextResponse.json({
82:   success: true,
83:   shareToken: {
84:     id: shareToken.id,
85:     url: shareUrl,
86:     token: shareToken.token,
87:     expiresAt: shareToken.expiresAt,
88:     maxViews: shareToken.maxViews,
89:     hasPassword: !!hashedPassword,
90:     createdAt: shareToken.createdAt,
91:     message: hashedPassword
92:       ? "Anyone with the password can access this report unlimited times"
93:       : "Public link - anyone with the URL can access",
94:   },
95: });
96: } catch (error: any) {
97:   console.error("Error creating share token:", error);

```

```

98:     return NextResponse.json(
99:       { error: "Failed to create share link", details: error.message },
100:      { status: 500 }
101:    );
102:  } finally {
103:    await prisma.$disconnect();
104:  }
105: }
106:
107: /**
108:  * GET - List all share tokens for a report
109:  */
110: export async function GET(request: NextRequest) {
111:   try {
112:     const { searchParams } = new URL(request.url);
113:     const reportId = searchParams.get("reportId");
114:     const patientId = searchParams.get("patientId");
115:
116:     if (!reportId && !patientId) {
117:       return NextResponse.json(
118:         { error: "Missing reportId or patientId parameter" },
119:         { status: 400 }
120:       );
121:     }
122:
123:     const where: any = { isActive: true };
124:     if (reportId) where.reportId = reportId;
125:     if (patientId) where.patientId = patientId;
126:
127:     const shareTokens = await prisma.shareToken.findMany({
128:       where,
129:       orderBy: { createdAt: "desc" },
130:       select: {
131:         id: true,
132:         token: true,
133:         expiresAt: true,
134:         maxViews: true,
135:         viewCount: true,
136:         createdAt: true,
137:         lastAccessedAt: true,
138:         isActive: true,
139:         password: true, // Just to check if exists
140:         report: {
141:           select: {
142:             id: true,
143:             name: true,
144:           },
145:         },
146:       },
147:     });
148:
149:     // Format response
150:     const formatted = shareTokens.map((st) => ({
151:       id: st.id,
152:       token: st.token,
153:       url: `${
154:         process.env.NEXT_PUBLIC_APP_URL || "http://localhost:3000"
155:       }/shared/${st.token}`,
156:       expiresAt: st.expiresAt,
157:       maxViews: st.maxViews,
158:       viewCount: st.viewCount,
159:       hasPassword: !!st.password,
160:       createdAt: st.createdAt,
161:       lastAccessedAt: st.lastAccessedAt,
162:       isActive: st.isActive,
163:       reportName: st.report.name || "Untitled Report",
164:       isExpired: st.expiresAt ? new Date() > st.expiresAt : false,
165:       isMaxViewsReached: st.maxViews ? st.viewCount >= st.maxViews : false,
166:     }));
167:
168:     return NextResponse.json({
169:       success: true,
170:       shareTokens: formatted,

```

```

171:     });
172:   } catch (error: any) {
173:     console.error("Error fetching share tokens:", error);
174:     return NextResponse.json(
175:       { error: "Failed to fetch share tokens", details: error.message },
176:       { status: 500 }
177:     );
178:   } finally {
179:     await prisma.$disconnect();
180:   }
181: }
182:
183: /**
184:  * DELETE - Revoke a share token
185:  */
186: export async function DELETE(request: NextRequest) {
187:   try {
188:     const { searchParams } = new URL(request.url);
189:     const tokenId = searchParams.get("tokenId");
190:
191:     if (!tokenId) {
192:       return NextResponse.json(
193:         { error: "Missing tokenId parameter" },
194:         { status: 400 }
195:       );
196:     }
197:
198:     // Deactivate instead of delete to maintain audit trail
199:     await prisma.shareToken.update({
200:       where: { id: tokenId },
201:       data: { isActive: false },
202:     });
203:
204:     return NextResponse.json({
205:       success: true,
206:       message: "Share token revoked successfully",
207:     });
208:   } catch (error: any) {
209:     console.error("Error revoking share token:", error);
210:     return NextResponse.json(
211:       { error: "Failed to revoke share token", details: error.message },
212:       { status: 500 }
213:     );
214:   } finally {
215:     await prisma.$disconnect();
216:   }
217: }

```

## File: app\api\shared-access\route.ts

```

=====
1: import { NextRequest, NextResponse } from "next/server";
2: import { PrismaClient } from "@prisma/client";
3: import { parseSharePayload, verifyPassword } from "@lib/encryption";
4:
5: const prisma = new PrismaClient();
6:
7: /**
8:  * POST - Verify and access shared report
9:  */
10: export async function POST(request: NextRequest) {
11:   try {
12:     const body = await request.json();
13:     const { token, password } = body;
14:
15:     if (!token) {
16:       return NextResponse.json({ error: "Missing token" }, { status: 400 });
17:     }
18:
19:     // Find share token
20:     const shareToken = await prisma.shareToken.findUnique({
21:       where: { token },

```

```

22:     include: {
23:       report: {
24:         include: {
25:           dynamicDietFieldDefinitions: {
26:             include: {
27:               fields: true,
28:             },
29:           },
30:           patientDietAnalysis: true,
31:           nutritionData: true,
32:           lifestyleConditions: true,
33:           lifestyleCategoryImages: true,
34:           metabolicCoreData: true,
35:           digestiveHealthData: true,
36:           addictionData: true,
37:           sleepData: true,
38:           allergyData: true,
39:           geneTestResults: true,
40:           geneticCategories: true,
41:           preventiveTests: true,
42:           nutritionalSupplements: true,
43:           familyGeneticImpacts: true,
44:           healthSummaryEntries: true,
45:           genomicAnalysisTable: {
46:             include: {
47:               categories: {
48:                 include: {
49:                   subcategories: true,
50:                 },
51:               },
52:             },
53:           },
54:         },
55:       },
56:       patient: true,
57:     },
58:   });
59:
60:   if (!shareToken) {
61:     return NextResponse.json(
62:       { error: "Invalid or expired share link" },
63:       { status: 404 }
64:     );
65:   }
66:
67:   // Check if token is active
68:   if (!shareToken.isActive) {
69:     return NextResponse.json(
70:       { error: "This share link has been revoked" },
71:       { status: 403 }
72:     );
73:   }
74:
75:   // Check expiration
76:   if (shareToken.expiresAt && new Date() > shareToken.expiresAt) {
77:     return NextResponse.json(
78:       { error: "This share link has expired" },
79:       { status: 403 }
80:     );
81:   }
82:
83:   // Check password if required
84:   if (shareToken.password) {
85:     if (!password) {
86:       return NextResponse.json(
87:         { error: "Password required", requiresPassword: true },
88:         { status: 401 }
89:       );
90:     }
91:
92:     const isValidPassword = await verifyPassword(
93:       password,
94:       shareToken.password

```

```

95:         );
96:         if (!isValidPassword) {
97:             return NextResponse.json(
98:                 { error: "Invalid password", requiresPassword: true },
99:                 { status: 401 }
100:            );
101:        }
102:    }
103:
104:    // Simple view count increment (counts every access)
105:    // No unique viewer tracking - anyone with password can access unlimited times
106:    await prisma.shareToken.update({
107:        where: { id: shareToken.id },
108:        data: {
109:            viewCount: { increment: 1 },
110:            lastAccessedAt: new Date(),
111:        },
112:    });
113:
114:    // Transform report data (similar to GET /api/patients-data)
115:    const report = shareToken.report;
116:    const patient = shareToken.patient;
117:
118:    const transformedReport = {
119:        id: report.id,
120:        name: report.name,
121:        content: {
122:            introduction: report.introduction || "",
123:            genomicsExplanation: report.genomicsExplanation || "",
124:            genesHealthImpact: report.genesHealthImpact || "",
125:            fundamentalsPRS: report.fundamentalsPRS || "",
126:            utilityDoctors: report.utilityDoctors || "",
127:            microarrayExplanation: report.microarrayExplanation || "",
128:            microarrayData: report.microarrayData || "",
129:        },
130:        settings: {
131:            title: report.title || "",
132:            subtitle: report.subtitle || "",
133:            companyName: report.companyName || "",
134:            headerColor: report.headerColor || "",
135:            accentColor: report.accentColor || "",
136:            fonts: {
137:                primary: report.primaryFont || "",
138:                secondary: report.secondaryFont || "",
139:                mono: report.monoFont || "",
140:            },
141:        },
142:        patientInfo: {
143:            name: patient.name || "",
144:            gender: patient.gender || "",
145:            birthDate: patient.birthDate || "",
146:            sampleCode: patient.sampleCode || "",
147:            sampleDate: patient.sampleDate || "",
148:            reportDate: patient.reportDate || "",
149:            checkedBy: patient.checkedBy || "",
150:            scientificContent: patient.scientificContent || "",
151:            disclaimer: patient.disclaimer || "",
152:            signature1: patient.signature1 || null,
153:            signature2: patient.signature2 || null,
154:        },
155:        // Add all other fields following the same pattern as patients-data route
156:        dynamicDietFieldDefinitions: (
157:            report.dynamicDietFieldDefinitions || []
158:        ).map((def: any) => ({
159:            meta: {
160:                quote: def.quote || "",
161:                description: def.description || "",
162:            },
163:            fields: (def.fields || []).map((field: any) => ({
164:                id: field.fieldId || field.id || "",
165:                label: field.label || "",
166:                category: field.category || "",
167:                min: field.min || 0,

```

```

168:         max: field.max || 100,
169:         highRecommendation: field.highRecommendation || "",
170:         normalRecommendation: field.normalRecommendation || "",
171:         lowRecommendation: field.lowRecommendation || "",
172:     })),
173:   })),
174:   // Continue with other transformations...
175:   // (Use the same transformation logic from your patients-data GET route)
176: };
177:
178: return NextResponse.json({
179:   success: true,
180:   report: transformedReport,
181:   shareInfo: {
182:     viewCount: shareToken.viewCount + 1,
183:     maxViews: shareToken.maxViews,
184:     expiresAt: shareToken.expiresAt,
185:     isReadOnly: true,
186:   },
187: });
188: } catch (error: any) {
189:   console.error("Error accessing shared report:", error);
190:   return NextResponse.json(
191:     { error: "Failed to access shared report", details: error.message },
192:     { status: 500 }
193:   );
194: } finally {
195:   await prisma.$disconnect();
196: }
197: }

```

## ■ File: app\api\sleep-image\route.ts

```

=====
1: import { promises as fs } from "fs";
2: import path from "path";
3: import { NextRequest, NextResponse } from "next/server";
4:
5: const folderName = "sleep";
6: const folderPath = path.join(process.cwd(), "public", folderName);
7:
8: // GET: List all sleep images
9: export async function GET() {
10:   try {
11:     const files = await fs.readdir(folderPath);
12:
13:     const images = files.map((file) => ({
14:       label: file.split(".")[0], // This will now be your custom label
15:       url: `/${folderName}/${file}`,
16:     }));
17:
18:     return NextResponse.json({ success: true, images });
19:   } catch (err) {
20:     console.error("Failed to read sleep image folder:", err);
21:     return NextResponse.json(
22:       { success: false, error: "Could not read images" },
23:       { status: 500 }
24:     );
25:   }
26: }
27:
28: // POST: Upload a new image to /public/sleep
29: export async function POST(req: NextRequest) {
30:   try {
31:     const formData = await req.formData();
32:     const file = formData.get("file") as File;
33:     const customLabel = formData.get("label") as string; // Get the custom label
34:
35:     if (!file) {
36:       return NextResponse.json(
37:         { success: false, error: "Missing file" },
38:         { status: 400 }

```

```

39:     });
40:   }
41:
42:   const arrayBuffer = await file.arrayBuffer();
43:   const buffer = Buffer.from(arrayBuffer);
44:
45:   // Use custom label if provided, otherwise use original filename
46:   const fileExtension = path.extname(file.name);
47:   const fileName = customLabel
48:     ? `${customLabel.replace(/\s+/g, "_")}${fileExtension}`
49:     : file.name.replace(/\s+/g, "_");
50:
51:   const filePath = path.join(folderPath, fileName);
52:
53:   await fs.mkdir(path.dirname(filePath), { recursive: true });
54:   await fs.writeFile(filePath, buffer);
55:
56:   return NextResponse.json({
57:     success: true,
58:     label: customLabel || file.name.split(".")[0], // Return the label used
59:     url: `/${folderName}/${fileName}`,
60:   });
61: } catch (err) {
62:   console.error("Upload failed:", err);
63:   return NextResponse.json(
64:     { success: false, error: "Upload failed" },
65:     { status: 500 }
66:   );
67: }
68: }
69:
70: // DELETE: Remove an image from /public/sleep
71: export async function DELETE(req: NextRequest) {
72:   try {
73:     const { searchParams } = new URL(req.url);
74:     const file = searchParams.get("file");
75:
76:     if (!file) {
77:       return NextResponse.json(
78:         { success: false, error: "Missing file parameter" },
79:         { status: 400 }
80:       );
81:     }
82:
83:     const filePath = path.join(folderPath, file);
84:     await fs.unlink(filePath);
85:
86:     return NextResponse.json({ success: true, message: "Image deleted" });
87:   } catch (err) {
88:     console.error("Deletion failed:", err);
89:     return NextResponse.json(
90:       { success: false, error: "Deletion failed" },
91:       { status: 500 }
92:     );
93:   }
94: }

```

## ■ File: `app\api\sports-images\route.ts`

```

=====
1: import { writeFile, readdir } from "fs/promises";
2: import path from "path";
3: import { NextRequest, NextResponse } from "next/server";
4: import { unlink } from "fs/promises";
5: import { existsSync } from "fs";
6:
7: // POST: Upload image to /public/sports
8: export async function POST(req: NextRequest) {
9:   const formData = await req.formData();
10:  const file = formData.get("file") as File;
11:  const label = formData.get("label") as string;
12:

```

```

13:   if (!file || !label) {
14:     return NextResponse.json(
15:       { success: false, error: "Missing file or label" },
16:       { status: 400 }
17:     );
18:   }
19:
20:   const ext = file.name.split(".").pop() || "png";
21:   const safeLabel = label
22:     .toLowerCase()
23:     .replace(/\\s+/g, "_")
24:     .replace(/[^\a-z0-9_]/gi, "");
25:   const fileName = `${safeLabel}.${ext}`;
26:   const filePath = path.join(process.cwd(), "public", "sports", fileName);
27:
28:   try {
29:     const bytes = await file.arrayBuffer();
30:     const buffer = Buffer.from(bytes);
31:     await writeFile(filePath, buffer);
32:
33:     const publicUrl = `/sports/${fileName}`;
34:     return NextResponse.json({ success: true, url: publicUrl });
35:   } catch (error) {
36:     return NextResponse.json(
37:       { success: false, error: "Failed to save file" },
38:       { status: 500 }
39:     );
40:   }
41: }
42:
43: //delete image
44:
45: export async function DELETE(req: NextRequest) {
46:   try {
47:     const { searchParams } = new URL(req.url);
48:     const fileName = searchParams.get("file");
49:
50:     if (!fileName) {
51:       return NextResponse.json(
52:         { success: false, error: "Missing file parameter" },
53:         { status: 400 }
54:       );
55:     }
56:
57:     const filePath = path.join(process.cwd(), "public", "sports", fileName);
58:
59:     if (existsSync(filePath)) {
60:       await unlink(filePath);
61:       return NextResponse.json({ success: true, deleted: fileName });
62:     } else {
63:       return NextResponse.json(
64:         { success: false, error: "File does not exist" },
65:         { status: 404 }
66:       );
67:     }
68:   } catch (error) {
69:     console.error("Error deleting sports image:", error);
70:     return NextResponse.json(
71:       { success: false, error: "Failed to delete image" },
72:       { status: 500 }
73:     );
74:   }
75: }
76: // GET: Return list of images in /public/sports
77: export async function GET() {
78:   try {
79:     const dirPath = path.join(process.cwd(), "public", "sports");
80:     const files = await readdir(dirPath);
81:
82:     const images = files.map((file) => ({
83:       label: path.parse(file).name.toLowerCase(), // "dumbbell"
84:       url: `/sports/${file}`,
85:     }));

```



```

86:
87:     return NextResponse.json({ success: true, images });
88:   } catch (error) {
89:     console.error("Failed to read sports directory:", error);
90:     return NextResponse.json(
91:       { success: false, error: "Could not read sports folder" },
92:       { status: 500 }
93:     );
94:   }
95: }

```

## ■ File: app\api\upload-image\route.ts

```

1: import { writeFile } from "fs/promises";
2: import path from "path";
3: import { NextRequest, NextResponse } from "next/server";
4:
5: export async function POST(req: NextRequest) {
6:   const formData = await req.formData();
7:   const file = formData.get("file") as File;
8:
9:   if (!file) {
10:    return NextResponse.json(
11:      { success: false, error: "No file uploaded" },
12:      { status: 400 }
13:    );
14:   }
15:
16:   const bytes = await file.arrayBuffer();
17:   const buffer = Buffer.from(bytes);
18:
19:   const fileName = `${Date.now()}-${file.name}`;
20:   const filePath = path.join(process.cwd(), "public", "table", fileName);
21:   try {
22:     await writeFile(filePath, buffer);
23:     const publicUrl = `/table/${fileName}`;
24:     return NextResponse.json({ success: true, url: publicUrl });
25:   } catch (error) {
26:     return NextResponse.json(
27:       { success: false, error: "Failed to save file" },
28:       { status: 500 }
29:     );
30:   }
31: }

```

## ■ File: app\api\upload-signature\route.ts

```

1: import { NextRequest, NextResponse } from "next/server";
2: import path from "path";
3: import fs from "fs/promises";
4: import { randomUUID } from "crypto";
5:
6: export async function POST(req: NextRequest) {
7:   const formData = await req.formData();
8:   const file = formData.get("file") as File | null;
9:   if (!file)
10:    return NextResponse.json({ error: "No file uploaded" }, { status: 400 });
11:
12:   const bytes = await file.arrayBuffer();
13:   const buffer = Buffer.from(bytes);
14:
15:   const fileName = `${randomUUID()}-${file.name}`;
16:   const uploadDir = path.join(process.cwd(), "public", "uploads", "signatures");
17:   await fs.mkdir(uploadDir, { recursive: true });
18:   await fs.writeFile(path.join(uploadDir, fileName), buffer);
19:
20:   const fileUrl = `/uploads/signatures/${fileName}`;
21:   return NextResponse.json({ url: fileUrl });
22: }

```

## ■ File: app\layout.tsx

```
=====
1: import type React from "react";
2: import type { Metadata } from "next";
3: import { Inter } from "next/font/google";
4: import "./globals.css";
5: import { ThemeProvider } from "@components/theme-provider";
6: import { Toaster } from "@components/ui/toaster";
7:
8: const inter = Inter({ subsets: ["latin"] });
9:
10: export const metadata: Metadata = {
11:   title: "Nutrigenomics Report App",
12:   description: "Comprehensive Nutrigenomics Test Report Application",
13:   generator: "",
14:   icons: {
15:     icon: "/favicon.jpg",
16:   },
17: };
18:
19: export default function RootLayout({
20:   children,
21: }: Readonly<{
22:   children: React.ReactNode;
23: }>) {
24:   return (
25:     <html lang="en" suppressHydrationWarning>
26:       <body className={inter.className}>
27:         <ThemeProvider
28:           attribute="class"
29:           defaultTheme="system"
30:           enableSystem
31:           disableTransitionOnChange
32:         >
33:           {children}
34:           <Toaster />
35:         </ThemeProvider>
36:       </body>
37:     </html>
38:   );
39: }
```

## ■ File: app\page.tsx

```
=====
1: import { CardDescription } from "@components/ui/card";
2: import Link from "next/link";
3: import { Button } from "@components/ui/button";
4: import { Card, CardContent, CardHeader, CardTitle } from "@components/ui/card";
5: import {
6:   Settings,
7:   FileText,
8:   Eye,
9:   Download,
10:   Database,
11:   Dna,
12:   Activity,
13:   Heart,
14:   Brain,
15:   Utensils,
16:   Zap,
17:   AlertCircle,
18:   Users,
19: } from "lucide-react";
20:
21: export default function HomePage() {
22:   return (
23:     <div className="flex flex-col items-center w-full justify-center min-h-screen bg-gradient-to-br from-gr
24:       <div className="w-200 items-center grid gap-6 my-12">
25:         <Link href="/admin?tab=patient-info">
26:           <Card className="hover:shadow-xl transition-all duration-300 border-0 bg-gradient-to-br from-blue
27:
```

```

28:         <CardContent className="p-6 text-center">
29:             { /* <Settings className="h-12 w-12 mx-auto mb-4" /> */}
30:             <h3 className="font-bold text-lg">Report Generator</h3>
31:             { /* <p className="text-blue-100 text-sm">Manage all data</p> */}
32:         </CardContent>
33:     </Card>
34: </Link>
35: </div>
36:
37: { /* Report Sections */}
38: <Card className="w-full shadow-lg border-0 bg-white mb-12">
39:     <CardHeader className="bg-gradient-to-r from-indigo-500 to-purple-600 text-white rounded-lg">
40:         <CardTitle className="text-2xl">Report Sections Included</CardTitle>
41:     </CardHeader>
42:     <CardContent className="p-8">
43:         <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-6">
44:             <div className="flex items-center gap-3 p-4 bg-red-50 rounded-lg">
45:                 <Utensils className="h-8 w-8 text-red-600" />
46:                 <div>
47:                     <h4 className="font-semibold text-red-800">Diet Analysis</h4>
48:                     <p className="text-sm text-red-600">
49:                         Macronutrients, sensitivities, meal patterns
50:                     </p>
51:                 </div>
52:             </div>
53:
54:             <div className="flex items-center gap-3 p-4 bg-blue-50 rounded-lg">
55:                 <Activity className="h-8 w-8 text-blue-600" />
56:                 <div>
57:                     <h4 className="font-semibold text-blue-800">Nutrition</h4>
58:                     <p className="text-sm text-blue-600">
59:                         Vitamins, minerals, supplements
60:                     </p>
61:                 </div>
62:             </div>
63:
64:             <div className="flex items-center gap-3 p-4 bg-green-50 rounded-lg">
65:                 <Zap className="h-8 w-8 text-green-600" />
66:                 <div>
67:                     <h4 className="font-semibold text-green-800">
68:                         Sports & Fitness
69:                     </h4>
70:                     <p className="text-sm text-green-600">
71:                         Exercise types, performance factors
72:                     </p>
73:                 </div>
74:             </div>
75:
76:             <div className="flex items-center gap-3 p-4 bg-yellow-50 rounded-lg">
77:                 <Heart className="h-8 w-8 text-yellow-600" />
78:                 <div>
79:                     <h4 className="font-semibold text-yellow-800">
80:                         Lifestyle Conditions
81:                     </h4>
82:                     <p className="text-sm text-yellow-600">
83:                         Health risks and strengths
84:                     </p>
85:                 </div>
86:             </div>
87:
88:             <div className="flex items-center gap-3 p-4 bg-purple-50 rounded-lg">
89:                 <Dna className="h-8 w-8 text-purple-600" />
90:                 <div>
91:                     <h4 className="font-semibold text-purple-800">
92:                         Metabolic Core
93:                     </h4>
94:                     <p className="text-sm text-purple-600">
95:                         Gene analysis and impacts
96:                     </p>
97:                 </div>
98:             </div>
99:
100:             <div className="flex items-center gap-3 p-4 bg-indigo-50 rounded-lg">

```

```

101:         <Brain className="h-8 w-8 text-indigo-600" />
102:     </div>
103:         <h4 className="font-semibold text-indigo-800">
104:             Digestive Health
105:         </h4>
106:         <p className="text-sm text-indigo-600">
107:             Gut health, intolerances
108:         </p>
109:     </div>
110: </div>
111:
112: <div className="flex items-center gap-3 p-4 bg-pink-50 rounded-lg">
113:     <Dna className="h-8 w-8 text-pink-600" />
114:     <div>
115:         <h4 className="font-semibold text-pink-800">
116:             Genes & Addiction
117:         </h4>
118:         <p className="text-sm text-pink-600">Tendencies, advice</p>
119:     </div>
120: </div>
121:
122: <div className="flex items-center gap-3 p-4 bg-teal-50 rounded-lg">
123:     <Brain className="h-8 w-8 text-teal-600" />
124:     <div>
125:         <h4 className="font-semibold text-teal-800">Sleep & Rest</h4>
126:         <p className="text-sm text-teal-600">Sleep patterns, quality</p>
127:     </div>
128: </div>
129:
130: <div className="flex items-center gap-3 p-4 bg-cyan-50 rounded-lg">
131:     <AlertCircle className="h-8 w-8 text-cyan-600" />
132:     <div>
133:         <h4 className="font-semibold text-cyan-800">
134:             Allergies & Sensitivity
135:         </h4>
136:         <p className="text-sm text-cyan-600">
137:             Environmental sensitivities
138:         </p>
139:     </div>
140: </div>
141:
142: <div className="flex items-center gap-3 p-4 bg-lime-50 rounded-lg">
143:     <Heart className="h-8 w-8 text-lime-600" />
144:     <div>
145:         <h4 className="font-semibold text-lime-800">
146:             Preventive Health
147:         </h4>
148:         <p className="text-sm text-lime-600">
149:             Diagnostic tests, supplements
150:         </p>
151:     </div>
152: </div>
153:
154: <div className="flex items-center gap-3 p-4 bg-amber-50 rounded-lg">
155:     <Users className="h-8 w-8 text-amber-600" />
156:     <div>
157:         <h4 className="font-semibold text-amber-800">
158:             Family Genetic Impact
159:         </h4>
160:         <p className="text-sm text-amber-600">
161:             Inherited traits, family recommendations
162:         </p>
163:     </div>
164: </div>
165: </div>
166: </CardContent>
167: </Card>
168: </div>
169: );
170: }

```

---

## ■ File: app\report\[id]\page.tsx

```
=====
1: import { notFound } from "next/navigation";
2: import ComprehensiveReportViewer from "@/components/comprehensive-report-viewer";
3: import type { ComprehensiveReportData } from "@/types/report-types";
4:
5: interface ReportPageProps {
6:   params: {
7:     id: string;
8:   };
9: }
10:
11: async function getReportData(
12:   id: string
13: ): Promise<ComprehensiveReportData | null> {
14:   try {
15:     const response = await fetch(
16:       `${
17:         process.env.NEXT_PUBLIC_APP_URL || "http://localhost:3000"
18:       }/api/patients-data`,
19:       { cache: "no-store" }
20:     );
21:
22:     if (!response.ok) {
23:       console.error(
24:         `Failed to fetch report data: ${response.status} ${response.statusText}`
25:       );
26:       return null;
27:     }
28:
29:     const data = await response.json();
30:     return data as ComprehensiveReportData;
31:   } catch (error) {
32:     console.error("Error fetching report data:", error);
33:     return null;
34:   }
35: }
36:
37: export default async function ReportPage({ params }: ReportPageProps) {
38:   const reportData = await getReportData(params.id);
39:
40:   if (!reportData) {
41:     notFound();
42:   }
43:
44:   return (
45:     <div className="min-h-screen bg-gray-100 print:bg-white">
46:       <ComprehensiveReportViewer reportData={reportData} />
47:     </div>
48:   );
49: }
```

## ■ File: app\report\pdf\page.tsx

```
=====
1: import ComprehensiveReportViewer from "@/components/comprehensive-report-viewer";
2: import type { ComprehensiveReportData } from "@/types/report-types";
3:
4: async function getReportData(): Promise<ComprehensiveReportData | null> {
5:   try {
6:     const response = await fetch(
7:       `${
8:         process.env.NEXT_PUBLIC_APP_URL || "http://localhost:3000"
9:       }/api/patients-data`,
10:      { cache: "no-store" }
11:    );
12:
13:    if (!response.ok) {
14:      console.error(
15:        `Failed to fetch report data: ${response.status} ${response.statusText}`
16:      );
17:      return null;

```

```

18:     }
19:
20:     const data = await response.json();
21:     return data as ComprehensiveReportData;
22:   } catch (error) {
23:     console.error("Error fetching report data:", error);
24:     return null;
25:   }
26: }
27:
28: export default async function ReportPdfPage() {
29:   const reportData = await getReportData();
30:
31:   if (!reportData) {
32:     return (
33:       <div className="text-center py-10 text-red-500">
34:         Failed to load report data for PDF generation.
35:       </div>
36:     );
37:   }
38:
39:   return (
40:     <div className="min-h-screen bg-white print:bg-white">
41:       <ComprehensiveReportViewer reportData={reportData} />
42:     </div>
43:   );
44: }

```

## ■ File: app/report/preview/page.tsx

```

1: "use client";
2:
3: import { useEffect, useState } from "react";
4: import { useSearchParams } from "next/navigation";
5: import ComprehensiveReportViewer from "@components/comprehensive-report-viewer";
6: import type { ComprehensiveReportData } from "@types/report-types";
7:
8: export default function ReportPreviewPage() {
9:   const searchParams = useSearchParams();
10:  const [reportData, setReportData] = useState<ComprehensiveReportData | null>(
11:    null
12:  );
13:  const [error, setError] = useState<string | null>(null);
14:
15:  useEffect(() => {
16:    const patientCode = searchParams.get("patient");
17:    const reportIndexStr = searchParams.get("report");
18:
19:    if (!patientCode || !reportIndexStr) {
20:      setError("Missing 'patient' or 'report' query parameters.");
21:      return;
22:    }
23:
24:    const reportIndex = parseInt(reportIndexStr);
25:    if (isNaN(reportIndex)) {
26:      setError("Invalid report index.");
27:      return;
28:    }
29:
30:    const fetchData = async () => {
31:      try {
32:        const res = await fetch("/api/patients-data", { cache: "no-store" });
33:        if (!res.ok) throw new Error(`API error ${res.status}`);
34:        const patients = await res.json();
35:
36:        const patient = patients.find(
37:          (p: any) => p.info?.sampleCode === patientCode
38:        );
39:        if (!patient) {
40:          setError(`No patient found with sampleCode "${patientCode}".`);
41:          return;

```

```

42:     }
43:
44:     const report = patient.reports?.[reportIndex];
45:     if (!report) {
46:         setError(
47:             `No report found at index ${reportIndex} for patient "${patientCode}".`
48:         );
49:         return;
50:     }
51:
52:     setReportData({
53:         ...report,
54:         patientInfo: patient.info,
55:     });
56:     } catch (err) {
57:         console.error("Error fetching report:", err);
58:         setError("Failed to load report data.");
59:     }
60: };
61:
62:     fetchData();
63: }, [searchParams]);
64:
65: if (error) {
66:     return (
67:         <div className="text-center text-red-500 py-10 font-semibold">
68:             ?? {error}
69:         </div>
70:     );
71: }
72:
73: if (!reportData) {
74:     return (
75:         <div className="text-center text-gray-500 py-10">Loading report...</div>
76:     );
77: }
78:
79: return (
80:     <div className="min-h-screen bg-gray-100 print:bg-white">
81:         <ComprehensiveReportViewer reportData={reportData} />
82:     </div>
83: );
84: }

```

## File: app\shared\[token]\page.tsx

```

1: "use client";
2:
3: import { useEffect, useState } from 'react';
4: import { useParams, useRouter } from 'next/navigation';
5: import Head from 'next/head';
6: import ComprehensiveReportViewer from '@components/comprehensive-report-viewer';
7: import { Input } from '@components/ui/input';
8: import { Button } from '@components/ui/button';
9: import { Card, CardContent, CardHeader, CardTitle } from '@components/ui/card';
10: import { Lock, Eye, Calendar, AlertCircle } from 'lucide-react';
11: import type { ComprehensiveReportData } from '@types/report-types';
12:
13: export default function SharedReportPage() {
14:     const params = useParams();
15:     const router = useRouter();
16:     const token = params.token as string;
17:
18:     const [reportData, setReportData] = useState<ComprehensiveReportData | null>(null);
19:     const [requiresPassword, setRequiresPassword] = useState(false);
20:     const [password, setPassword] = useState('');
21:     const [error, setError] = useState<string | null>(null);
22:     const [loading, setLoading] = useState(true);
23:     const [shareInfo, setShareInfo] = useState<any>(null);
24:
25:     // Force desktop mode on mobile

```

```

26: useEffect(() => {
27:   // Set viewport meta tag for desktop mode
28:   let viewportMeta = document.querySelector('meta[name="viewport"]');
29:   if (!viewportMeta) {
30:     viewportMeta = document.createElement('meta');
31:     viewportMeta.setAttribute('name', 'viewport');
32:     document.head.appendChild(viewportMeta);
33:   }
34:   viewportMeta.setAttribute('content', 'width=1280, initial-scale=1, maximum-scale=1, user-scalable=yes');
35: }, []);
36:
37: const fetchReport = async (pwd?: string) => {
38:   setLoading(true);
39:   setError(null);
40:
41:   try {
42:     const response = await fetch('/api/shared-access', {
43:       method: 'POST',
44:       headers: { 'Content-Type': 'application/json' },
45:       body: JSON.stringify({
46:         token,
47:         password: pwd || password
48:       })
49:     });
50:
51:     const data = await response.json();
52:
53:     if (response.status === 401 && data.requiresPassword) {
54:       setRequiresPassword(true);
55:       setLoading(false);
56:       return;
57:     }
58:
59:     if (!response.ok) {
60:       setError(data.error || 'Failed to load report');
61:       setLoading(false);
62:       return;
63:     }
64:
65:     setReportData(data.report);
66:     setShareInfo(data.shareInfo);
67:     setRequiresPassword(false);
68:     setLoading(false);
69:   } catch (err: any) {
70:     setError('Network error. Please try again.');
```

```

71:     setLoading(false);
72:   }
73: };
74:
75: useEffect(() => {
76:   if (token) {
77:     fetchReport();
78:   }
79: }, [token]);
80:
81: const handlePasswordSubmit = (e: React.FormEvent) => {
82:   e.preventDefault();
83:   if (password.trim()) {
84:     fetchReport(password);
85:   }
86: };
87:
88: // Loading state
89: if (loading) {
90:   return (
91:     <div className="min-h-screen flex items-center justify-center bg-gray-50">
92:       <div className="text-center">
93:         <div className="inline-block animate-spin rounded-full h-12 w-12 border-b-2 border-blue-600 mb-4">
94:           <p className="text-gray-600 font-medium">Loading shared report...</p>
95:         </div>
96:       </div>
97:     );
98:   }

```



```

99:
100: // Password required
101: if (requiresPassword) {
102:   return (
103:     <div className="min-h-screen flex items-center justify-center bg-gradient-to-br from-blue-50 to-indigo-50">
104:       <Card className="w-full max-w-md shadow-xl">
105:         <CardHeader className="text-center pb-4">
106:           <div className="mx-auto w-16 h-16 bg-blue-100 rounded-full flex items-center justify-center mb-4">
107:             <Lock className="w-8 h-8 text-blue-600" />
108:           </div>
109:           <CardTitle className="text-2xl font-bold text-gray-800">
110:             Password Protected
111:           </CardTitle>
112:           <p className="text-sm text-gray-600 mt-2">
113:             This report is password protected. Please enter the password to continue.
114:           </p>
115:         </CardHeader>
116:         <CardContent>
117:           <form onSubmit={handlePasswordSubmit} className="space-y-4">
118:             <div>
119:               <Input
120:                 type="password"
121:                 placeholder="Enter password"
122:                 value={password}
123:                 onChange={(e) => setPassword(e.target.value)}
124:                 className="w-full"
125:                 autoFocus
126:               />
127:             </div>
128:             {error && (
129:               <div className="bg-red-50 border border-red-200 text-red-700 px-4 py-3 rounded flex items-s">
130:                 <AlertCircle className="w-5 h-5 flex-shrink-0 mt-0.5" />
131:                 <span className="text-sm">{error}</span>
132:               </div>
133:             )}
134:             <Button type="submit" className="w-full" disabled={!password.trim()}>
135:               Access Report
136:             </Button>
137:           </form>
138:         </CardContent>
139:       </Card>
140:     </div>
141:   );
142: }
143:
144: // Error state
145: if (error && !requiresPassword) {
146:   return (
147:     <div className="min-h-screen flex items-center justify-center bg-gray-50 p-4">
148:       <Card className="w-full max-w-md shadow-lg">
149:         <CardHeader className="text-center pb-4">
150:           <div className="mx-auto w-16 h-16 bg-red-100 rounded-full flex items-center justify-center mb-4">
151:             <AlertCircle className="w-8 h-8 text-red-600" />
152:           </div>
153:           <CardTitle className="text-xl font-bold text-gray-800">
154:             Access Denied
155:           </CardTitle>
156:         </CardHeader>
157:         <CardContent className="text-center">
158:           <p className="text-gray-600 mb-4">{error}</p>
159:           <Button onClick={() => router.push('/')}> variant="outline">
160:             Go to Home
161:           </Button>
162:         </CardContent>
163:       </Card>
164:     </div>
165:   );
166: }
167:
168: // Report viewer with info banner
169: return (
170:   <div className="min-h-screen bg-gray-100">
171:     { /* Info Banner */ }

```

```

172:         <div className="bg-blue-600 text-white py-3 px-4 print:hidden">
173:             <div className="max-w-7xl mx-auto flex flex-wrap items-center justify-between gap-4">
174:                 <div className="flex items-center gap-2">
175:                     <Eye className="w-5 h-5" />
176:                     <span className="font-medium">Read-Only Shared Report</span>
177:                 </div>
178:                 {shareInfo && (
179:                     <div className="flex flex-wrap items-center gap-4 text-sm">
180:                         <div className="flex items-center gap-1">
181:                             <Eye className="w-5 h-5" />
182:                             <span>Total Views: {shareInfo.viewCount}</span>
183:                         </div>
184:                         {shareInfo.expiresAt && (
185:                             <div className="flex items-center gap-1">
186:                                 <Calendar className="w-4 h-4" />
187:                                 <span>
188:                                     Expires: {new Date(shareInfo.expiresAt).toLocaleDateString()}
189:                                 </span>
190:                             </div>
191:                         )}
192:                     </div>
193:                 )}
194:             </div>
195:         </div>
196:
197:         {/* Report Content */}
198:         {reportData && <ComprehensiveReportViewer reportData={reportData} />}
199:     </div>
200: );
201: }

```

## ■ File: components\admin\GenomicAnalysisAdmin.tsx

```

=====
1: "use client";
2:
3: import { Button } from "@/components/ui/button";
4: import { Card, CardHeader, CardTitle, CardContent } from "@/components/ui/card";
5: import { Input } from "@/components/ui/input";
6: import { Textarea } from "@/components/ui/textarea";
7: import { Plus, Trash2 } from "lucide-react";
8: import type {
9:     GenomicAnalysisTable,
10:     GenomicCategoryGroup,
11: } from "@/types/report-types";
12:
13: interface Props {
14:     genomicAnalysisTable: GenomicAnalysisTable;
15:     setGenomicAnalysisTable: (table: GenomicAnalysisTable) => void;
16:     onSave: () => void;
17:     onReset: () => void;
18: }
19:
20: export default function GenomicAnalysisAdmin({
21:     genomicAnalysisTable,
22:     setGenomicAnalysisTable,
23:     onSave,
24:     onReset,
25: }: Props) {
26:     const updateDescription = (value: string) => {
27:         setGenomicAnalysisTable({
28:             ...genomicAnalysisTable,
29:             description: value,
30:         });
31:     };
32:
33:     const addCategory = () => {
34:         const newCategory: GenomicCategoryGroup = {
35:             category: "New Category",
36:             subcategories: [],
37:         };
38:         setGenomicAnalysisTable({

```

```

39:     ...genomicAnalysisTable,
40:     categories: [...genomicAnalysisTable.categories, newCategory],
41:   });
42: };
43:
44: const removeCategory = (catIndex: number) => {
45:   setGenomicAnalysisTable({
46:     ...genomicAnalysisTable,
47:     categories: genomicAnalysisTable.categories.filter(
48:       (_, i) => i !== catIndex
49:     ),
50:   });
51: };
52:
53: const updateCategoryName = (catIndex: number, value: string) => {
54:   const updated = [...genomicAnalysisTable.categories];
55:   updated[catIndex] = { ...updated[catIndex], category: value };
56:   setGenomicAnalysisTable({ ...genomicAnalysisTable, categories: updated });
57: };
58:
59: const addSubcategory = (catIndex: number) => {
60:   const updated = [...genomicAnalysisTable.categories];
61:   const newSub = { area: "", trait: "", genes: [] };
62:   updated[catIndex] = {
63:     ...updated[catIndex],
64:     subcategories: [...updated[catIndex].subcategories, newSub],
65:   };
66:   setGenomicAnalysisTable({ ...genomicAnalysisTable, categories: updated });
67: };
68:
69: const removeSubcategory = (catIndex: number, subIndex: number) => {
70:   const updated = [...genomicAnalysisTable.categories];
71:   updated[catIndex] = {
72:     ...updated[catIndex],
73:     subcategories: updated[catIndex].subcategories.filter(
74:       (_, i) => i !== subIndex
75:     ),
76:   };
77:   setGenomicAnalysisTable({ ...genomicAnalysisTable, categories: updated });
78: };
79:
80: const updateSubcategory = (
81:   catIndex: number,
82:   subIndex: number,
83:   field: "area" | "trait" | "genes",
84:   value: string | string[]
85: ) => {
86:   const updated = [...genomicAnalysisTable.categories];
87:   const subs = [...updated[catIndex].subcategories];
88:   subs[subIndex] = { ...subs[subIndex], [field]: value };
89:   updated[catIndex] = { ...updated[catIndex], subcategories: subs };
90:   setGenomicAnalysisTable({ ...genomicAnalysisTable, categories: updated });
91: };
92:
93: return (
94:   <Card className="shadow-lg border-0 bg-white w-full">
95:     <CardHeader className="bg-gradient-to-r from-purple-500 to-pink-500 text-white rounded-lg p-4 sm:p-5">
96:       <CardTitle className="text-lg sm:text-xl md:text-2xl font-bold uppercase flex items-center justify-center">
97:         Genomic Analysis Table
98:       </CardTitle>
99:     </CardHeader>
100:
101:     <CardContent className="space-y-6 p-4">
102:       { /* Action buttons */ }
103:       <div className="flex justify-end gap-3 items-center">
104:         <Button
105:           onClick={addCategory}
106:           className="bg-purple-600 hover:bg-purple-700"
107:         >
108:           <Plus className="h-4 w-4 mr-2" />
109:           Add Category
110:         </Button>
111:       </div>

```



```

185:                 onChange={(e) =>
186:                     updateSubcategory(
187:                         catIndex,
188:                         subIndex,
189:                         "trait",
190:                         e.target.value
191:                     )
192:                 }
193:             />
194:         </td>
195:         <td className="border p-2">
196:             <Input
197:                 value={row.genes.join(", ")}
198:                 onChange={(e) =>
199:                     updateSubcategory(
200:                         catIndex,
201:                         subIndex,
202:                         "genes",
203:                         e.target.value.split(",").map((g) => g.trim())
204:                     )
205:                 }
206:             />
207:         </td>
208:
209:         <td className="border p-2 text-center">
210:             <Button
211:                 size="sm"
212:                 variant="destructive"
213:                 onClick={() => removeSubcategory(catIndex, subIndex)}
214:             >
215:                 <Trash2 className="h-4 w-4" />
216:             </Button>
217:         </td>
218:     </tr>
219: ))}
220: {cat.subcategories.length === 0 && (
221:     <tr>
222:         <td colspan={5} className="text-center p-4 text-gray-500">
223:             No rows added yet
224:         </td>
225:     </tr>
226: )}
227: </tbody>
228: </table>
229: </div>
230: </div>
231: )}
232: </CardContent>
233: </Card>
234: );
235: }

```

## ■ File: components\admin\ReportPreview.tsx

```

1: "use client";
2:
3: import { useRef } from "react";
4: import { useReactToPrint } from "react-to-print";
5: import { Button } from "@/components/ui/button";
6: import ComprehensiveReportViewer from "@/components/comprehensive-report-viewer";
7: import type { Report, Patient } from "@/types/report-types";
8:
9: interface ReportPreviewProps {
10:     selectedReport: Report;
11:     selectedPatient: Patient;
12: }
13:
14: export default function ReportPreview({
15:     selectedReport,
16:     selectedPatient,
17: }: ReportPreviewProps) {

```

```

18: // ? Ref to the DOM node
19: const reportRef = useRef<HTMLDivElement>(null);
20:
21: // ? Use contentRef instead of content for v3+
22: const handlePrint = useReactToPrint({
23:   contentRef: reportRef,
24:   documentTitle: `Report_${selectedPatient.info?.sampleCode || "Patient"}`,
25: });
26:
27: return (
28:   <div className="space-y-4">
29:     <div className="flex justify-end gap-2 no-print">
30:       <Button onClick={handlePrint}>Print / Save as PDF</Button>
31:     </div>
32:
33:     { /* Ref should be on a DOM node */ }
34:     <div ref={reportRef}>
35:       <ComprehensiveReportViewer
36:         reportData={{
37:           ...selectedReport,
38:           patientInfo: selectedPatient.info,
39:         }}
40:       />
41:     </div>
42:   </div>
43: );
44: }

```

## ■ File: components\admin\allergies-sensitivity-admin.tsx

```

1: "use client";
2:
3: import { useEffect, useState } from "react";
4: import { Card, CardContent, CardHeader, CardTitle } from "@/components/ui/card";
5: import { Input } from "@/components/ui/input";
6: import { Textarea } from "@/components/ui/textarea";
7: import { Label } from "@/components/ui/label";
8: import { Button } from "@/components/ui/button";
9: import {
10:   Plus,
11:   Save,
12:   Trash2,
13:   Upload,
14:   X,
15:   Check,
16:   AlertCircle,
17: } from "lucide-react";
18: import Image from "next/image";
19: import { AllergiesAndSensitivity, AllergyEntry } from "@/types/report-types";
20: import {
21:   Select,
22:   SelectTrigger,
23:   SelectValue,
24:   SelectContent,
25:   SelectItem,
26: } from "@/components/ui/select";
27: import {
28:   AlertDialog,
29:   AlertDialogAction,
30:   AlertDialogCancel,
31:   AlertDialogContent,
32:   AlertDialogDescription,
33:   AlertDialogFooter,
34:   AlertDialogHeader,
35:   AlertDialogTitle,
36: } from "@/components/ui/alert-dialog";
37:
38: interface AllergiesSensitivityAdminProps {
39:   allergiesAndSensitivity: AllergiesAndSensitivity;
40:   updateAllergiesAndSensitivity: (
41:     data: Partial<AllergiesAndSensitivity>

```

```

42:   ) => void;
43:   onSave: () => void;
44:   onReset: () => void;
45: }
46:
47: interface Notification {
48:   type: "success" | "error";
49:   message: string;
50:   show: boolean;
51: }
52:
53: export default function AllergiesSensitivityAdmin({
54:   allergiesAndSensitivity,
55:   updateAllergiesAndSensitivity,
56:   onSave,
57:   onReset,
58: }: AllergiesSensitivityAdminProps) {
59:   const [quote, setQuote] = useState(allergiesAndSensitivity.quote || "");
60:   const [description, setDescription] = useState(
61:     allergiesAndSensitivity.description || ""
62:   );
63:   const [generalAdvice, setGeneralAdvice] = useState(
64:     allergiesAndSensitivity.generalAdvice || ""
65:   );
66:   const [entries, setEntries] = useState<Record<string, AllergyEntry>>({
67:     allergiesAndSensitivity.data || {}
68:   });
69:   const [newTitle, setNewTitle] = useState("");
70:   const [images, setImages] = useState<Record<string, string>>({});
71:   const [imageLabel, setImageLabel] = useState("");
72:   const [imageFile, setImageFile] = useState<File | null>(null);
73:   const [showImageModal, setShowImageModal] = useState(false);
74:   const [showModal, setShowModal] = useState(false);
75:   const [selectedImageField, setSelectedImageField] = useState<string | null>(
76:     null
77:   );
78:
79:   // New state for confirmations and notifications
80:   const [deleteDialog, setDeleteDialog] = useState({
81:     show: false,
82:     key: "",
83:     title: "",
84:   });
85:
86:   const [imageDeleteDialog, setImageDeleteDialog] = useState({
87:     show: boolean;
88:     label: string;
89:     filename: string;
90:   })>({ show: false, label: "", filename: "" });
91:
92:   const [notification, setNotification] = useState<Notification>({
93:     type: "success",
94:     message: "",
95:     show: false,
96:   });
97:   const [isUploading, setIsUploading] = useState(false);
98:
99:   useEffect(() => {
100:     updateAllergiesAndSensitivity({
101:       quote,
102:       description,
103:       generalAdvice,
104:       data: entries,
105:     });
106:   }, [quote, description, generalAdvice, entries]);
107:
108:   useEffect(() => {
109:     const fetchImages = async () => {
110:       const res = await fetch("/api/allergies-image");
111:       const json = await res.json();
112:       if (json.success) {
113:         const map: Record<string, string> = {};
114:         for (const img of json.images) {

```

```

115:         map[img.label] = img.url;
116:     }
117:     setImages(map);
118: }
119: };
120: fetchImages();
121: }, []);
122:
123: // Show notification function
124: const showNotification = (type: "success" | "error", message: string) => {
125:     setNotification({ type, message, show: true });
126:     setTimeout(() => {
127:         setNotification((prev) => ({ ...prev, show: false }));
128:     }, 4000);
129: };
130:
131: const handleUploadImage = async () => {
132:     if (!imageLabel.trim() || !imageFile) return;
133:
134:     setIsUploading(true);
135:     try {
136:         const formData = new FormData();
137:         formData.append("file", imageFile);
138:         formData.append(
139:             "label",
140:             imageLabel.trim().toLowerCase().replace(/\s+/g, "_")
141:         );
142:         formData.append("folder", "allergies");
143:
144:         const res = await fetch("/api/allergies-image", {
145:             method: "POST",
146:             body: formData,
147:         });
148:
149:         const json = await res.json();
150:         if (json.success) {
151:             setImages((prev) => ({ ...prev, [json.label]: json.url }));
152:             setImageLabel("");
153:             setImageFile(null);
154:             showNotification("success", "Image uploaded successfully!");
155:         } else {
156:             showNotification("error", json.message || "Failed to upload image");
157:         }
158:     } catch (error) {
159:         showNotification("error", "An error occurred while uploading the image");
160:     } finally {
161:         setIsUploading(false);
162:     }
163: };
164:
165: const handleFieldChange = (
166:     key: string,
167:     field: keyof AllergyEntry,
168:     value: string
169: ) => {
170:     const updated = {
171:         ...entries,
172:         [key]: {
173:             ...entries[key],
174:             [field]: value,
175:         },
176:     };
177:     setEntries(updated);
178: };
179:
180: const handleAddEntry = () => {
181:     if (!newTitle.trim()) return;
182:     const newKey = `entry_${Date.now()}`;
183:     setEntries({
184:         ...entries,
185:         [newKey]: { title: newTitle.trim(), image: "" },
186:     });
187:     setNewTitle("");

```



```

188:     showNotification("success", "New allergy entry added successfully!");
189: };
190:
191: const handleDeleteClick = (key: string, title: string) => {
192:     setDeleteDialog({ show: true, key, title });
193: };
194:
195: const handleDeleteConfirm = () => {
196:     const updated = { ...entries };
197:     delete updated[deleteDialog.key];
198:     setEntries(updated);
199:     updateAllergiesAndSensitivity({
200:         data: updated,
201:         quote,
202:         description,
203:         generalAdvice,
204:     });
205:     setDeleteDialog({ show: false, key: "", title: "" });
206:     showNotification("success", "Allergy entry deleted successfully!");
207: };
208:
209: const handleSave = () => {
210:     onSave();
211: };
212:
213: return (
214:     <div className="min-h-screen bg-gradient-to-br from-gray-50 via-gray-100 to-gray-200 ">
215:         <div className="max-w-8xl mx-auto space-y-8">
216:             {/* Notification Toast */}
217:             {notification.show && (
218:                 <div
219:                     className={`fixed top-4 right-4 z-50 flex items-center gap-3 px-6 py-4 rounded-lg shadow-lg transition-all
220:                         ${notification.type === "success"
221:                             ? "bg-gray-800 text-white border border-gray-600"
222:                             : "bg-gray-900 text-white border border-gray-700"}
223:                 >
224:                     {notification.type === "success" ? (
225:                         <Check className="w-5 h-5" />
226:                     ) : (
227:                         <AlertCircle className="w-5 h-5" />
228:                     )}
229:                     <span className="font-medium">{notification.message}</span>
230:                     <button
231:                         onClick={() =>
232:                             setNotification((prev) => ({ ...prev, show: false }))
233:                         }
234:                         className="ml-2 hover:bg-white/20 rounded-full p-1"
235:                     >
236:                         <X className="w-4 h-4" />
237:                     </button>
238:                 </div>
239:             )}
240:
241:             {/* Header */}
242:             <Card className="shadow-xl border border-gray-300">
243:                 <CardHeader className="bg-zinc-600 text-white rounded-lg p-4 sm:p-5">
244:                     <CardTitle className="text-lg sm:text-xl md:text-2xl font-bold uppercase flex items-center justify-between">
245:                         Allergies & Sensitivity Management
246:                     </CardTitle>
247:                 </CardHeader>
248:             </Card>
249:
250:             <div className="text-end my-4">
251:                 <Button onClick={handleSave}>
252:                     <Save className="w-5 h-5 mr-2" />
253:                     Save Changes
254:                 </Button>
255:             </div>
256:
257:             {/* Quote & Description */}
258:             <div className="grid grid-cols-1 lg:grid-cols-2 gap-6 p-4">
259:                 <Card className="shadow-lg border border-gray-300 hover:shadow-xl transition-shadow">

```

```

261:         <CardHeader className="bg-gradient-to-r from-gray-300 to-gray-400 text-black rounded-t-lg">
262:             <CardTitle className="text-xl flex items-center gap-2">
263:                 Content Settings
264:             </CardTitle>
265:         </CardHeader>
266:         <Label className="text-sm font-medium px-6">Quote</Label>
267:         <CardContent className="p-6 bg-white">
268:             <Input
269:                 value={quote}
270:                 onChange={(e) => setQuote(e.target.value)}
271:                 placeholder="e.g. 'Allergies are not your destiny, they're just a part of your journey...'"
272:                 className="resize-none border-2 border-gray-300 focus:border-gray-500 bg-white text-gray-800"
273:             />
274:         </CardContent>
275:         <Label className="text-sm font-medium px-6">Description</Label>
276:         <CardContent className="p-6 bg-white rounded-lg">
277:             <Input
278:                 value={description}
279:                 onChange={(e) => setDescription(e.target.value)}
280:                 placeholder="Provide a comprehensive description for the allergies section..."
281:                 className="resize-none border-2 border-gray-300 focus:border-gray-500 bg-white text-gray-800"
282:             />
283:         </CardContent>
284:     </Card>
285:
286:     <Card className="shadow-lg border border-gray-300 hover:shadow-xl transition-shadow">
287:         <CardHeader className="bg-gradient-to-r from-gray-300 to-pink-200 text-white rounded-t-lg">
288:             <CardTitle className="text-xl flex items-center gap-2">
289:                 Upload New Image
290:             </CardTitle>
291:         </CardHeader>
292:         <CardContent className="p-6 bg-white rounded-lg">
293:             <div className="grid grid-cols-1 md:grid-cols-3 gap-4 items-end">
294:                 <div>
295:                     <Label className="text-sm font-semibold text-gray-700 mb-2 block">
296:                         Image Label
297:                     </Label>
298:                     <Input
299:                         placeholder="e.g. peanut_allergy"
300:                         value={imageLabel}
301:                         onChange={(e) => setImageLabel(e.target.value)}
302:                         className="border-2 border-gray-300 focus:border-gray-500 bg-white text-gray-800"
303:                     />
304:                 </div>
305:                 <div>
306:                     <Label className="text-sm font-semibold text-gray-700 mb-2 block">
307:                         Select File
308:                     </Label>
309:                     <Input
310:                         type="file"
311:                         accept="image/*"
312:                         onChange={(e) => setImageFile(e.target.files?.[0] || null)}
313:                         className="border-2 border-gray-300 focus:border-gray-500 bg-white text-gray-800"
314:                     />
315:                 </div>
316:                 <Button
317:                     onClick={handleUploadImage}
318:                     disabled={!imageLabel || !imageFile || isUploading}
319:                     className="bg-gray-800 hover:bg-black text-white font-semibold py-3"
320:                 >
321:                     {isUploading ? (
322:                         <>
323:                             <div className="w-4 h-4 mr-2 border-2 border-white border-t-transparent rounded-full" style={{border: 2px solid transparent, border-top: 2px solid white; border-radius: 50%;}}>
324:                                 Uploading...
325:                             </div>
326:                         </>
327:                     ) : (
328:                         <>
329:                             <Upload className="w-4 h-4 mr-2" />
330:                             Upload Image
331:                         </>
332:                     )}
333:                 </Button>

```

```

334:         <Button
335:           variant="outline"
336:           onClick={() => setShowImageModal(true)}
337:           className="text-gray-700 border-gray-400 hover:bg-gray-100"
338:         >
339:           View Uploaded Images
340:         </Button>
341:       </div>
342:     </CardContent>
343:   </Card>
344:</div>
345:
346: { /* Add Allergy Entry */ }
347:<Card className="shadow-lg border border-gray-300 hover:shadow-xl transition-shadow m-4">
348:  <CardHeader className="bg-gradient-to-r from-pink-300 to-pink-400 text-white rounded-t-lg">
349:    <CardTitle className="text-lg sm:text-xl flex items-center gap-2">
350:      Add New Allergy Entry
351:    </CardTitle>
352:  </CardHeader>
353:
354:  <CardContent className="p-4 sm:p-6 rounded-lg bg-white">
355:    <div className="flex flex-col sm:flex-row gap-4 items-stretch sm:items-end rounded-lg">
356:      { /* Input Field */ }
357:      <div className="flex-1">
358:        <Label className="text-sm font-semibold text-gray-700 mb-1 block">
359:          Allergy Title
360:        </Label>
361:        <Input
362:          placeholder="e.g. Peanut Allergy, Dust Mites, etc."
363:          value={newTitle}
364:          onChange={(e) => setNewTitle(e.target.value)}
365:          onKeyDown={(e) => e.key === "Enter" && handleAddEntry()}
366:          className="border-2 border-gray-300 focus:border-gray-500 bg-white text-gray-800"
367:        />
368:      </div>
369:
370:      { /* Button */ }
371:      <Button
372:        onClick={handleAddEntry}
373:        disabled={!newTitle.trim()}
374:        className="w-full sm:w-auto bg-gray-800 hover:bg-black text-white font-semibold py-3 sm:px-4"
375:      >
376:        <Plus className="w-5 h-5 mr-2" />
377:        Add Entry
378:      </Button>
379:    </div>
380:  </CardContent>
381:</Card>
382:
383: { /* Allergy Entries */ }
384:{Object.keys(entries).length > 0 && (
385:  <div className="rounded-lg shadow-lg border border-gray-300 hover:shadow-xl transition-shadow p-4">
386:    <h2 className="text-2xl font-bold text-gray-800 mb-6 flex items-center gap-2">
387:      Allergy Entries ({Object.keys(entries).length})
388:    </h2>
389:    <div className="grid grid-cols-1 md:grid-cols-2 xl:grid-cols-3 gap-6 ">
390:      {Object.entries(entries).map(([key, entry], idx) => (
391:        <Card
392:          key={key}
393:          className="shadow-lg border border-gray-300 hover:shadow-xl "
394:        >
395:          <CardHeader className="bg-gradient-to-r from-gray-700 to-gray-900 text-white rounded-lg">
396:            <div className="flex justify-between items-center">
397:              <CardTitle className="text-lg font-semibold">
398:                {entry.title || `Allergy ${idx + 1}`}
399:              </CardTitle>
400:              <Button
401:                size="icon"
402:                variant="ghost"
403:                className="text-white hover:bg-white/20"
404:                onClick={() => handleDeleteClick(key, entry.title)}
405:              >
406:                <Trash2 className="w-4 h-4" />

```

```

407:         </Button>
408:     </div>
409: </CardHeader>
410: <CardContent className="space-y-4 p-6 bg-white rounded-lg">
411:     <div>
412:         <Label className="text-sm font-semibold text-gray-700 mb-2 block">
413:             Title
414:         </Label>
415:         <Input
416:             value={entry.title}
417:             onChange={(e) =>
418:                 handleFieldChange(key, "title", e.target.value)
419:             }
420:             className="border-2 border-gray-300 focus:border-gray-500 bg-white text-gray-800"
421:         />
422:     </div>
423:     <div>
424:         <Label className="text-sm font-semibold text-gray-700 mb-2 block">
425:             Associated Image
426:         </Label>
427:         <Select
428:             value={entry.image}
429:             onValueChange={(val) =>
430:                 handleFieldChange(key, "image", val)
431:             }
432:         >
433:             <SelectTrigger className="border-2 border-gray-300 focus:border-gray-500 bg-white text-gray-800">
434:                 <SelectValue placeholder="Choose an image" />
435:             </SelectTrigger>
436:             <SelectContent className="bg-white border border-gray-300">
437:                 {Object.entries(images).map(([label, url]) => (
438:                     <SelectItem
439:                         key={label}
440:                         value={label}
441:                         className="hover:bg-gray-100"
442:                     >
443:                         <div className="flex items-center gap-3">
444:                             <Image
445:                                 src={url}
446:                                 alt={label}
447:                                 width={24}
448:                                 height={24}
449:                                 className="rounded"
450:                             />
451:                             <span className="capitalize text-gray-800">
452:                                 {label.replace(/_/g, " ")}
453:                             </span>
454:                         </div>
455:                     </SelectItem>
456:                 )))
457:             </SelectContent>
458:         </Select>
459:     </div>
460: </CardContent>
461: </Card>
462: ))}
463: </div>
464: </div>
465: )}
466:
467: { /* General Advice */}
468: <Card className="shadow-lg border border-gray-300 hover:shadow-xl transition-shadow m-4">
469:     <CardHeader className="bg-gradient-to-r from-gray-800 to-black text-white rounded-lg">
470:         <CardTitle className="text-2xl flex items-center gap-2">
471:             General Advice & Guidelines
472:         </CardTitle>
473:     </CardHeader>
474:     <CardContent className="p-6 bg-white">
475:         <Textarea
476:             rows={6}
477:             value={generalAdvice}
478:             onChange={(e) => setGeneralAdvice(e.target.value)}
479:             placeholder="Enter comprehensive guidance for allergy management, prevention tips, emergency p

```

```

480:         className="min-h-[160px] resize-none border-2 border-gray-300 focus:border-gray-500 bg-white
481:     />
482: </CardContent>
483: </Card>
484:
485: { /* Delete Confirmation Dialog */}
486: <AlertDialog
487:   open={deleteDialog.show}
488:   onOpenChange={({open}) =>
489:     !open && setDeleteDialog({ show: false, key: "", title: "" })
490:   }
491: >
492:   <AlertDialogContent className="max-w-md bg-white border border-gray-300">
493:     <AlertDialogHeader>
494:       <AlertDialogTitle className="flex items-center gap-2 text-gray-800">
495:         <Trash2 className="w-5 h-5" />
496:         Confirm Deletion
497:       </AlertDialogTitle>
498:       <AlertDialogDescription className="text-gray-600">
499:         Are you sure you want to delete the allergy entry "
500:         {deleteDialog.title}"? This action cannot be undone.
501:       </AlertDialogDescription>
502:     </AlertDialogHeader>
503:     <AlertDialogFooter>
504:       <AlertDialogCancel className="font-semibold border-gray-300 text-gray-700 hover:bg-gray-100">
505:         Cancel
506:       </AlertDialogCancel>
507:       <AlertDialogAction
508:         onClick={handleDeleteConfirm}
509:         className="bg-gray-800 hover:bg-black text-white font-semibold"
510:       >
511:         Delete Entry
512:       </AlertDialogAction>
513:     </AlertDialogFooter>
514:   </AlertDialogContent>
515: </AlertDialog>
516:
517: <AlertDialog
518:   open={imageDeleteDialog.show}
519:   onOpenChange={({open}) =>
520:     !open &&
521:     setImageDeleteDialog({ show: false, label: "", filename: "" })
522:   }
523: >
524:   <AlertDialogContent className="max-w-md bg-white border border-gray-300">
525:     <AlertDialogHeader>
526:       <AlertDialogTitle className="flex items-center gap-2 text-gray-800">
527:         <Trash2 className="w-5 h-5" />
528:         Delete Image: {imageDeleteDialog.label.replace(/_/g, " ")}
529:       </AlertDialogTitle>
530:       <AlertDialogDescription className="text-gray-600">
531:         Are you sure you want to delete this image? This action cannot
532:         be undone.
533:       </AlertDialogDescription>
534:     </AlertDialogHeader>
535:     <AlertDialogFooter>
536:       <AlertDialogCancel className="font-semibold border-gray-300 text-gray-700 hover:bg-gray-100">
537:         Cancel
538:       </AlertDialogCancel>
539:       <AlertDialogAction
540:         onClick={async () => {
541:           try {
542:             const res = await fetch("/api/allergies-image", {
543:               method: "DELETE",
544:               body: JSON.stringify({
545:                 filename: imageDeleteDialog.filename,
546:               }),
547:               headers: {
548:                 "Content-Type": "application/json",
549:               },
550:             });
551:             const json = await res.json();
552:             if (json.success) {

```

```

553:         const updated = { ...images };
554:         delete updated[imageDeleteDialog.label];
555:         setImages(updated);
556:         showNotification(
557:             "success",
558:             "Image deleted successfully!"
559:         );
560:     } else {
561:         showNotification(
562:             "error",
563:             json.error || "Failed to delete image"
564:         );
565:     }
566: } catch (err) {
567:     console.error("Error deleting image:", err);
568:     showNotification("error", "Error deleting image");
569: } finally {
570:     setImageDeleteDialog({
571:         show: false,
572:         label: "",
573:         filename: "",
574:     });
575: }
576: }}
577: className="bg-red-600 hover:bg-red-700 text-white"
578: >
579:     Confirm Delete
580: </AlertDialogAction>
581: </AlertDialogFooter>
582: </AlertDialogContent>
583: </AlertDialog>
584:
585: { /* Image List Modal */}
586: {showImageModal && (
587:     <div className="fixed inset-0 z-50 flex items-center justify-center p-2 sm:p-4 ">
588:         <div className="bg-white rounded-xl shadow-2xl w-full max-w-5xl max-h-[90vh] overflow-hidden flex flex-col">
589:             { /* Header */}
590:             <div className="flex justify-between items-center px-4 sm:px-6 py-4 border-b bg-gradient-to-r from-red-600 to-red-700">
591:                 <h3 className="text-lg sm:text-xl font-semibold text-gray-800">
592:                     Uploaded Allergy Images
593:                 </h3>
594:                 <button
595:                     onClick={() => setShowImageModal(false)}
596:                     className="text-gray-500 hover:text-gray-700 hover:bg-gray-100 p-1 rounded-full transition-colors">
597:                     <X className="w-5 h-5" />
598:                 </button>
599:             </div>
600:             </div>
601:
602:             { /* Content */}
603:             <div className="px-4 sm:px-6 py-4 overflow-y-auto max-h-[calc(90vh-120px)]">
604:                 <div className="space-y-3">
605:                     {Object.entries(images).map(([label, url]) => {
606:                         const filename = url.split("/").pop()!;
607:                         return (
608:                             <div
609:                                 key={label}
610:                                 className="flex flex-col sm:flex-row sm:items-center sm:justify-between gap-3 border-1 rounded-md p-3">
611:                                 <div className="flex items-center gap-4 flex-1 overflow-hidden">
612:                                     <Image
613:                                         src={url}
614:                                         alt={label}
615:                                         width={50}
616:                                         height={50}
617:                                         className="rounded-md object-cover flex-shrink-0"
618:                                     />
619:                                     <span className="capitalize text-sm text-gray-700 truncate w-full">
620:                                         {label.replace(/_/g, " ")}
621:                                     </span>
622:                                 </div>
623:                                 </div>
624:
625:                                 <div className="flex justify-end sm:justify-start">

```

```

626:                 <Button
627:                     size="icon"
628:                     variant="destructive"
629:                     onClick={() =>
630:                         setImageDeleteDialog({
631:                             show: true,
632:                             label,
633:                             filename,
634:                         })
635:                     }
636:                 >
637:                     <Trash2 className="w-4 h-4" />
638:                 </Button>
639:             </div>
640:         </div>
641:     );
642:   })}
643: </div>
644: </div>
645: </div>
646: </div>
647:   })
648: </div>
649: </div>
650: );
651: }

```

## ■ File: components\admin\content-admin.tsx

```

1: "use client";
2:
3: import {
4:   Card,
5:   CardContent,
6:   CardDescription,
7:   CardHeader,
8:   CardTitle,
9: } from "@components/ui/card";
10: import { Label } from "@components/ui/label";
11: import { Textarea } from "@components/ui/textarea";
12: import { Input } from "@components/ui/input";
13: import { Button } from "@components/ui/button";
14: import { Save, RotateCcw } from "lucide-react"; // Import icons
15: import type { ReportContent } from "@types/report-types";
16:
17: interface ContentAdminProps {
18:   content: ReportContent;
19:   updateContent: (field: keyof ReportContent, value: string) => void;
20:   onSave: () => void; // Added onSave
21:   onReset: () => void; // Added onReset
22: }
23:
24: export default function ContentAdmin({
25:   content,
26:   updateContent,
27:   onSave,
28:   onReset,
29: }: ContentAdminProps) {
30:   // Prevent crash when content or its fields are undefined
31:   const safeContent: ReportContent = {
32:     introduction: content?.introduction ?? "",
33:     genomicsExplanation: content?.genomicsExplanation ?? "",
34:     genesHealthImpact: content?.genesHealthImpact ?? "",
35:     fundamentalsPRS: content?.fundamentalsPRS ?? "",
36:     utilityDoctors: content?.utilityDoctors ?? "",
37:     microarrayExplanation: content?.microarrayExplanation ?? "",
38:     microarrayData: content?.microarrayData ?? "",
39:   };
40:
41:   return (
42:     <Card className="shadow-lg border-0 bg-white">

```

```

43:     <CardHeader className="bg-gradient-to-r from-green-500 to-teal-600 text-white rounded-lg px-3 py-4 sm:
44:         <CardTitle className="text-lg sm:text-xl md:text-2xl lg:text-3xl font-semibold uppercase text-center
45:             Report Content
46:         </CardTitle>
47:     </CardHeader>
48:     <CardContent className="space-y-8 w-full">
49:         { /* Action Buttons */ }
50:         <div className="flex justify-end gap-2 my-6">
51:             <Button onClick={onSave} size="sm">
52:                 <Save className="h-4 w-4 mr-2" />
53:                 Save Changes
54:             </Button>
55:         </div>
56:
57:         <div className="space-y-3">
58:             <Label
59:                 htmlFor="introduction"
60:                 className="text-lg font-semibold text-gray-800"
61:             >
62:                 Welcome Letter Introduction
63:             </Label>
64:             <Textarea
65:                 id="introduction"
66:                 value={safeContent.introduction}
67:                 onChange={(e) => updateContent("introduction", e.target.value)}
68:                 placeholder="Enter introduction text"
69:                 rows={9}
70:                 className="border-2 focus:border-green-500 text-sm leading-relaxed"
71:             />
72:         </div>
73:         <div className="space-y-3">
74:             <Label
75:                 htmlFor="genomicsExplanation"
76:                 className="text-lg font-semibold text-gray-800"
77:             >
78:                 What is Genomics?
79:             </Label>
80:             <Input
81:                 id="genomicsExplanation"
82:                 value={safeContent.genomicsExplanation}
83:                 onChange={(e) =>
84:                     updateContent("genomicsExplanation", e.target.value)
85:                 }
86:                 placeholder="Enter genomics explanation"
87:                 className="border-2 focus:border-green-500 text-sm leading-relaxed"
88:             />
89:         </div>
90:         <div className="space-y-3">
91:             <Label
92:                 htmlFor="genesHealthImpact"
93:                 className="text-lg font-semibold text-gray-800"
94:             >
95:                 Genes and Health Impact
96:             </Label>
97:             <Input
98:                 id="genesHealthImpact"
99:                 value={safeContent.genesHealthImpact}
100:                 onChange={(e) => updateContent("genesHealthImpact", e.target.value)}
101:                 placeholder="Enter genes and health impact content"
102:                 className="border-2 focus:border-green-500 text-sm leading-relaxed"
103:             />
104:         </div>
105:         <div className="space-y-3">
106:             <Label
107:                 htmlFor="fundamentalsPRS"
108:                 className="text-lg font-semibold text-gray-800"
109:             >
110:                 Fundamentals and PRS
111:             </Label>
112:             <Input
113:                 id="fundamentalsPRS"
114:                 value={safeContent.fundamentalsPRS}
115:                 onChange={(e) => updateContent("fundamentalsPRS", e.target.value)}

```



```

116:         placeholder="Enter fundamentals and PRS content"
117:         className="border-2 focus:border-green-500 text-sm leading-relaxed"
118:     />
119: </div>
120: <div className="space-y-3">
121:     <Label
122:         htmlFor="utilityDoctors"
123:         className="text-lg font-semibold text-gray-800"
124:     >
125:         Utility for Doctors and Dietitians
126:     </Label>
127:     <Input
128:         id="utilityDoctors"
129:         value={safeContent.utilityDoctors}
130:         onChange={(e) => updateContent("utilityDoctors", e.target.value)}
131:         placeholder="Enter utility for doctors content"
132:         className="border-2 focus:border-green-500 text-sm leading-relaxed"
133:     />
134: </div>
135: <div className="space-y-3">
136:     <Label
137:         htmlFor="microarrayExplanation"
138:         className="text-lg font-semibold text-gray-800"
139:     >
140:         Microarray Explanation
141:     </Label>
142:     <Input
143:         id="microarrayExplanation"
144:         value={safeContent.microarrayExplanation}
145:         onChange={(e) =>
146:             updateContent("microarrayExplanation", e.target.value)
147:         }
148:         placeholder="Enter microarray explanation"
149:         className="border-2 focus:border-green-500 text-sm leading-relaxed"
150:     />
151: </div>
152: <div className="space-y-3">
153:     <Label
154:         htmlFor="microarrayData"
155:         className="text-lg font-semibold text-gray-800"
156:     >
157:         Your Microarray Data
158:     </Label>
159:     <Input
160:         id="microarrayData"
161:         value={safeContent.microarrayData}
162:         onChange={(e) => updateContent("microarrayData", e.target.value)}
163:         placeholder="Enter microarray data content"
164:         className="border-2 focus:border-green-500 text-sm leading-relaxed"
165:     />
166: </div>
167: </CardContent>
168: </Card>
169: );
170: }

```

## **File: components\admin\diet-analysis-admin.tsx**

```

=====
1: "use client";
2:
3: import {
4:   Card,
5:   CardContent,
6:   CardDescription,
7:   CardHeader,
8:   CardTitle,
9: } from "@components/ui/card";
10: import { Input } from "@components/ui/input";
11: import { Label } from "@components/ui/label";
12: import { Textarea } from "@components/ui/textarea";
13: import {

```

```

14:   Select,
15:   SelectContent,
16:   SelectItem,
17:   SelectTrigger,
18:   SelectValue,
19: } from "@components/ui/select";
20: import { Button } from "@components/ui/button"; // Import Button
21: import { Save, RotateCcw } from "lucide-react"; // Import icons
22: import type { DietAnalysis, MacronutrientData } from "@types/report-types";
23:
24: interface DietAnalysisAdminProps {
25:   dietAnalysis: DietAnalysis;
26:   updateDietAnalysis: (
27:     section: keyof DietAnalysis,
28:     field: string,
29:     data: Partial<MacronutrientData>
30:   ) => void;
31:   onSave: () => void; // Added onSave
32:   onReset: () => void; // Added onReset
33: }
34:
35: export default function DietAnalysisAdmin({
36:   dietAnalysis,
37:   updateDietAnalysis,
38:   onSave,
39:   onReset,
40: }: DietAnalysisAdminProps) {
41:   const renderMacronutrientField = (
42:     section: keyof DietAnalysis,
43:     field: string,
44:     data: MacronutrientData,
45:     title: string
46:   ) => (
47:     <Card key={field} className="border-2 border-blue-100 bg-blue-50/30">
48:       <CardHeader className="pb-3">
49:         <CardTitle className="text-lg text-blue-800">{title}</CardTitle>
50:       </CardHeader>
51:       <CardContent className="space-y-4">
52:         <div className="grid grid-cols-2 gap-4">
53:           <div className="space-y-2">
54:             <Label className="text-sm font-medium">Score (1-10)</Label>
55:             <Input
56:               type="number"
57:               min="1"
58:               max="10"
59:               value={data.score}
60:               onChange={(e) =>
61:                 updateDietAnalysis(section, field, {
62:                   score: Number.parseInt(e.target.value) || 1,
63:                 })
64:               }
65:             className="border-2 focus:border-blue-500"
66:           />
67:         </div>
68:         <div className="space-y-2">
69:           <Label className="text-sm font-medium">Level</Label>
70:           <Select
71:             value={data.level}
72:             onValueChange={(value) =>
73:               updateDietAnalysis(section, field, { level: value })
74:             }
75:           >
76:             <SelectTrigger className="border-2 focus:border-blue-500">
77:               <SelectValue />
78:             </SelectTrigger>
79:             <SelectContent>
80:               <SelectItem value="HIGH">HIGH</SelectItem>
81:               <SelectItem value="NORMAL">NORMAL</SelectItem>
82:               <SelectItem value="LOW">LOW</SelectItem>
83:             </SelectContent>
84:           </Select>
85:         </div>
86:       </div>

```

```

87:         <div className="space-y-2">
88:             <Label className="text-sm font-medium">Recommendation</Label>
89:             <Textarea
90:                 value={data.recommendation}
91:                 onChange={(e) =>
92:                     updateDietAnalysis(section, field, {
93:                         recommendation: e.target.value,
94:                     })
95:             }
96:             rows={3}
97:             className="border-2 focus:border-blue-500 text-sm"
98:         />
99:     </div>
100: </CardContent>
101: </Card>
102: );
103:
104: return (
105:     <Card className="shadow-lg border-0 bg-white">
106:         <CardHeader className="bg-gradient-to-r from-amber-500 to-orange-600 text-white rounded-t-lg">
107:             <CardTitle className="text-2xl">? Diet Analysis</CardTitle>
108:             <CardDescription className="text-amber-100">
109:                 Configure macronutrients, meal patterns, and food sensitivities
110:             </CardDescription>
111:         </CardHeader>
112:         <CardContent className="space-y-8 p-8">
113:             { /* Action Buttons */ }
114:             <div className="flex justify-end gap-2 mb-6">
115:                 <Button onClick={onReset} variant="outline" size="sm">
116:                     <RotateCcw className="h-4 w-4 mr-2" />
117:                     Reset
118:                 </Button>
119:                 <Button onClick={onSave} size="sm">
120:                     <Save className="h-4 w-4 mr-2" />
121:                     Save Changes
122:                 </Button>
123:             </div>
124:
125:             { /* Macronutrients Section */ }
126:             <div className="space-y-4">
127:                 <h3 className="text-xl font-semibold text-gray-800 border-b-2 border-amber-200 pb-2">
128:                     ? Macronutrients
129:                 </h3>
130:                 <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-6">
131:                     {renderMacronutrientField(
132:                         "macronutrients",
133:                         "carbohydrateSensitivity",
134:                         dietAnalysis.macronutrients.carbohydrateSensitivity,
135:                         "Carbohydrate Sensitivity"
136:                     )}
137:                     {renderMacronutrientField(
138:                         "macronutrients",
139:                         "fatSensitivity",
140:                         dietAnalysis.macronutrients.fatSensitivity,
141:                         "Fat Sensitivity"
142:                     )}
143:                     {renderMacronutrientField(
144:                         "macronutrients",
145:                         "proteinRequirement",
146:                         dietAnalysis.macronutrients.proteinRequirement,
147:                         "Protein Requirement"
148:                     )}
149:                 </div>
150:             </div>
151:
152:             { /* Meal Pattern Section */ }
153:             <div className="space-y-4">
154:                 <h3 className="text-xl font-semibold text-gray-800 border-b-2 border-amber-200 pb-2">
155:                     ?? Meal Pattern
156:                 </h3>
157:                 <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-6">
158:                     {renderMacronutrientField(
159:                         "mealPattern",

```

```

160:         "mealFrequency",
161:         dietAnalysis.mealPattern.mealFrequency,
162:         "Meal Frequency"
163:     )}
164:     {renderMacronutrientField(
165:         "mealPattern",
166:         "mealReplacement",
167:         dietAnalysis.mealPattern.mealReplacement,
168:         "Meal Replacement"
169:     )}
170:     {renderMacronutrientField(
171:         "mealPattern",
172:         "weightMaintenance",
173:         dietAnalysis.mealPattern.weightMaintenance,
174:         "Weight Maintenance"
175:     )}
176: </div>
177: </div>
178:
179: { /* Food Sensitivities Section */}
180: <div className="space-y-4">
181:     <h3 className="text-xl font-semibold text-gray-800 border-b-2 border-amber-200 pb-2">
182:         Food Sensitivities
183:     </h3>
184:     <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-6">
185:         {renderMacronutrientField(
186:             "foodSensitivities",
187:             "alcoholSensitivity",
188:             dietAnalysis.foodSensitivities.alcoholSensitivity,
189:             "Alcohol Sensitivity"
190:         )}
191:         {renderMacronutrientField(
192:             "foodSensitivities",
193:             "caffeineSensitivity",
194:             dietAnalysis.foodSensitivities.caffeineSensitivity,
195:             "Caffeine Sensitivity"
196:         )}
197:         {renderMacronutrientField(
198:             "foodSensitivities",
199:             "glutenSensitivity",
200:             dietAnalysis.foodSensitivities.glutenSensitivity,
201:             "Gluten Sensitivity"
202:         )}
203:         {renderMacronutrientField(
204:             "foodSensitivities",
205:             "lactoseSensitivity",
206:             dietAnalysis.foodSensitivities.lactoseSensitivity,
207:             "Lactose Sensitivity"
208:         )}
209:         {renderMacronutrientField(
210:             "foodSensitivities",
211:             "saltSensitivity",
212:             dietAnalysis.foodSensitivities.saltSensitivity,
213:             "Salt Sensitivity"
214:         )}
215:     </div>
216: </div>
217:
218: { /* Taste Sensitivities Section */}
219: <div className="space-y-4">
220:     <h3 className="text-xl font-semibold text-gray-800 border-b-2 border-amber-200 pb-2">
221:         ? Taste Sensitivities
222:     </h3>
223:     <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-6">
224:         {renderMacronutrientField(
225:             "tasteSensitivities",
226:             "spiceTolerance",
227:             dietAnalysis.tasteSensitivities.spiceTolerance,
228:             "Spice Tolerance"
229:         )}
230:         {renderMacronutrientField(
231:             "tasteSensitivities",
232:             "sweetTaste",

```

```

233:         dietAnalysis.tasteSensitivities.sweetTaste,
234:         "Sweet Taste"
235:     )}
236:     {renderMacronutrientField(
237:         "tasteSensitivities",
238:         "tasteSensitivity",
239:         dietAnalysis.tasteSensitivities.tasteSensitivity,
240:         "Taste Sensitivity"
241:     )}
242: </div>
243: </div>
244: </CardContent>
245: </Card>
246: );
247: }

```

## ■ File: components\admin\digestive-health-admin.tsx

```

1: "use client";
2:
3: import { Card, CardContent, CardHeader, CardTitle } from "@/components/ui/card";
4: import { Input } from "@/components/ui/input";
5: import { Label } from "@/components/ui/label";
6: import { Textarea } from "@/components/ui/textarea";
7: import { Button } from "@/components/ui/button";
8: import { Plus, Trash2, Save, RotateCcw, Eye, Upload, X } from "lucide-react";
9: import Image from "next/image";
10: import { useEffect, useState } from "react";
11: import {
12:   Dialog,
13:   DialogTrigger,
14:   DialogContent,
15:   DialogHeader,
16:   DialogTitle,
17:   DialogDescription,
18:   DialogClose,
19: } from "@/components/ui/dialog";
20:
21: import {
22:   Select,
23:   SelectTrigger,
24:   SelectValue,
25:   SelectContent,
26:   SelectItem,
27: } from "@/components/ui/select";
28:
29: import type {
30:   DigestiveHealth,
31:   DigestiveHealthEntry,
32: } from "@/types/report-types";
33:
34: interface DigestiveHealthAdminProps {
35:   digestiveHealth: DigestiveHealth;
36:   updateField: (key: string, data: Partial<DigestiveHealthEntry>) => void;
37:   addField: (key: string) => void;
38:   deleteField: (key: string) => void;
39:   updateQuoteAndDescription: (quote: string, description: string) => void;
40:   onSave: () => void;
41:   onReset: () => void;
42: }
43:
44: export default function DigestiveHealthAdmin({
45:   digestiveHealth,
46:   updateField,
47:   addField,
48:   deleteField,
49:   updateQuoteAndDescription,
50:   onSave,
51:   onReset,
52: }: DigestiveHealthAdminProps) {
53:   const [newFieldName, setNewFieldName] = useState("");

```

```

54: const [showDeleteModal, setShowDeleteModal] = useState(false);
55: const [deleteKey, setDeleteKey] = useState<string | null>(null);
56: const [customImages, setCustomImages] = useState<Record<string, string>>({});
57: const [showImageForm, setShowImageForm] = useState(false);
58: const [imageLabel, setImageLabel] = useState("");
59: const [imagePreview, setImagePreview] = useState<string>("");
60: const [selectedFile, setSelectedFile] = useState<File | null>(null);
61: const [showImageList, setShowImageList] = useState(false);
62:
63: // Image preview for file input
64: const handleFileChange = (e: React.ChangeEvent<HTMLInputElement>) => {
65:   const file = e.target.files?.[0];
66:   if (file) {
67:     setSelectedFile(file);
68:     const reader = new FileReader();
69:     reader.onloadend = () => {
70:       setImagePreview(reader.result as string);
71:     };
72:     reader.readAsDataURL(file);
73:   } else {
74:     setImagePreview("");
75:     setSelectedFile(null);
76:   }
77: };
78:
79: const clearImagePreview = () => {
80:   setImagePreview("");
81:   setSelectedFile(null);
82:   setImageLabel("");
83: };
84:
85: // Fetch image list from /api/digestive-images
86: useEffect(() => {
87:   const fetchImages = async () => {
88:     try {
89:       const res = await fetch("/api/digestive-images");
90:       const data = await res.json();
91:       if (data.success) {
92:         const imageMap: Record<string, string> = {};
93:         data.images.forEach((img: { label: string; url: string }) => {
94:           imageMap[img.label.toLowerCase()] = img.url;
95:         });
96:         setCustomImages(imageMap);
97:       }
98:     } catch (err) {
99:       console.error("Failed to load digestive images", err);
100:     }
101:   };
102:   fetchImages();
103: }, []);
104:
105: const handleQuoteChange = (quote: string) =>
106:   updateQuoteAndDescription(quote, digestiveHealth.description);
107:
108: const handleDescriptionChange = (desc: string) =>
109:   updateQuoteAndDescription(digestiveHealth.quote, desc);
110:
111: const handleAddNewField = () => {
112:   const key = newFieldName.trim();
113:   if (!key || digestiveHealth.data[key]) return;
114:   addField(key);
115:   setNewFieldName("");
116: };
117:
118: const handleDeleteFieldConfirm = () => {
119:   if (deleteKey) deleteField(deleteKey);
120:   setShowDeleteModal(false);
121:   setDeleteKey(null);
122: };
123:
124: const handleImageUpload = async () => {
125:   const label = imageLabel.trim().toLowerCase();
126:   const file = selectedFile;

```

```

127:
128:     if (!file || !label) {
129:         alert("Both label and file required");
130:         return;
131:     }
132:
133:     const uploadData = new FormData();
134:     uploadData.append("file", file);
135:     uploadData.append("folder", "digestive");
136:     uploadData.append("label", label);
137:
138:     try {
139:         const response = await fetch("/api/digestive-images", {
140:             method: "POST",
141:             body: uploadData,
142:         });
143:         const result = await response.json();
144:
145:         if (result.success) {
146:             setCustomImages((prev) => ({
147:                 ...prev,
148:                 [label]: result.url,
149:             }));
150:             clearImagePreview();
151:         } else {
152:             alert("Upload failed");
153:         }
154:     } catch (err) {
155:         console.error("Upload error", err);
156:         alert("Upload error");
157:     }
158: };
159:
160: return (
161:     <div className="w-full max-w-8xl mx-auto space-y-6">
162:         {/* Header Card */}
163:         <Card className="border-0 shadow-lg">
164:             <CardHeader className="bg-gradient-to-r from-red-500 to-orange-500 text-white rounded-lg p-4 sm:p-5">
165:                 <CardTitle className="text-lg sm:text-xl md:text-2xl font-bold uppercase flex items-center justify-center">
166:                     Digestive Health
167:                 </CardTitle>
168:             </CardHeader>
169:         </Card>
170:
171:         <div className="text-end mt-4">
172:             <Button onClick={onSave}>
173:                 <Save className="h-4 w-4 mr-2" />
174:                 Save Changes
175:             </Button>
176:         </div>
177:
178:         {/* Configuration Cards */}
179:         <div className="grid grid-cols-1 lg:grid-cols-2 gap-6">
180:             {/* Quote & Description Card */}
181:             <Card className="shadow-md">
182:                 <CardContent className="space-y-4 mt-4">
183:                     <div className="space-y-2">
184:                         <Label className="text-sm font-medium">Quote</Label>
185:                         <Input
186:                             value={digestiveHealth.quote}
187:                             onChange={(e) => handleQuoteChange(e.target.value)}
188:                             placeholder="Enter inspirational quote..."
189:                         />
190:                     </div>
191:                     <div className="space-y-2">
192:                         <Label className="text-sm font-medium">Description</Label>
193:                         <Input
194:                             value={digestiveHealth.description}
195:                             onChange={(e) => handleDescriptionChange(e.target.value)}
196:                             placeholder="Enter description..."
197:                         />
198:                     </div>
199:                 </CardContent>

```

```

200:     </Card>
201:
202:     { /* Image Upload Card */ }
203:     <Card className="shadow-md">
204:         <CardHeader className="pb-3">
205:             <CardTitle className="text-lg text-gray-800">
206:                 Image Management
207:             </CardTitle>
208:         </CardHeader>
209:         <CardContent className="space-y-4 col-span-2">
210:             <Button
211:                 variant="outline"
212:                 onClick={() => setShowImageForm(!showImageForm)}
213:                 className="w-full"
214:             >
215:                 <Upload className="h-4 w-4 mr-2" />
216:                 {showImageForm ? "Hide Upload Form" : "Upload New Image"}
217:             </Button>
218:             <Dialog>
219:                 <DialogTrigger asChild>
220:                     <Button
221:                         variant="outline"
222:                         size="lg"
223:                         className="bg-black text-white"
224:                     >
225:                         Available Images
226:                     </Button>
227:                 </DialogTrigger>
228:
229:                 <DialogContent className="max-w-2xl max-h-[80vh] overflow-y-auto">
230:                     <DialogHeader>
231:                         <DialogTitle>Uploaded Images</DialogTitle>
232:                         <DialogDescription>
233:                             Click the trash icon to delete an image. This action is
234:                             irreversible.
235:                         </DialogDescription>
236:                     </DialogHeader>
237:
238:                     <div className="divide-y divide-gray-200 mt-4">
239:                         {Object.entries(customImages).map(([label, url]) => {
240:                             const fileName = url.split("/").pop();
241:
242:                             return (
243:                                 <div
244:                                     key={label}
245:                                     className="flex items-center justify-between py-3 gap-4"
246:                                 >
247:                                     <div className="flex items-center gap-4 overflow-hidden">
248:                                         <Image
249:                                             src={url}
250:                                             alt={label}
251:                                             width={48}
252:                                             height={48}
253:                                             className="rounded border object-cover w-12 h-12"
254:                                         />
255:                                         <div className="text-sm font-medium truncate">
256:                                             {label}
257:                                         </div>
258:                                     </div>
259:
260:                                     <Button
261:                                         size="icon"
262:                                         variant="destructive"
263:                                         className="h-8 w-8"
264:                                         onClick={async () => {
265:                                             if (!fileName) return;
266:
267:                                             const confirmed = confirm(
268:                                                 `Are you sure you want to delete "${fileName}"?`
269:                                             );
270:                                             if (!confirmed) return;
271:
272:                                             try {

```



```

273:         const res = await fetch("/api/digestive-images", {
274:             method: "DELETE",
275:             headers: { "Content-Type": "application/json" },
276:             body: JSON.stringify({ file: fileName }),
277:         });
278:
279:         const result = await res.json();
280:         if (result.success) {
281:             const updated = { ...customImages };
282:             delete updated[label];
283:             setCustomImages(updated);
284:         } else {
285:             alert("Delete failed");
286:         }
287:     } catch (err) {
288:         console.error("Failed to delete image", err);
289:         alert("Delete failed");
290:     }
291: }}
292: >
293:     <Trash2 className="h-4 w-4" />
294:   </Button>
295: </div>
296:   );
297: }}}
298: </div>
299:
300: { /* <div className="mt-4 text-right">
301:   <DialogClose asChild>
302:     <Button variant="outline">Close</Button>
303:   </DialogClose>
304: </div> */}
305: </DialogContent>
306: </Dialog>
307:
308: {showImageForm && (
309:   <div className="space-y-4 p-4 bg-gray-50 rounded-lg">
310:     <div className="space-y-2">
311:       <Label className="text-sm font-medium">Image Label</Label>
312:       <Input
313:         type="text"
314:         placeholder="e.g. gut, stomach, liver"
315:         value={imageLabel}
316:         onChange={(e) => setImageLabel(e.target.value)}
317:       />
318:     </div>
319:
320:     <div className="space-y-2">
321:       <Label className="text-sm font-medium">Select Image</Label>
322:       <Input
323:         type="file"
324:         accept="image/*"
325:         onChange={handleFileChange}
326:       />
327:     </div>
328:
329:     {imagePreview && (
330:       <div className="space-y-2">
331:         <Label className="text-sm font-medium">Preview</Label>
332:         <div className="relative inline-block">
333:           <Image
334:             src={imagePreview}
335:             alt="Preview"
336:             width={120}
337:             height={120}
338:             className="rounded-lg border-2 border-gray-200 object-cover"
339:           />
340:           <Button
341:             variant="destructive"
342:             size="sm"
343:             className="absolute -top-2 -right-2 h-6 w-6 rounded-full p-0"
344:             onClick={clearImagePreview}
345:           >

```

```

346:         <X className="h-3 w-3" />
347:     </Button>
348: </div>
349: </div>
350: )}
351:
352:     <div className="flex w-fit gap-2">
353:         <Button onClick={handleImageUpload} className="flex-1">
354:             Upload Image
355:         </Button>
356:     </div>
357: </div>
358: )}
359: </CardContent>
360: </Card>
361: </div>
362:
363: { /* Add New Condition Card */}
364: <Card className="shadow-md">
365:     <CardHeader className="pb-3">
366:         <CardTitle className="text-lg text-gray-800">
367:             Add New Condition
368:         </CardTitle>
369:     </CardHeader>
370:     <CardContent>
371:         <div className="flex flex-col sm:flex-row gap-3">
372:             <div className="flex-1">
373:                 <Input
374:                     value={newFieldName}
375:                     onChange={(e) => setNewFieldName(e.target.value)}
376:                     placeholder="e.g., Gut Health, Digestion, Metabolism"
377:                     className="w-full"
378:                 />
379:             </div>
380:             <Button
381:                 onClick={handleAddNewField}
382:                 className="bg-red-600 hover:bg-red-700 sm:w-auto w-full"
383:             >
384:                 <Plus className="w-4 h-4 mr-2" />
385:                 Add Condition
386:             </Button>
387:         </div>
388:     </CardContent>
389: </Card>
390:
391: { /* Data Management Card */}
392: <Card className="shadow-md">
393:     <CardHeader className="pb-3">
394:         <CardTitle className="text-lg md:text-xl text-gray-800">
395:             Conditions Management
396:         </CardTitle>
397:     </CardHeader>
398:     <CardContent className="p-0">
399:         { /* Desktop Table View - Hidden on mobile and tablet */}
400:         <div className="hidden xl:block overflow-x-auto">
401:             <div className="grid grid-cols-11 bg-red-500 rounded text-white text-sm font-semibold">
402:                 <div className="col-span-2 text-center p-3">Title</div>
403:                 <div className="col-span-2 text-center p-3">Icon</div>
404:                 <div className="col-span-2 text-center p-3">Sensitivity</div>
405:                 <div className="col-span-2 text-center p-3">Low</div>
406:                 <div className="col-span-2 text-center p-3">High</div>
407:                 <div className="col-span-1 text-center p-3">Actions</div>
408:             </div>
409:
410:             {Object.entries(digestiveHealth.data).map(([key, entry]) => {
411:                 const iconUrl = customImages[entry.icon?.toLowerCase() || ""];
412:
413:                 return (
414:                     <div
415:                         key={key}
416:                         className="grid grid-cols-11 gap-3 items-center border-b border-gray-200 p-3 hover:bg-gray-100"
417:                     >
418:                         { /* Title */}

```

```

419:         <div className="col-span-2">
420:             <Input
421:                 value={entry.title}
422:                 onChange={(e) =>
423:                     updateField(key, { title: e.target.value })
424:                 }
425:                 className="border-gray-200 focus:border-red-300"
426:             />
427:         </div>
428:
429:         { /* Icon */ }
430:         <div className="col-span-2">
431:             <Select
432:                 value={entry.icon}
433:                 onChange={(val) => updateField(key, { icon: val })}
434:             >
435:                 <SelectTrigger className="text-xs h-9 border-gray-200 focus:border-red-300">
436:                     <SelectValue placeholder="Select icon" />
437:                 </SelectTrigger>
438:                 <SelectContent>
439:                     {Object.entries(customImages).map(
440:                         ([imgKey, imgUrl]) => (
441:                             <SelectItem key={imgKey} value={imgKey}>
442:                                 <div className="flex items-center gap-2">
443:                                     <Image
444:                                         src={imgUrl}
445:                                         alt={imgKey}
446:                                         width={20}
447:                                         height={20}
448:                                         className="rounded object-cover"
449:                                     />
450:                                     <span>{imgKey.toUpperCase()}</span>
451:                                 </div>
452:                             </SelectItem>
453:                         )
454:                     )}
455:                 </SelectContent>
456:             </Select>
457:         </div>
458:
459:         { /* Sensitivity */ }
460:         <div className="col-span-2">
461:             <Select
462:                 value={entry.sensitivity}
463:                 onChange={(val) =>
464:                     updateField(key, { sensitivity: val as "Low" | "High" })
465:                 }
466:             >
467:                 <SelectTrigger className="text-xs h-9 border-gray-200 focus:border-red-300">
468:                     <SelectValue placeholder="Select sensitivity" />
469:                 </SelectTrigger>
470:                 <SelectContent>
471:                     <SelectItem value="Low">Low</SelectItem>
472:                     <SelectItem value="High">High</SelectItem>
473:                 </SelectContent>
474:             </Select>
475:         </div>
476:
477:         { /* Good Textarea */ }
478:         <div className="col-span-2">
479:             <Input
480:                 value={entry.good}
481:                 onChange={(e) =>
482:                     updateField(key, { good: e.target.value })
483:                 }
484:                 className="border-gray-200 focus:border-red-300 resize-none"
485:                 placeholder="Good factors..."
486:             />
487:         </div>
488:
489:         { /* Bad Textarea */ }
490:         <div className="col-span-2">
491:             <Input

```

```

492:         value={entry.bad}
493:         onChange={(e) =>
494:             updateField(key, { bad: e.target.value })
495:         }
496:         className="border-gray-200 focus:border-red-300 resize-none"
497:         placeholder="Bad factors..."
498:     />
499: </div>
500:
501:     {/* Delete Button */}
502:     <div className="col-span-1 flex justify-center">
503:         <Button
504:             variant="ghost"
505:             size="icon"
506:             className="w-8 h-8 bg-red-500 text-white"
507:             onClick={() => {
508:                 setDeleteKey(key);
509:                 setShowDeleteModal(true);
510:             }}
511:         >
512:             <Trash2 className="w-4 h-4" />
513:         </Button>
514:     </div>
515: </div>
516: );
517: }}}
518: </div>
519:
520:     {/* Tablet List View */}
521:     <div className="hidden md:block xl:hidden">
522:         <div className="bg-red-500 text-white p-4 text-sm font-semibold">
523:             Conditions Management
524:         </div>
525:         <div className="space-y-3 p-4">
526:             {Object.entries(digestiveHealth.data).map(([key, entry]) => {
527:                 const imageUrl = customImages[entry.icon?.toLowerCase() || ""];
528:
529:                 return (
530:                     <div
531:                         key={key}
532:                         className="border border-gray-200 rounded-lg p-4 hover:bg-gray-50 relative"
533:                     >
534:                         {/* Delete Button */}
535:                         <Button
536:                             variant="ghost"
537:                             size="icon"
538:                             className="absolute top-2 right-2 w-8 h-8 text-red-600 hover:bg-red-50"
539:                             onClick={() => {
540:                                 setDeleteKey(key);
541:                                 setShowDeleteModal(true);
542:                             }}
543:                         >
544:                             <Trash2 className="w-4 h-4" />
545:                         </Button>
546:
547:                         <div className="grid grid-cols-2 gap-4 pr-10">
548:                             {/* Left Column */}
549:                             <div className="space-y-4">
550:                                 <div>
551:                                     <Label className="text-sm font-medium text-gray-700 mb-1 block">
552:                                         Title
553:                                     </Label>
554:                                     <Input
555:                                         value={entry.title}
556:                                         onChange={(e) =>
557:                                             updateField(key, { title: e.target.value })
558:                                         }
559:                                     className="border-gray-200 focus:border-red-300"
560:                                 />
561:                             </div>
562:
563:                             <div>
564:                                 <Label className="text-sm font-medium text-gray-700 mb-1 block">

```

```

565:         Icon
566:     </Label>
567:     <div className="flex items-center gap-3">
568:         {iconUrl && (
569:             <Image
570:                 src={iconUrl}
571:                 alt="Icon"
572:                 width={40}
573:                 height={40}
574:                 className="rounded border border-gray-200 object-cover"
575:             />
576:         )}
577:         <Select
578:             value={entry.icon}
579:             onChange={(val) =>
580:                 updateField(key, { icon: val })
581:             }
582:         >
583:             <SelectTrigger className="border-gray-200 focus:border-red-300">
584:                 <SelectValue placeholder="Select icon" />
585:             </SelectTrigger>
586:             <SelectContent>
587:                 {Object.entries(customImages).map(
588:                     ([imgKey, imgUrl]) => (
589:                         <SelectItem key={imgKey} value={imgKey}>
590:                             <div className="flex items-center gap-2">
591:                                 <Image
592:                                     src={imgUrl}
593:                                     alt={imgKey}
594:                                     width={20}
595:                                     height={20}
596:                                     className="rounded object-cover"
597:                                 />
598:                                 <span>{imgKey.toUpperCase()}</span>
599:                             </div>
600:                             </SelectItem>
601:                         )
602:                     )}
603:             </SelectContent>
604:         </Select>
605:     </div>
606: </div>
607:
608: <div>
609:     <Label className="text-sm font-medium text-gray-700 mb-1 block">
610:         Sensitivity
611:     </Label>
612:     <Select
613:         value={entry.sensitivity}
614:         onChange={(val) =>
615:             updateField(key, {
616:                 sensitivity: val as "Low" | "High",
617:             })
618:         }
619:     >
620:         <SelectTrigger className="border-gray-200 focus:border-red-300">
621:             <SelectValue placeholder="Select sensitivity" />
622:         </SelectTrigger>
623:         <SelectContent>
624:             <SelectItem value="Low">Low</SelectItem>
625:             <SelectItem value="High">High</SelectItem>
626:         </SelectContent>
627:     </Select>
628: </div>
629: </div>
630:
631: { /* Right Column */ }
632: <div className="space-y-4">
633:     <div>
634:         <Label className="text-sm font-medium text-green-700 mb-1 block">
635:             Good Factors
636:         </Label>
637:         <Textarea

```

```

638:         value={entry.good}
639:         onChange={(e) =>
640:             updateField(key, { good: e.target.value })
641:         }
642:         rows={3}
643:         className="border-gray-200 focus:border-green-300 resize-none"
644:         placeholder="Enter good factors..."
645:     />
646: </div>
647:
648: <div>
649:     <Label className="text-sm font-medium text-red-700 mb-1 block">
650:         Bad Factors
651:     </Label>
652:     <Textarea
653:         value={entry.bad}
654:         onChange={(e) =>
655:             updateField(key, { bad: e.target.value })
656:         }
657:         rows={3}
658:         className="border-gray-200 focus:border-red-300 resize-none"
659:         placeholder="Enter bad factors..."
660:     />
661: </div>
662: </div>
663: </div>
664: </div>
665:     );
666:   })}
667: </div>
668: </div>
669:
670: { /* Mobile Card View */}
671: <div className="md:hidden space-y-4 p-3 sm:p-4">
672:   {Object.entries(digestiveHealth.data).map(([key, entry]) => {
673:     const iconUrl = customImages[entry.icon?.toLowerCase() || ""];
674:
675:     return (
676:       <Card
677:         key={key}
678:         className="border border-gray-200 shadow-sm relative"
679:       >
680:         <CardContent className="p-4 space-y-4 ">
681:           { /* Icon Section */}
682:           <div className="flex flex-col sm:flex-row items-start sm:items-center gap-4">
683:             <Button
684:               variant="ghost"
685:               size="icon"
686:               className="absolute top-2 bg-red-500 right-2 w-8 h-8 text-white z-10"
687:               onClick={() => {
688:                 setDeleteKey(key);
689:                 setShowDeleteModal(true);
690:               }}
691:             >
692:               <Trash2 className="w-4 h-4" />
693:             </Button>
694:             <div className="flex-shrink-0">
695:               {iconUrl ? (
696:                 <Image
697:                   src={iconUrl}
698:                   alt="Icon"
699:                   width={100}
700:                   height={50}
701:                   className="rounded-lg text-center object-cover"
702:                 />
703:               ) : (
704:                 <div className="w-12 h-12 bg-gray-100 rounded-lg flex items-center justify-center"
705:                   <span className="text-xs text-gray-400">
706:                     No Icon
707:                   </span>
708:                 </div>
709:               )}
710:             </div>

```

```

711:         <div className="flex-1 w-full sm:w-auto">
712:             <Label className="text-sm font-medium text-gray-700">
713:                 Icon Selection
714:             </Label>
715:             <Select
716:                 value={entry.icon}
717:                 onChange={(val) =>
718:                     updateField(key, { icon: val })
719:                 }
720:             >
721:                 <SelectTrigger className="mt-1 border-gray-200 focus:border-red-300">
722:                     <SelectValue placeholder="Select icon" />
723:                 </SelectTrigger>
724:                 <SelectContent>
725:                     {Object.entries(customImages).map(
726:                         ([imgKey, imgUrl]) => (
727:                             <SelectItem key={imgKey} value={imgKey}>
728:                                 <div className="flex items-center gap-2">
729:                                     <Image
730:                                         src={imgUrl}
731:                                         alt={imgKey}
732:                                         width={20}
733:                                         height={20}
734:                                         className="rounded object-cover"
735:                                     />
736:                                     <span>{imgKey.toUpperCase()}</span>
737:                                 </div>
738:                             </SelectItem>
739:                         )
740:                     )}
741:                 </SelectContent>
742:             </Select>
743:         </div>
744:     </div>
745:
746:     { /* Title & Sensitivity */ }
747:     <div className="grid grid-cols-1 gap-4">
748:         <div className="space-y-2">
749:             <Label className="text-sm font-medium text-gray-700">
750:                 Title
751:             </Label>
752:             <Input
753:                 value={entry.title}
754:                 onChange={(e) =>
755:                     updateField(key, { title: e.target.value })
756:                 }
757:                 className="border-gray-200 focus:border-red-300"
758:             />
759:         </div>
760:         <div className="space-y-2">
761:             <Label className="text-sm font-medium text-gray-700">
762:                 Sensitivity
763:             </Label>
764:             <Select
765:                 value={entry.sensitivity}
766:                 onChange={(val) =>
767:                     updateField(key, {
768:                         sensitivity: val as "Low" | "High",
769:                     })
770:             }
771:             >
772:                 <SelectTrigger className="border-gray-200 focus:border-red-300">
773:                     <SelectValue placeholder="Select sensitivity" />
774:                 </SelectTrigger>
775:                 <SelectContent>
776:                     <SelectItem value="Low">Low</SelectItem>
777:                     <SelectItem value="High">High</SelectItem>
778:                 </SelectContent>
779:             </Select>
780:         </div>
781:     </div>
782:
783:     { /* Good & Bad Factors */ }

```

```

784:         <div className="space-y-4">
785:             <div className="space-y-2">
786:                 <Label className="text-sm font-medium text-green-700">
787:                     Good Factors
788:                 </Label>
789:                 <Textarea
790:                     value={entry.good}
791:                     onChange={(e) =>
792:                         updateField(key, { good: e.target.value })
793:                     }
794:                     rows={3}
795:                     className="min-h-[80px] border-gray-200 focus:border-green-300 resize-none"
796:                     placeholder="Enter good factors..."
797:                 />
798:             </div>
799:             <div className="space-y-2">
800:                 <Label className="text-sm font-medium text-red-700">
801:                     Bad Factors
802:                 </Label>
803:                 <Textarea
804:                     value={entry.bad}
805:                     onChange={(e) =>
806:                         updateField(key, { bad: e.target.value })
807:                     }
808:                     rows={3}
809:                     className="min-h-[80px] border-gray-200 focus:border-red-300 resize-none"
810:                     placeholder="Enter bad factors..."
811:                 />
812:             </div>
813:         </div>
814:     </CardContent>
815: </Card>
816: );
817: }}}
818: </div>
819: </CardContent>
820: </Card>
821:
822: { /* Delete Confirmation Modal */ }
823: {showDeleteModal && deleteKey && (
824:     <div className="fixed inset-0 bg-black/50 z-50 flex items-center justify-center p-4">
825:         <div className="bg-white p-6 rounded-lg shadow-xl w-full max-w-md">
826:             <div className="flex items-center gap-3 mb-4">
827:                 <div className="w-10 h-10 bg-red-100 rounded-full flex items-center justify-center">
828:                     <Trash2 className="w-5 h-5 text-red-600" />
829:                 </div>
830:                 <h3 className="text-lg font-semibold text-gray-900">
831:                     Confirm Deletion
832:                 </h3>
833:             </div>
834:             <p className="text-gray-600 mb-6">
835:                 Are you sure you want to delete the condition{" "}
836:                 <strong className="text-gray-900">{deleteKey}</strong>? This
837:                 action cannot be undone.
838:             </p>
839:             <div className="flex gap-3 justify-end">
840:                 <Button
841:                     variant="outline"
842:                     onClick={() => setShowDeleteModal(false)}
843:                 >
844:                     Cancel
845:                 </Button>
846:                 <Button
847:                     onClick={handleDeleteFieldConfirm}
848:                     className="bg-red-600 hover:bg-red-700 text-white"
849:                 >
850:                     Delete
851:                 </Button>
852:             </div>
853:         </div>
854:     </div>
855: )}
856: </div>

```



```
857:   );
858: }
```

## ■ File: components\admin\dynamic-diet-field-admin.tsx

```
=====
1: "use client";
2: import { useState, useEffect } from "react";
3: import { DragDropContext, Droppable, Draggable } from "@hello-pangea/dnd";
4: import { Label } from "@/components/ui/label";
5: import { Input } from "@/components/ui/input";
6: import { Textarea } from "@/components/ui/textarea";
7: import {
8:   Card,
9:   CardContent,
10:  CardDescription,
11:  CardHeader,
12:  CardTitle,
13: } from "@/components/ui/card";
14: import {
15:   Select,
16:   SelectTrigger,
17:   SelectValue,
18:   SelectContent,
19:   SelectItem,
20: } from "../ui/select";
21: import { Button } from "@/components/ui/button";
22: import {
23:   Plus,
24:   Save,
25:   RotateCcw,
26:   Trash2,
27:   GripVertical,
28:   X,
29:   File,
30: } from "lucide-react";
31: import { useToast } from "@/components/ui/use-toast";
32: import {
33:   FieldInput,
34:   FieldTextarea,
35:   FieldSelect,
36: } from "@/components/admin/dynamic-field-helpers";
37: import type {
38:   DynamicDietFieldDefinition,
39:   DynamicDiet,
40:   PatientDietAnalysisResult,
41: } from "@/types/report-types";
42:
43: interface DynamicDietFieldAdminProps {
44:   dynamicDietFieldDefinitions: DynamicDietFieldDefinition[];
45:   onUpdateFields: (definitions: DynamicDietFieldDefinition[]) => void;
46:   onSave: () => Promise<void> | void; // allow async saves
47:   onReset: () => void;
48:   categories: string[];
49:   onUpdateCategories: (updated: string[]) => void;
50:   patientDietAnalysisResults: PatientDietAnalysisResult[];
51:   onUpdateResults: (results: PatientDietAnalysisResult[]) => void;
52: }
53:
54: export default function DynamicDietFieldAdmin({
55:   dynamicDietFieldDefinitions,
56:   onUpdateFields,
57:   onSave,
58:   onReset,
59:   categories,
60:   onUpdateCategories,
61:   patientDietAnalysisResults,
62:   onUpdateResults,
63: }: DynamicDietFieldAdminProps) {
64:   const [showResultModal, setShowResultModal] = useState(false);
65:   const [selectedFieldId, setSelectedFieldId] = useState("");
66:   const [enteredScore, setEnteredScore] = useState<number>(5);
```

```

67: const [fieldScores, setFieldScores] = useState<Record<string, number>>({});
68: const [results, setResults] = useState<PatientDietAnalysisResult[]>(
69:   patientDietAnalysisResults || []
70: );
71: const [existingFieldIds, setExistingFieldIds] = useState<
72:   Record<string, string>
73: >({});
74: const [showResultsModal, setShowResultsModal] = useState(false);
75:
76: // Updated to handle the new structure - get meta from first definition
77: const [quote, setQuote] = useState(
78:   dynamicDietFieldDefinitions?.[0]?.meta?.quote || ""
79: );
80: const [description, setDescription] = useState(
81:   dynamicDietFieldDefinitions?.[0]?.meta?.description || ""
82: );
83:
84: const [localDefinitions, setLocalDefinitions] = useState<
85:   DynamicDietFieldDefinition[]
86: >(dynamicDietFieldDefinitions || []);
87: const [localCategories, setLocalCategories] = useState<string[]>(
88:   categories || []
89: );
90: const [showCreateModal, setShowCreateModal] = useState(false);
91: const [newFieldData, setNewFieldData] = useState<Omit<DynamicDiet, "id">>(
92:   () => ({
93:     label: "",
94:     category: localCategories.length > 0 ? localCategories[0] : "",
95:     min: 1,
96:     max: 10,
97:     highRecommendation: "",
98:     normalRecommendation: "",
99:     lowRecommendation: "",
100:   })
101: );
102: const { toast } = useToast();
103:
104: // Helper function to get all fields from all definitions
105: const getAllFields = (): DynamicDiet[] => {
106:   return localDefinitions.flatMap((def) => def.fields || []);
107: };
108:
109: const handleAddCategory = () => {
110:   const newCategory = prompt("Enter new category name:");
111:   if (newCategory && !localCategories.includes(newCategory)) {
112:     const updated = [...localCategories, newCategory];
113:     setLocalCategories(updated);
114:     onUpdateCategories(updated);
115:   }
116: };
117:
118: const handleDeleteCategory = (cat: string) => {
119:   if (confirm(`Delete "${cat}"?`)) {
120:     const updated = localCategories.filter((c) => c !== cat);
121:     setLocalCategories(updated);
122:     onUpdateCategories(updated);
123:   }
124: };
125:
126: useEffect(() => {
127:   if (localCategories.length > 0 && !newFieldData.category) {
128:     setNewFieldData((prev) => ({
129:       ...prev,
130:       category: localCategories[0],
131:     }));
132:   }
133: }, [localCategories, newFieldData.category]);
134:
135: useEffect(() => {
136:   setLocalDefinitions(dynamicDietFieldDefinitions || []);
137: }, [dynamicDietFieldDefinitions]);
138:
139: useEffect(() => {

```

```

140:     setResults(patientDietAnalysisResults || []);
141:   }, [patientDietAnalysisResults]);
142:
143:   useEffect(() => {
144:     setLocalCategories(categories || []);
145:   }, [categories]);
146:
147:   useEffect(() => {
148:     const newDefinitions = dynamicDietFieldDefinitions || [];
149:     setLocalDefinitions(newDefinitions);
150:
151:     const firstDef = newDefinitions[0];
152:     if (firstDef?.meta) {
153:       setQuote(firstDef.meta.quote || "");
154:       setDescription(firstDef.meta.description || "");
155:     }
156:   }, [dynamicDietFieldDefinitions]);
157:
158:   // Updated to handle meta changes
159:   useEffect(() => {
160:     if (localDefinitions.length > 0) {
161:       const currentMeta = localDefinitions[0]?.meta;
162:       // Only update if the meta actually changed
163:       if (
164:         currentMeta?.quote !== quote ||
165:         currentMeta?.description !== description
166:       ) {
167:         const updatedDefinitions = [...localDefinitions];
168:         updatedDefinitions[0] = {
169:           ...updatedDefinitions[0],
170:           meta: {
171:             quote,
172:             description,
173:           },
174:         };
175:         setLocalDefinitions(updatedDefinitions);
176:         onUpdateFields(updatedDefinitions);
177:       }
178:     }
179:   }, [quote, description]);
180:
181:   const addNewField = () => {
182:     if (!newFieldData.label.trim()) {
183:       toast({
184:         title: "Validation Error",
185:         description: "Field label is required.",
186:         variant: "destructive",
187:         duration: 3000,
188:       });
189:       return;
190:     }
191:
192:     // ? Check if categories exist
193:     if (localCategories.length === 0) {
194:       toast({
195:         title: "Validation Error",
196:         description: "Please add at least one category first.",
197:         variant: "destructive",
198:         duration: 3000,
199:       });
200:       return;
201:     }
202:
203:     // ? Ensure category is set
204:     const categoryToUse = newFieldData.category || localCategories[0];
205:
206:     // Generate a unique ID for the field
207:     const fieldId = crypto.randomUUID();
208:
209:     // Check for duplicates by label across all fields
210:     const allFields = getAllFields();
211:     const isDuplicate = allFields.some(
212:       (field) => field.label.toLowerCase() === newFieldData.label.toLowerCase()

```

```

213:     );
214:
215:     if (isDuplicate) {
216:         toast({
217:             title: "Validation Error",
218:             description: "A field with this label already exists.",
219:             variant: "destructive",
220:             duration: 3000,
221:         });
222:         return;
223:     }
224:
225:     const newField: DynamicDiet = {
226:         id: fieldId,
227:         ...newFieldData,
228:         category: categoryToUse, // ? Ensure category is always set
229:     };
230:
231:     console.log("? FRONTEND: Adding new field:", newField);
232:
233:     // Add to the first definition, or create a new one if none exist
234:     const updatedDefinitions = [...localDefinitions];
235:     if (updatedDefinitions.length === 0) {
236:         updatedDefinitions.push({
237:             meta: { quote: quote || "", description: description || "" }, // ? Use current quote/description
238:             fields: [newField],
239:         });
240:     } else {
241:         updatedDefinitions[0] = {
242:             ...updatedDefinitions[0],
243:             fields: [...(updatedDefinitions[0].fields || []), newField],
244:         };
245:     }
246:
247:     setLocalDefinitions(updatedDefinitions);
248:     onUpdateFields(updatedDefinitions);
249:     setShowCreateModal(false);
250:     setNewFieldData({
251:         label: "",
252:         category: localCategories[0] || "",
253:         min: 1,
254:         max: 10,
255:         highRecommendation: "",
256:         normalRecommendation: "",
257:         lowRecommendation: "",
258:     });
259:     toast({
260:         title: "Field Added",
261:         description: `"$${newField.label}" has been added. Remember to Save Changes.`,
262:         variant: "success",
263:         duration: 3000,
264:     });
265: };
266:
267: const handleCreateModalClose = () => {
268:     setShowCreateModal(false);
269:     setNewFieldData({
270:         label: "",
271:         category: localCategories[0] || "",
272:         min: 1,
273:         max: 10,
274:         highRecommendation: "",
275:         normalRecommendation: "",
276:         lowRecommendation: "",
277:     });
278: };
279:
280: // Updated to handle field updates within definitions
281: const updateLocalField = (
282:     fieldId: string,
283:     key: keyof DynamicDiet,
284:     value: any
285: ) => {

```

```

286:     if (key === "id") {
287:         toast({
288:             title: "Validation Error",
289:             description: "Field ID cannot be modified (it's auto-generated).",
290:             variant: "destructive",
291:             duration: 3000,
292:         });
293:         return;
294:     }
295:
296:     // Check for duplicate labels (excluding current field)
297:     if (key === "label") {
298:         const allFields = getAllFields();
299:         const isDuplicate = allFields.some(
300:             (f) =>
301:                 f.id !== fieldId &&
302:                 f.label.toLowerCase() === String(value).toLowerCase()
303:         );
304:         if (isDuplicate) {
305:             toast({
306:                 title: "Validation Error",
307:                 description: "Field label must be unique.",
308:                 variant: "destructive",
309:                 duration: 3000,
310:             });
311:             return;
312:         }
313:     }
314:
315:     const updatedDefinitions = localDefinitions.map((definition) => ({
316:         ...definition,
317:         fields: (definition.fields || []).map((field) =>
318:             field.id === fieldId ? { ...field, [key]: value } : field
319:         ),
320:     }));
321:
322:     setLocalDefinitions(updatedDefinitions);
323:     onUpdateFields(updatedDefinitions);
324: };
325:
326: const handleBatchAddResults = () => {
327:     const updatedResults: PatientDietAnalysisResult[] = [...results];
328:     const allFields = getAllFields();
329:
330:     Object.entries(fieldScores).forEach(([fieldId, score]) => {
331:         const field = allFields.find((f) => f.id === fieldId);
332:         if (!field || score < field.min || score > field.max) return;
333:
334:         let level: "LOW" | "NORMAL" | "HIGH" = "NORMAL";
335:         if (score <= 3) level = "LOW";
336:         else if (score >= 7) level = "HIGH";
337:
338:         const recommendations = {
339:             LOW: field.lowRecommendation,
340:             NORMAL: field.normalRecommendation,
341:             HIGH: field.highRecommendation,
342:         };
343:
344:         const recommendation = recommendations[level];
345:
346:         const existingIndex = updatedResults.findIndex(
347:             (r) => r.fieldId === fieldId
348:         );
349:
350:         if (existingIndex >= 0) {
351:             // ? Update existing
352:             updatedResults[existingIndex] = {
353:                 ...updatedResults[existingIndex],
354:                 score,
355:                 level,
356:                 recommendation,
357:                 recommendations,
358:                 selectedLevel: level,

```

```

359:     };
360:   } else {
361:     // ? Add new
362:     updatedResults.push({
363:       id: existingFieldIds[fieldId] || undefined,
364:       fieldId,
365:       score,
366:       level,
367:       recommendation,
368:       recommendations,
369:       selectedLevel: level,
370:     } as PatientDietAnalysisResult);
371:   }
372: });
373:
374: setResults(updatedResults);
375: onUpdateResults(updatedResults);
376: setShowResultModal(false);
377:
378: toast({
379:   title: "Results Saved",
380:   description: "Diet analysis results updated successfully.",
381:   variant: "success",
382:   duration: 3000,
383: });
384: };
385:
386: const handleDeleteResult = (fieldId: string) => {
387:   const updated = results.filter((r) => r.fieldId !== fieldId);
388:   setResults(updated);
389:   onUpdateResults(updated);
390:
391:   toast({
392:     title: "Result Deleted",
393:     description: `Analysis result for "${fieldId}" has been removed.`,
394:     variant: "destructive",
395:     duration: 3000,
396:   });
397: };
398:
399: const deleteField = (fieldId: string) => {
400:   if (
401:     !window.confirm(
402:       "Are you sure you want to delete this field? This action cannot be undone."
403:     )
404:   ) {
405:     return;
406:   }
407:
408:   const updatedDefinitions = localDefinitions.map((definition) => ({
409:     ...definition,
410:     fields: (definition.fields || []).filter((field) => field.id !== fieldId),
411:   }));
412:
413:   setLocalDefinitions(updatedDefinitions);
414:   onUpdateFields(updatedDefinitions);
415:   toast({
416:     title: "Field Deleted",
417:     description: "Field removed. Remember to Save Changes.",
418:     variant: "destructive",
419:     duration: 3000,
420:   });
421: };
422:
423: // Updated drag and drop to work with flattened fields
424: const onDragEnd = (result: any) => {
425:   if (!result.destination) return;
426:
427:   const allFields = getAllFields();
428:   const reordered = Array.from(allFields);
429:   const [moved] = reordered.splice(result.source.index, 1);
430:   reordered.splice(result.destination.index, 0, moved);
431:

```

```

432: // Update the first definition with reordered fields
433: const updatedDefinitions = [...localDefinitions];
434: if (updatedDefinitions.length > 0) {
435:   updatedDefinitions[0] = {
436:     ...updatedDefinitions[0],
437:     fields: reordered,
438:   };
439: }
440:
441: setLocalDefinitions(updatedDefinitions);
442: onUpdateFields(updatedDefinitions);
443: toast({
444:   title: "Fields Reordered",
445:   description: "Order updated. Remember to Save Changes.",
446:   variant: "default",
447:   duration: 3000,
448: });
449: };
450:
451: // New: ensure latest local state is pushed to parent before calling onSave
452: const handleSaveAll = async () => {
453:   console.log("? FRONTEND: About to save data:");
454:   console.log(" - Local definitions:", localDefinitions.length);
455:   console.log(" - Local categories:", localCategories);
456:   console.log(" - Results:", results.length);
457:
458:   // ? Merge latest field data and ensure IDs
459:   const mergedDefinitions = localDefinitions.map((def) => ({
460:     ...def,
461:     fields: (Array.isArray(def.fields) ? def.fields : []).map((f) => ({
462:       ...f,
463:       id: f.id || crypto.randomUUID(), // ensure stable ID for backend
464:     })),
465:   }));
466:
467:   console.log(
468:     "? FRONTEND: Sending definitions:",
469:     JSON.stringify(mergedDefinitions, null, 2)
470:   );
471:
472:   // Push updated values to parent state
473:   onUpdateFields(mergedDefinitions);
474:   onUpdateResults(results);
475:   onUpdateCategories(localCategories);
476:
477:   try {
478:     const maybePromise = onSave();
479:     if (maybePromise && typeof (maybePromise as any).then === "function") {
480:       await (maybePromise as Promise<any>);
481:     }
482:     toast({
483:       title: "Saved",
484:       description: "All changes were submitted to the server.",
485:       variant: "success",
486:       duration: 1500,
487:     });
488:   } catch (err) {
489:     console.error("? FRONTEND: Save failed:", err);
490:     toast({
491:       title: "Save failed",
492:       description:
493:         "Something went wrong while saving. Check console or server logs.",
494:       variant: "destructive",
495:       duration: 1500,
496:     });
497:   }
498: };
499:
500: return (
501:   <div>
502:     <Card className="shadow-lg border-0 bg-white">
503:       <CardHeader className="bg-gradient-to-r from-amber-500 to-orange-600 rounded-lg p-4 sm:p-6 uppercase">
504:         <CardTitle className="text-base sm:text-xl md:text-2xl font-bold flex items-center justify-center">

```

```

505:         Dynamic Diet Field Management
506:     </CardTitle>
507: </CardHeader>
508:
509: <CardContent className="space-y-6 sm:space-y-8 p-4 sm:p-6 lg:p-8">
510:     { /* Header Section */ }
511:     <div className="flex flex-col sm:flex-row gap-4">
512:         <h3 className="text-lg sm:text-xl font-semibold text-gray-800">
513:             Configured Fields
514:         </h3>
515:
516:         { /* Action Buttons */ }
517:         <div className="flex flex-row flex-wrap justify-end gap-2 w-full">
518:             <Button
519:                 onClick={() => setShowCreateModal(true)}
520:                 size="sm"
521:                 className="flex-1 sm:flex-initial min-w-[160px]"
522:             >
523:                 Add New Field
524:             </Button>
525:
526:             <Button
527:                 className="flex-1 sm:flex-initial min-w-[160px]"
528:                 onClick={() => {
529:                     const scores: Record<string, number> = {};
530:                     const ids: Record<string, string> = {};
531:
532:                     (patientDietAnalysisResults || []).forEach((res) => {
533:                         scores[res.fieldId] = res.score;
534:                         ids[res.fieldId] = (res as any).id ?? "";
535:                     });
536:
537:                     setFieldScores(scores);
538:                     setExistingFieldIds(ids);
539:                     setShowResultModal(true);
540:                 }}
541:                 size="sm"
542:             >
543:                 Add Analysis Result
544:             </Button>
545:
546:             <Button
547:                 onClick={() => setShowResultsModal(true)}
548:                 size="sm"
549:                 className="flex-1 sm:flex-initial min-w-[160px]"
550:             >
551:                 View Saved Results
552:             </Button>
553:
554:             <Button
555:                 onClick={handleSaveAll}
556:                 size="sm"
557:                 className="flex-1 sm:flex-initial min-w-[160px]"
558:             >
559:                 Save All Changes
560:             </Button>
561:         </div>
562:     </div>
563:
564:     { /* Categories Management Section */ }
565:     <div className="bg-gray-50 rounded-lg p-4 sm:p-6">
566:         <h4 className="font-semibold text-gray-700 mb-4 text-base sm:text-lg">
567:             Diet Field Categories
568:         </h4>
569:
570:         <div className="space-y-3">
571:             { /* Categories List */ }
572:             <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-2">
573:                 {localCategories.map((cat) => (
574:                     <div
575:                         key={cat}
576:                         className="flex justify-between items-center bg-white px-3 py-2 rounded-md shadow-sm bo
577:                     >

```



```

578:         <span className="text-sm font-medium truncate mr-2">
579:             {cat}
580:         </span>
581:         <Button
582:             variant="destructive"
583:             size="sm"
584:             onClick={() => handleDeleteCategory(cat)}
585:             className="flex-shrink-0"
586:         >
587:             <Trash2 className="h-3 w-3" />
588:         </Button>
589:     </div>
590:   )}
591: </div>
592:
593:   <Button
594:     onClick={handleAddCategory}
595:     size="sm"
596:     className="w-full sm:w-auto"
597:   >
598:     <Plus className="h-4 w-4 mr-2" />
599:     Add Category
600:   </Button>
601: </div>
602: </div>
603:
604: <div className="mb-6">
605:   <Label>Quote</Label>
606:   <Input
607:     value={quote}
608:     onChange={(e) => setQuote(e.target.value)}
609:     placeholder="Enter inspirational quote..."
610:     className="mb-4"
611:   />
612:
613:   <Label>Description</Label>
614:   <Input
615:     value={description}
616:     onChange={(e) => setDescription(e.target.value)}
617:     placeholder="Brief description or note for this section..."
618:   />
619: </div>
620:
621: { /* Fields List */ }
622: {getAllFields().length === 0 ? (
623:   <div className="text-center py-12 text-gray-500 italic bg-gray-50 rounded-lg">
624:     <div className="text-4xl mb-4"></div>
625:     <p className="text-lg">No dynamic diet fields defined yet.</p>
626:     <p className="text-sm mt-2">
627:       Click "Add New Field" to get started.
628:     </p>
629:   </div>
630: ) : (
631:   <DragDropContext onDragEnd={onDragEnd}>
632:     <Draggable droppableId="fields">
633:       {(provided) => (
634:         <div
635:           {...provided.droppableProps}
636:           ref={provided.innerRef}
637:           className="space-y-4"
638:         >
639:           {getAllFields().map((field, index) => (
640:             <Draggable
641:               key={field.id}
642:               draggableId={field.id}
643:               index={index}
644:             >
645:               {(provided, snapshot) => (
646:                 <div
647:                   ref={provided.innerRef}
648:                   {...provided.draggableProps}
649:                   className={`border rounded-lg p-4 sm:p-6 bg-white shadow-sm transition-all duration-200 ${
650:                     snapshot.isDragging

```

```

651:             ? "bg-gray-100 ring-2 ring-blue-500 shadow-lg"
652:             : "hover:shadow-md"
653:         }` }
654:     >
655:     { /* Field Header */ }
656:     <div className="flex flex-col sm:flex-row items-start sm:items-center justify-b
657:         <div className="flex items-center gap-2 min-w-0 flex-1">
658:             <span
659:                 { ...provided.dragHandleProps }
660:                 className="cursor-grab text-gray-400 flex-shrink-0 touch-none"
661:             >
662:                 <GripVertical className="h-5 w-5" />
663:             </span>
664:             <div className="min-w-0 flex-1">
665:                 <h4 className="text-base sm:text-lg font-semibold text-gray-900 truncate">
666:                     { index + 1 }. { field.label }
667:                 </h4>
668:             </div>
669:         </div>
670:         <Button
671:             variant="destructive"
672:             size="sm"
673:             onClick={() => deleteField(field.id)}
674:             className="flex-shrink-0"
675:         >
676:             <Trash2 className="h-4 w-4" />
677:             <span className="hidden sm:inline ml-2">
678:                 Delete
679:             </span>
680:         </Button>
681:     </div>
682:
683:     { /* Field Form */ }
684:     <div className="space-y-4">
685:         { /* Basic Info Row */ }
686:         <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-4 gap-4">
687:             <FieldInput
688:                 label="Label"
689:                 value={ field.label }
690:                 onChange={ (val) =>
691:                     updateLocalField(field.id, "label", val)
692:                 }
693:             />
694:
695:             { /* Fixed Category Select with label */ }
696:             <div className="flex flex-col">
697:                 <label className="mb-1 text-sm font-medium text-gray-700">
698:                     Category
699:                 </label>
700:                 <Select
701:                     value={ field.category }
702:                     onValueChange={ (val) =>
703:                         updateLocalField(
704:                             field.id,
705:                             "category",
706:                             val
707:                         )
708:                     }
709:                 >
710:                     <SelectTrigger className="w-full">
711:                         <SelectValue placeholder="Select Category" />
712:                     </SelectTrigger>
713:                     <SelectContent>
714:                         { localCategories.map((cat) => (
715:                             <SelectItem key={ cat } value={ cat }>
716:                                 { cat }
717:                             </SelectItem>
718:                         )) }
719:                     </SelectContent>
720:                 </Select>
721:             </div>
722:
723:             <FieldInput

```

```

724:         label="Min Score"
725:         type="number"
726:         value={field.min}
727:         onChange={(val) =>
728:             updateLocalField(field.id, "min", val)
729:         }
730:         min={0}
731:         max={field.max}
732:     />
733:     <FieldInput
734:         label="Max Score"
735:         type="number"
736:         value={field.max}
737:         onChange={(val) =>
738:             updateLocalField(field.id, "max", val)
739:         }
740:         min={field.min}
741:     />
742: </div>
743:
744: { /* Recommendations Row */}
745: <div className="grid grid-cols-1 lg:grid-cols-3 gap-4">
746:     <FieldTextarea
747:         label="High Recommendation"
748:         value={field.highRecommendation}
749:         onChange={(val) =>
750:             updateLocalField(
751:                 field.id,
752:                 "highRecommendation",
753:                 val
754:             )
755:         }
756:     />
757:     <FieldTextarea
758:         label="Normal Recommendation"
759:         value={field.normalRecommendation}
760:         onChange={(val) =>
761:             updateLocalField(
762:                 field.id,
763:                 "normalRecommendation",
764:                 val
765:             )
766:         }
767:     />
768:     <FieldTextarea
769:         label="Low Recommendation"
770:         value={field.lowRecommendation}
771:         onChange={(val) =>
772:             updateLocalField(
773:                 field.id,
774:                 "lowRecommendation",
775:                 val
776:             )
777:         }
778:     />
779: </div>
780: </div>
781: </div>
782: )}
783: </Draggable>
784: )}
785: {provided.placeholder}
786: </div>
787: )}
788: </Draggable>
789: </DragDropContext>
790: )}
791: </CardContent>
792: </Card>
793:
794: { /* Create New Field Modal */}
795: {showResultModal && (
796:     <div className="fixed inset-0 z-50 bg-black bg-opacity-50 flex justify-center items-center p-4">

```

```

797:     <div className="bg-white rounded-lg shadow-xl w-full max-w-3xl p-6">
798:       <h3 className="text-lg font-bold mb-4 text-center">
799:         Add Diet Analysis Results
800:       </h3>
801:
802:       <div className="grid grid-cols-1 sm:grid-cols-2 gap-4 max-h-[60vh] overflow-y-auto mb-6">
803:         {getAllFields().map((field) => (
804:           <div
805:             key={field.id}
806:             className="flex items-center justify-between"
807:           >
808:             <label className="mr-4 w-2/3 font-medium">
809:               {field.label}
810:             </label>
811:             <input
812:               type="number"
813:               min={field.min}
814:               max={field.max}
815:               className="w-20 border rounded px-2 py-1"
816:               value={fieldScores[field.id] ?? ""}
817:               onChange={(e) =>
818:                 setFieldScores((prev) => ({
819:                   ...prev,
820:                   [field.id]: Number(e.target.value),
821:                 }))
822:             >
823:           </div>
824:         </div>
825:       )]}
826:     </div>
827:
828:     <div className="flex justify-end gap-2">
829:       <Button
830:         variant="outline"
831:         onClick={() => setShowResultModal(false)}
832:       >
833:         Cancel
834:       </Button>
835:       <Button onClick={handleBatchAddResults}>Save All</Button>
836:     </div>
837:   </div>
838: </div>
839: )}
840:
841: {showCreateModal && (
842:   <div className="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center z-50 p-2 sm:p-4">
843:     <div className="bg-white rounded-lg w-full max-w-2xl shadow-xl">
844:       <div className="flex items-center justify-between p-3 sm:p-4 md:p-6 border-b">
845:         <h3 className="text-base sm:text-lg md:text-xl font-bold">
846:           Add New Diet Field Definition
847:         </h3>
848:         <Button
849:           variant="ghost"
850:           size="sm"
851:           onClick={handleCreateModalClose}
852:           className="flex-shrink-0"
853:         >
854:           <X className="h-4 w-4" />
855:         </Button>
856:       </div>
857:
858:       <div className="p-2 sm:p-4 md:p-6">
859:         <div className="space-y-4">
860:           <div className="grid grid-cols-1 sm:grid-cols-2 gap-3 sm:gap-4">
861:             <FieldInput
862:               label="Field Label"
863:               value={newFieldData.label}
864:               onChange={(val) =>
865:                 setNewFieldData({ ...newFieldData, label: String(val) })
866:             >
867:           </div>
868:         </div>
869:       </div>
870:     </div>
871:   </div>
872: )}

```

```

870:         placeholder="e.g., Gluten Sensitivity"
871:         required
872:     />
873:
874:     <div className="flex flex-col space-y-1">
875:         <label className="text-sm font-medium">Category</label>
876:         <Select
877:             value={newFieldData.category}
878:             onChange={(val) =>
879:                 setNewFieldData({ ...newFieldData, category: val })
880:             }
881:         >
882:             <SelectTrigger className="w-full">
883:                 <SelectValue placeholder="Select category" />
884:             </SelectTrigger>
885:             <SelectContent>
886:                 {localCategories.map((cat) => (
887:                     <SelectItem key={cat} value={cat}>
888:                         {cat}
889:                     </SelectItem>
890:                 ))}
891:             </SelectContent>
892:         </Select>
893:     </div>
894: </div>
895:
896: {/* Score Range */}
897: <div className="grid grid-cols-1 sm:grid-cols-2 gap-3 sm:gap-4">
898:     <FieldInput
899:         label="Min Score"
900:         type="number"
901:         value={newFieldData.min}
902:         onChange={(val) =>
903:             setNewFieldData({ ...newFieldData, min: Number(val) })
904:         }
905:         min={0}
906:         max={newFieldData.max}
907:     />
908:     <FieldInput
909:         label="Max Score"
910:         type="number"
911:         value={newFieldData.max}
912:         onChange={(val) =>
913:             setNewFieldData({ ...newFieldData, max: Number(val) })
914:         }
915:         min={newFieldData.min}
916:     />
917: </div>
918:
919: {/* Recommendations */}
920: <div className="space-y-4">
921:     <FieldTextarea
922:         label="High Recommendation"
923:         value={newFieldData.highRecommendation}
924:         onChange={(val) =>
925:             setNewFieldData({
926:                 ...newFieldData,
927:                 highRecommendation: val,
928:             })
929:         }
930:     />
931:     <FieldTextarea
932:         label="Normal Recommendation"
933:         value={newFieldData.normalRecommendation}
934:         onChange={(val) =>
935:             setNewFieldData({
936:                 ...newFieldData,
937:                 normalRecommendation: val,
938:             })
939:         }
940:     />
941:     <FieldTextarea
942:         label="Low Recommendation"

```

```

943:         value={newFieldData.lowRecommendation}
944:         onChange={ (val) =>
945:             setNewFieldData({
946:                 ...newFieldData,
947:                 lowRecommendation: val,
948:             })
949:         }
950:     />
951: </div>
952: </div>
953: </div>
954:
955: { /* Modal Footer */ }
956: <div className="flex flex-col sm:flex-row justify-end gap-2 sm:gap-3 p-3 sm:p-4 md:p-6 border-t
957:     <Button
958:         variant="outline"
959:         onClick={handleCreateModalClose}
960:         className="w-full sm:w-auto bg-transparent order-2 sm:order-1"
961:     >
962:         Cancel
963:     </Button>
964:     <Button
965:         onClick={addNewField}
966:         className="w-full sm:w-auto order-1 sm:order-2"
967:     >
968:         Add Field
969:     </Button>
970: </div>
971: </div>
972: </div>
973: )}
974:
975: {showResultsModal && (
976:     <div className="fixed inset-0 z-50 bg-black bg-opacity-50 flex justify-center items-center p-4">
977:         <div className="bg-white rounded-lg shadow-xl w-full max-w-3xl p-4 max-h-[80vh] overflow-y-auto r
978:             <button
979:                 onClick={() => setShowResultsModal(false)}
980:                 className="absolute top-3 left-3 text-gray-500 hover:text-black"
981:             >
982:                 <X className="h-5 w-5" />
983:             </button>
984:
985:             { /* Delete All Button */ }
986:             <div className="mb-4 flex justify-end">
987:                 <Button
988:                     variant="destructive"
989:                     onClick={() => {
990:                         if (
991:                             confirm(
992:                                 "Are you sure you want to delete all analysis results?"
993:                             )
994:                         ) {
995:                             setResults([]);
996:                             onUpdateResults([]);
997:                             toast({
998:                                 title: "All Results Deleted",
999:                                 description:
1000:                                     "All diet analysis results have been cleared.",
1001:                                 variant: "destructive",
1002:                             });
1003:                         }
1004:                     }}
1005:                 >
1006:                     <Trash2 className="h-4 w-4 mr-2" />
1007:                     Delete All Results
1008:                 </Button>
1009:             </div>
1010:
1011:             {results.length === 0 ? (
1012:                 <p className="text-gray-500 italic text-center">
1013:                     No analysis results yet.
1014:                 </p>
1015:             ) : (

```

```

1016:         <div className="space-y-3">
1017:             {results.map((res) => (
1018:                 <div
1019:                     key={res.fieldId}
1020:                     className="flex justify-between items-center border p-3 rounded-md bg-gray-50"
1021:                 >
1022:                     <div>
1023:                         { /* <p className="font-medium">{res.fieldId}</p> */}
1024:
1025:                         <p className="text-sm text-gray-600">
1026:                             Score: {res.score}, Level: {res.level}
1027:                         </p>
1028:                         <p className="text-sm text-gray-600">
1029:                             Recommendation: {res.recommendation}
1030:                         </p>
1031:                     </div>
1032:                     <Button
1033:                         variant="destructive"
1034:                         size="sm"
1035:                         onClick={() => handleDeleteResult(res.fieldId)}
1036:                     >
1037:                         <Trash2 className="h-4 w-4" />
1038:                     </Button>
1039:                 </div>
1040:             )})}
1041:         </div>
1042:     )}
1043: </div>
1044: </div>
1045: )}
1046: </div>
1047: );
1048: }

```

## **File: components\admin\dynamic-field-helpers.tsx**

```

1: "use client"
2:
3: import type React from "react"
4:
5: interface FieldInputProps {
6:   label: string
7:   value: string | number | undefined
8:   onChange: (value: string | number) => void
9:   type?: React.HTMLInputTypeAttribute
10: disabled?: boolean
11: placeholder?: string
12: required?: boolean
13: min?: number
14: max?: number
15: }
16:
17: export const FieldInput: React.FC<FieldInputProps> = ({
18:   label,
19:   value,
20:   onChange,
21:   type = "text",
22:   disabled = false,
23:   placeholder = "",
24:   required = false,
25:   min,
26:   max,
27: }) => (
28:   <div className="flex flex-col">
29:     <label className="text-sm font-medium text-gray-700 mb-1">{label}</label>
30:     <input
31:       type={type}
32:       disabled={disabled}
33:       placeholder={placeholder}
34:       required={required}
35:       min={min}

```

```

36:         max={max}
37:         className={`w-full border rounded-md px-3 py-2 text-sm focus:outline-none focus:ring-2 focus:ring-blue-500
38:             disabled ? "bg-gray-100 text-gray-500 cursor-not-allowed" : "bg-white border-gray-300 hover:border-gray-300"}
39:     `}
40:     value={value ?? ""}
41:     onChange={(e) => onChange(type === "number" ? Number(e.target.value) : e.target.value)}
42:   />
43: </div>
44: )
45:
46: interface FieldTextareaProps {
47:   label: string
48:   value: string | undefined
49:   onChange: (value: string) => void
50: }
51:
52: export const FieldTextarea: React.FC<FieldTextareaProps> = ({ label, value, onChange }) => (
53:   <div className="flex flex-col">
54:     <label className="text-sm font-medium text-gray-700 mb-1">{label}</label>
55:     <textarea
56:       rows={3}
57:       className="w-full border border-gray-300 rounded-md px-3 py-2 text-sm focus:outline-none focus:ring-2 focus:ring-blue-500
58:         disabled ? "bg-gray-100 text-gray-500 cursor-not-allowed" : "bg-white border-gray-300 hover:border-gray-300"}
59:       value={value ?? ""}
60:       onChange={(e) => onChange(e.target.value)}
61:     />
62:   </div>
63: )
64:
65: interface FieldSelectProps {
66:   label: string
67:   value: string | undefined
68:   options: string[]
69:   onChange: (value: string) => void
70: }
71:
72: export const FieldSelect: React.FC<FieldSelectProps> = ({ label, value, options, onChange }) => (
73:   <div className="flex flex-col">
74:     <label className="text-sm font-medium text-gray-700 mb-1">{label}</label>
75:     <select
76:       className="w-full border border-gray-300 rounded-md px-3 py-2 text-sm focus:outline-none focus:ring-2 focus:ring-blue-500
77:         disabled ? "bg-gray-100 text-gray-500 cursor-not-allowed" : "bg-white border-gray-300 hover:border-gray-300"}
78:       value={value ?? ""}
79:       onChange={(e) => onChange(e.target.value)}
80:     >
81:       <option value="">? Select Category ?</option>
82:       {options.map((opt, i) => (
83:         <option key={i} value={opt}>
84:           {opt}
85:         </option>
86:       ))}
87:     </select>
88:   </div>
89: )

```

## File: components\admin\family-genetic-impact-admin.tsx

```

1: "use client";
2:
3: import {
4:   Card,
5:   CardContent,
6:   CardDescription,
7:   CardHeader,
8:   CardTitle,
9: } from "@components/ui/card";
10: import { Input } from "@components/ui/input";
11: import { Label } from "@components/ui/label";
12: import { Textarea } from "@components/ui/textarea";
13: import { Button } from "@components/ui/button";
14: import { Save, RotateCcw, Plus, Trash2, Dna } from "lucide-react";
15: import type {
16:   FamilyGeneticImpact,

```



```

17:   FamilyGeneticImpactSection,
18: } from "@types/report-types";
19:
20: interface FamilyGeneticImpactAdminProps {
21:   familyGeneticImpactSection: FamilyGeneticImpactSection;
22:   setFamilyGeneticImpactSection: (section: FamilyGeneticImpactSection) => void;
23:   onSave: () => void;
24:   onReset: () => void;
25: }
26:
27: export default function FamilyGeneticImpactAdmin({
28:   familyGeneticImpactSection,
29:   setFamilyGeneticImpactSection,
30:   onSave,
31:   onReset,
32: }: FamilyGeneticImpactAdminProps) {
33:   const addFamilyGeneticImpact = () => {
34:     setFamilyGeneticImpactSection({
35:       ...familyGeneticImpactSection,
36:       impacts: [
37:         ...familyGeneticImpactSection.impacts,
38:         {
39:           gene: "",
40:           normalAlleles: "",
41:           yourResult: "",
42:           healthImpact: "",
43:         },
44:       ],
45:     });
46:   };
47:
48:   const updateFamilyGeneticImpact = (
49:     index: number,
50:     field: keyof FamilyGeneticImpact,
51:     value: string
52:   ) => {
53:     const updatedImpacts = [...familyGeneticImpactSection.impacts];
54:     updatedImpacts[index] = { ...updatedImpacts[index], [field]: value };
55:     setFamilyGeneticImpactSection({
56:       ...familyGeneticImpactSection,
57:       impacts: updatedImpacts,
58:     });
59:   };
60:
61:   const removeFamilyGeneticImpact = (index: number) => {
62:     setFamilyGeneticImpactSection({
63:       ...familyGeneticImpactSection,
64:       impacts: familyGeneticImpactSection.impacts.filter((_, i) => i !== index),
65:     });
66:   };
67:
68:   const setDescription = (value: string) => {
69:     setFamilyGeneticImpactSection({
70:       ...familyGeneticImpactSection,
71:       description: value,
72:     });
73:   };
74:
75:   return (
76:     <div className="w-full max-w-8xl mx-auto ">
77:       <Card className="shadow-xl border-0 bg-gradient-to-br from-slate-50 to-blue-50/20 overflow-hidden">
78:         <CardHeader className="bg-gradient-to-r from-indigo-600 via-purple-600 to-blue-500 text-white rounded">
79:           <CardTitle className="text-lg sm:text-xl md:text-2xl font-bold uppercase flex items-center justify">
80:             Family Genetic Impact Management
81:           </CardTitle>
82:         </CardHeader>
83:
84:         <div className="flex flex-col sm:flex-row justify-end items-start sm:items-center mt-3 gap-4 mb-4 p">
85:           <Button onClick={addFamilyGeneticImpact}>
86:             <Plus className="h-4 w-4 mr-2" />
87:             Add Genetic Impact
88:           </Button>
89:           <div className="flex gap-3 w-full sm:w-auto">

```

```

90:         <Button onClick={onSave}>
91:             <Save className="h-4 w-4 mr-2" />
92:             Save Changes
93:         </Button>
94:     </div>
95: </div>
96:
97: <CardContent className="mt-6">
98:     <div className="space-y-2">
99:         <Label className="text-sm font-semibold text-gray-700">
100:             Introductory Description
101:         </Label>
102:         <Textarea
103:             rows={9}
104:             value={familyGeneticImpactSection.description}
105:             onChange={(e) => setDescription(e.target.value)}
106:             placeholder="Describe the overall family genetic impact, key findings, and any recommendation."
107:             className="border-2 border-gray-200 focus:border-blue-500 focus:ring-2 focus:ring-blue-200 text-gray-600"
108:         />
109:     </div>
110:     { /* Action Buttons */ }
111:
112:     { /* Family Genetic Impact List */ }
113:     <div className="space-y-6">
114:         {familyGeneticImpactSection.impacts.length === 0 && (
115:             <div className="text-center py-16 px-4">
116:                 <div className="mx-auto w-24 h-24 bg-gradient-to-br from-blue-100 to-indigo-100 rounded-full" />
117:                 <Dna className="h-12 w-12 text-blue-600" />
118:             </div>
119:             <h3 className="text-xl font-semibold text-gray-800 mb-3">
120:                 No Genetic Impacts Added
121:             </h3>
122:             <p className="text-gray-600 mb-8 max-w-md mx-auto">
123:                 Start building your family genetic profile by adding the first
124:                 genetic impact analysis.
125:             </p>
126:             <Button
127:                 onClick={addFamilyGeneticImpact}
128:                 className="bg-gradient-to-r from-blue-600 to-indigo-600 hover:from-blue-700 hover:to-indigo-700 text-white font-medium"
129:             >
130:                 <Plus className="h-4 w-4 mr-2" />
131:                 Add First Impact
132:             </Button>
133:         </div>
134:     )}
135:
136:     {familyGeneticImpactSection.impacts?.map((impact, index) => (
137:         <Card
138:             key={index}
139:             className="border-0 bg-white shadow-lg hover:shadow-xl transition-all duration-300 overflow-hidden"
140:         >
141:             <CardHeader className="bg-gradient-to-r from-slate-50 to-blue-50 border-b border-gray-100">
142:                 <div className="flex justify-between items-start">
143:                     <div>
144:                         <CardTitle className="text-lg sm:text-xl font-bold text-gray-800 flex items-center gap-2">
145:                             <div className="w-8 h-8 bg-gradient-to-br from-blue-500 to-indigo-500 rounded-full flex items-center justify-center text-white">
146:                                 {index + 1}
147:                             </div>
148:                             Genetic Impact Analysis
149:                         </CardTitle>
150:                         <p className="text-sm text-gray-600 mt-1">
151:                             Configure genetic variation #{index + 1}
152:                         </p>
153:                     </div>
154:                     <Button
155:                         onClick={() => removeFamilyGeneticImpact(index)}
156:                         size="sm"
157:                         variant="destructive"
158:                         className="bg-red-500 hover:bg-red-600 text-white shadow-md hover:shadow-lg transition-colors"
159:                     >
160:                         <Trash2 className="h-4 w-4" />
161:                     </Button>
162:                 </div>

```

```

163:         </CardHeader>
164:
165:         <CardContent className="p-6 space-y-6">
166:             <div className="grid grid-cols-1 lg:grid-cols-3 gap-6">
167:                 <div className="space-y-2">
168:                     <Label className="text-sm font-semibold text-gray-700 flex items-center gap-2">
169:                         Gene Identifier
170:                         <span className="text-xs text-gray-500 bg-gray-100 px-2 py-1 rounded">
171:                             Required
172:                         </span>
173:                     </Label>
174:                     <Input
175:                         value={impact.gene}
176:                         onChange={(e) =>
177:                             updateFamilyGeneticImpact(
178:                                 index,
179:                                 "gene",
180:                                 e.target.value
181:                             )
182:                     }
183:                     placeholder="e.g., MnSOD, COMT, MTHFR"
184:                     className="border-2 border-gray-200 focus:border-blue-500 focus:ring-2 focus:ring-b
185:                 />
186:             </div>
187:
188:             <div className="space-y-2">
189:                 <Label className="text-sm font-semibold text-gray-700 flex items-center gap-2">
190:                     Normal Alleles
191:                     <span className="text-xs text-gray-500 bg-gray-100 px-2 py-1 rounded">
192:                         Reference
193:                     </span>
194:                 </Label>
195:                 <Input
196:                     value={impact.normalAlleles}
197:                     onChange={(e) =>
198:                         updateFamilyGeneticImpact(
199:                             index,
200:                             "normalAlleles",
201:                             e.target.value
202:                         )
203:                 }
204:                 placeholder="e.g., AA, GG, CC"
205:                 className="border-2 border-gray-200 focus:border-blue-500 focus:ring-2 focus:ring-b
206:             />
207:             </div>
208:
209:             <div className="space-y-2">
210:                 <Label className="text-sm font-semibold text-gray-700 flex items-center gap-2">
211:                     Your Result
212:                     <span className="text-xs text-white bg-blue-500 px-2 py-1 rounded">
213:                         Current
214:                     </span>
215:                 </Label>
216:                 <Input
217:                     value={impact.yourResult}
218:                     onChange={(e) =>
219:                         updateFamilyGeneticImpact(
220:                             index,
221:                             "yourResult",
222:                             e.target.value
223:                         )
224:                 }
225:                 placeholder="e.g., AG, CT, TT"
226:                 className="border-2 border-blue-200 focus:border-blue-500 focus:ring-2 focus:ring-b
227:             />
228:             </div>
229:         </div>
230:
231:         <div className="space-y-2">
232:             <Label className="text-sm font-semibold text-gray-700 flex items-center gap-2">
233:                 Health Impact Analysis
234:                 <span className="text-xs text-gray-500 bg-gray-100 px-2 py-1 rounded">
235:                     Detailed

```

```

236:         </span>
237:     </Label>
238:     <Textarea
239:       value={impact.healthImpact}
240:       onChange={(e) =>
241:         updateFamilyGeneticImpact(
242:           index,
243:           "healthImpact",
244:           e.target.value
245:         )
246:       }
247:       rows={4}
248:       placeholder="Describe the potential health implications, recommendations, and any rel
249:       className="border-2 border-gray-200 focus:border-blue-500 focus:ring-2 focus:ring-blue-500"
250:     />
251:     <p className="text-xs text-gray-500 mt-1">
252:       Provide detailed analysis of how this genetic variation
253:       may impact health outcomes
254:     </p>
255:   </div>
256: </CardContent>
257: </Card>
258:   )})
259: </div>
260: </CardContent>
261: </Card>
262: </div>
263:   );
264: }

```

## ■ File: components\admin\gene-results-admin.tsx

```

1: "use client";
2:
3: import { Button } from "@/components/ui/button";
4: import {
5:   Card,
6:   CardContent,
7:   CardDescription,
8:   CardHeader,
9:   CardTitle,
10: } from "@/components/ui/card";
11: import { Input } from "@/components/ui/input";
12: import { Label } from "@/components/ui/label";
13: import { ScrollArea } from "@/components/ui/scroll-area";
14: import {
15:   AlertDialog,
16:   AlertDialogAction,
17:   AlertDialogCancel,
18:   AlertDialogContent,
19:   AlertDialogDescription,
20:   AlertDialogFooter,
21:   AlertDialogHeader,
22:   AlertDialogTitle,
23:   AlertDialogTrigger,
24: } from "@/components/ui/alert-dialog";
25: import { PlusCircle, Trash2, Save, RotateCcw } from "lucide-react";
26: import type { GeneTestResult } from "@/types/report-types";
27:
28: interface GeneResultsAdminProps {
29:   geneTestResults: GeneTestResult[];
30:   addGeneTestResult: () => void;
31:   updateGeneTestResult: (
32:     index: number,
33:     field: keyof GeneTestResult,
34:     value: string
35:   ) => void;
36:   removeGeneTestResult: (index: number) => void;
37:   onSave: () => void;
38:   onReset: () => void;
39: }

```

```

40:
41: export default function GeneResultsAdmin({
42:   geneTestResults,
43:   addGeneTestResult,
44:   updateGeneTestResult,
45:   removeGeneTestResult,
46:   onSave,
47:   onReset,
48: }): GeneResultsAdminProps) {
49:   return (
50:     <Card className="shadow-lg border-0 bg-white w-full max-w-full">
51:       <CardHeader className="bg-gradient-to-r from-indigo-500 to-blue-600 text-white rounded-lg p-4 sm:p-5">
52:         <CardTitle className="text-lg sm:text-xl md:text-2xl font-bold uppercase flex items-center justify-
53:           Gene Test Results
54:         </CardTitle>
55:       </CardHeader>
56:       <CardContent className="space-y-4 sm:space-y-6 p-3 sm:p-4 lg:p-6 xl:p-8">
57:         <div className="flex flex-col sm:flex-row sm:justify-between sm:items-center gap-3 sm:gap-4">
58:           <h3 className="text-base sm:text-lg font-semibold">
59:             Gene Test Results ({geneTestResults.length})
60:           </h3>
61:           <div className="flex flex-wrap gap-2 justify-end">
62:             <Button
63:               onClick={addGeneTestResult}
64:               className="bg-indigo-600 hover:bg-indigo-700 text-xs sm:text-sm"
65:             >
66:               Add Gene Result
67:             </Button>
68:             <Button onClick={onSave}>Save Change</Button>
69:           </div>
70:         </div>
71:
72:         <div className="space-y-3">
73:           {geneTestResults.map((result, index) => (
74:             <div
75:               key={index}
76:               className="border-2 border-gray-200 rounded-lg p-3 sm:p-4 lg:p-6 bg-gray-50"
77:             >
78:               { /* Mobile Layout - Stacked */ }
79:               <div className="block sm:hidden space-y-3">
80:                 <div className="flex items-center justify-between">
81:                   <span className="inline-flex items-center justify-center bg-green-100 text-indigo-800 rou
82:                     {index + 1}
83:                   </span>
84:                   <AlertDialog>
85:                     <AlertDialogTrigger asChild>
86:                       <Button variant="destructive" size="sm">
87:                         <Trash2 className="h-4 w-4" />
88:                       </Button>
89:                     </AlertDialogTrigger>
90:                     <AlertDialogContent>
91:                       <AlertDialogHeader>
92:                         <AlertDialogTitle>Are you sure?</AlertDialogTitle>
93:                         <AlertDialogDescription>
94:                           This will permanently delete the gene result for{" "}
95:                           <strong>
96:                             {result.genecode || `Gene #${index + 1}`}
97:                           </strong>
98:                           . This action cannot be undone.
99:                         </AlertDialogDescription>
100:                       </AlertDialogHeader>
101:                       <AlertDialogFooter>
102:                         <AlertDialogCancel>Cancel</AlertDialogCancel>
103:                         <AlertDialogAction
104:                           onClick={() => removeGeneTestResult(index)}
105:                           className="bg-red-600 hover:bg-red-700"
106:                         >
107:                           Delete
108:                         </AlertDialogAction>
109:                       </AlertDialogFooter>
110:                     </AlertDialogContent>
111:                   </AlertDialog>
112:                 </div>

```

```

113:
114:     <div className="space-y-3">
115:         <div>
116:             <Label className="text-xs font-medium text-gray-600 mb-1 block">
117:                 Gene Code
118:             </Label>
119:             <Input
120:                 value={result.genecode}
121:                 onChange={(e) =>
122:                     updateGeneTestResult(index, "genecode", e.target.value)
123:                 }
124:                 placeholder="Gene Code"
125:                 className="border-2 focus:border-indigo-500"
126:             />
127:         </div>
128:
129:         <div>
130:             <Label className="text-xs font-medium text-gray-600 mb-1 block">
131:                 Gene Name
132:             </Label>
133:             <Input
134:                 value={result.geneName}
135:                 onChange={(e) =>
136:                     updateGeneTestResult(index, "geneName", e.target.value)
137:                 }
138:                 placeholder="Gene Name"
139:                 className="border-2 focus:border-indigo-500"
140:             />
141:         </div>
142:
143:         <div>
144:             <Label className="text-xs font-medium text-gray-600 mb-1 block">
145:                 Variation
146:             </Label>
147:             <Input
148:                 value={result.variation}
149:                 onChange={(e) =>
150:                     updateGeneTestResult(index, "variation", e.target.value)
151:                 }
152:                 placeholder="Variation"
153:                 className="border-2 focus:border-indigo-500 font-mono"
154:             />
155:         </div>
156:
157:         <div>
158:             <Label className="text-xs font-medium text-gray-600 mb-1 block">
159:                 Result
160:             </Label>
161:             <Input
162:                 value={result.result}
163:                 onChange={(e) =>
164:                     updateGeneTestResult(index, "result", e.target.value)
165:                 }
166:                 placeholder="Result"
167:                 className="border-2 focus:border-indigo-500 font-mono font-bold"
168:             />
169:         </div>
170:     </div>
171: </div>
172:
173: {/* Tablet and Desktop Layout - Inline */}
174: <div className="hidden sm:flex items-center gap-2 md:gap-3 lg:gap-4">
175:     {/* Gene Number - Left corner */}
176:     <div className="flex-shrink-0">
177:         <span className="inline-flex items-center justify-center bg-green-100 text-indigo-800 rounded-md px-2 py-1">
178:             {index + 1}
179:         </span>
180:     </div>
181:
182:     {/* Four Input Fields - Responsive grid */}
183:     <div className="flex-1 grid grid-cols-2 md:grid-cols-4 gap-2 md:gap-3">
184:         {/* Gene Code */}
185:         <div className="space-y-1">

```

```

186:         <Label className="text-xs font-medium text-gray-600 md:hidden">
187:             Gene Code
188:         </Label>
189:         <Input
190:             value={result.genecode}
191:             onChange={(e) =>
192:                 updateGeneTestResult(index, "genecode", e.target.value)
193:             }
194:             placeholder="Gene Code"
195:             className="border-2 focus:border-indigo-500"
196:         />
197:     </div>
198:
199:     { /* Gene Name */ }
200:     <div className="space-y-1">
201:         <Label className="text-xs font-medium text-gray-600 md:hidden">
202:             Gene Name
203:         </Label>
204:         <Input
205:             value={result.geneName}
206:             onChange={(e) =>
207:                 updateGeneTestResult(index, "geneName", e.target.value)
208:             }
209:             placeholder="Gene Name"
210:             className="border-2 focus:border-indigo-500"
211:         />
212:     </div>
213:
214:     { /* Variation */ }
215:     <div className="space-y-1">
216:         <Label className="text-xs font-medium text-gray-600 md:hidden">
217:             Variation
218:         </Label>
219:         <Input
220:             value={result.variation}
221:             onChange={(e) =>
222:                 updateGeneTestResult(index, "variation", e.target.value)
223:             }
224:             placeholder="Variation"
225:             className="border-2 focus:border-indigo-500 font-mono"
226:         />
227:     </div>
228:
229:     { /* Result */ }
230:     <div className="space-y-1">
231:         <Label className="text-xs font-medium text-gray-600 md:hidden">
232:             Result
233:         </Label>
234:         <Input
235:             value={result.result}
236:             onChange={(e) =>
237:                 updateGeneTestResult(index, "result", e.target.value)
238:             }
239:             placeholder="Result"
240:             className="border-2 focus:border-indigo-500 font-mono font-bold"
241:         />
242:     </div>
243: </div>
244:
245: { /* Delete Button - Right corner */ }
246: <div className="flex-shrink-0">
247:     <AlertDialog>
248:         <AlertDialogTrigger asChild>
249:             <Button variant="destructive" size="sm">
250:                 <Trash2 className="h-4 w-4" />
251:             </Button>
252:         </AlertDialogTrigger>
253:         <AlertDialogContent>
254:             <AlertDialogHeader>
255:                 <AlertDialogTitle>Are you sure?</AlertDialogTitle>
256:                 <AlertDialogDescription>
257:                     This will permanently delete the gene result for{" "}
258:                     <strong>

```

```

259:             {result.genecode || `Gene #${index + 1}`}
260:         </strong>
261:         . This action cannot be undone.
262:     </AlertDialogDescription>
263: </AlertDialogHeader>
264: <AlertDialogFooter>
265:     <AlertDialogCancel>Cancel</AlertDialogCancel>
266:     <AlertDialogAction
267:         onClick={() => removeGeneTestResult(index)}
268:         className="bg-red-600 hover:bg-red-700"
269:     >
270:         Delete
271:     </AlertDialogAction>
272: </AlertDialogFooter>
273: </AlertDialogContent>
274: </AlertDialog>
275: </div>
276: </div>
277: </div>
278: )}}
279: </div>
280:
281: {geneTestResults.length === 0 && (
282:     <div className="text-center py-8 sm:py-12 text-gray-500">
283:         <div className="text-4xl sm:text-6xl mb-3 sm:mb-4">?</div>
284:         <p className="text-base sm:text-lg mb-3 sm:mb-4">
285:             No gene test results added yet
286:         </p>
287:         <Button
288:             onClick={addGeneTestResult}
289:             className="bg-indigo-600 hover:bg-indigo-700"
290:         >
291:             <PlusCircle className="h-4 w-4 mr-2" />
292:             Add First Gene Result
293:         </Button>
294:     </div>
295: )}
296: </CardContent>
297: </Card>
298: );
299: }

```

## ■ File: components\admin\genes-addiction-admin.tsx

```

1: "use client";
2:
3: import { Card, CardContent, CardHeader, CardTitle } from "@/components/ui/card";
4: import { Input } from "@/components/ui/input";
5: import { Textarea } from "@/components/ui/textarea";
6: import { Label } from "@/components/ui/label";
7: import { Button } from "@/components/ui/button";
8: import { Upload, Save, X, Trash2, Plus } from "lucide-react";
9: import Image from "next/image";
10: import { useEffect, useState } from "react";
11: import { GenesAndAddiction, AddictionEntry } from "@/types/report-types";
12: import {
13:     Select,
14:     SelectContent,
15:     SelectItem,
16:     SelectTrigger,
17:     SelectValue,
18: } from "@/components/ui/select";
19: import {
20:     Dialog,
21:     DialogTrigger,
22:     DialogContent,
23:     DialogHeader,
24:     DialogTitle,
25:     DialogDescription,
26:     DialogClose,
27: } from "@/components/ui/dialog";

```



```

28:
29: interface Props {
30:   data: GenesAndAddiction;
31:   updateData: (data: Partial<GenesAndAddiction>) => void;
32:   onSave: () => void;
33:   onReset: () => void;
34: }
35:
36: interface AddictionImage {
37:   label: string;
38:   url: string;
39: }
40:
41: export default function GenesAddictionAdmin({
42:   data,
43:   updateData,
44:   onSave,
45:   onReset,
46: }: Props) {
47:   const [quote, setQuote] = useState(data.quote || "");
48:   const [description, setDescription] = useState(data.description || "");
49:   const [fields, setFields] = useState<Record<string, AddictionEntry>>({
50:     data.data || {}
51:   });
52:   const [selectedFieldKey, setSelectedFieldKey] = useState<string | null>(null);
53:   const [labelIconLabel, setLabelIconLabel] = useState("");
54:   const [labelIconFile, setLabelIconFile] = useState<File | null>(null);
55:   const [sensitivityIconLabel, setSensitivityIconLabel] = useState("");
56:   const [sensitivityIconFile, setSensitivityIconFile] = useState<File | null>(
57:     null
58:   );
59:   const [labelIcons, setLabelIcons] = useState<Record<string, string>>({});
60:   const [sensitivityIcons, setSensitivityIcons] = useState<
61:     Record<string, string>
62:   >({});
63:   const [showIconsModal, setShowIconsModal] = useState(false);
64:   const [showAddEntryModal, setShowAddEntryModal] = useState(false);
65:   const [newEntryTitle, setNewEntryTitle] = useState("");
66:
67:   useEffect(() => {
68:     const fetchImages = async () => {
69:       try {
70:         const res = await fetch("/api/addiction-images");
71:         const json = await res.json();
72:
73:         if (json.success && json.icons && json.sensitivities) {
74:           const labelMap: Record<string, string> = {};
75:           const sensitivityMap: Record<string, string> = {};
76:
77:           for (const img of json.icons) {
78:             if (img.label && img.url) {
79:               labelMap[img.label.toLowerCase()] = img.url;
80:             }
81:           }
82:
83:           for (const img of json.sensitivities) {
84:             if (img.label && img.url) {
85:               sensitivityMap[img.label.toLowerCase()] = img.url;
86:             }
87:           }
88:
89:           setLabelIcons(labelMap);
90:           setSensitivityIcons(sensitivityMap);
91:         } else {
92:           console.warn("No images found in API response:", json);
93:         }
94:       } catch (err) {
95:         console.error("Failed to fetch addiction images:", err);
96:       }
97:     };
98:
99:     fetchImages();
100:   }, []);

```

```

101:
102: const handleUploadIcon = async (type: "icon" | "sensitivityIcon") => {
103:   const label = type === "icon" ? labelIconLabel : sensitivityIconLabel;
104:   const file = type === "icon" ? labelIconFile : sensitivityIconFile;
105:
106:   if (!label.trim() || !file) return;
107:
108:   const formData = new FormData();
109:   formData.append("file", file);
110:   formData.append("label", label.trim().toLowerCase());
111:   formData.append(
112:     "folder",
113:     type === "icon" ? "addiction/icons" : "addiction/sensitivity"
114:   );
115:
116:   const res = await fetch("/api/addiction-images", {
117:     method: "POST",
118:     body: formData,
119:   });
120:
121:   const json = await res.json();
122:   if (json.success) {
123:     if (type === "icon") {
124:       setLabelIcons((prev) => ({ ...prev, [json.label]: json.url }));
125:       setLabelIconFile(null);
126:       setLabelIconLabel("");
127:     } else {
128:       setSensitivityIcons((prev) => ({ ...prev, [json.label]: json.url }));
129:       setSensitivityIconFile(null);
130:       setSensitivityIconLabel("");
131:     }
132:
133:     if (type === "icon") {
134:       setLabelIconFile(null);
135:       setLabelIconLabel("");
136:     } else {
137:       setSensitivityIconFile(null);
138:       setSensitivityIconLabel("");
139:     }
140:   }
141: };
142:
143: const handleFieldChange = (
144:   key: string,
145:   field: keyof AddictionEntry,
146:   value: string
147: ) => {
148:   const updated = {
149:     ...fields,
150:     [key]: {
151:       ...fields[key],
152:       [field]: value,
153:     },
154:   };
155:   setFields(updated);
156:   updateData({ data: updated });
157: };
158:
159: const handleDeleteField = (key: string) => {
160:   const updated = { ...fields };
161:   delete updated[key];
162:   setFields(updated);
163:   updateData({ data: updated });
164: };
165:
166: const handleAddField = () => {
167:   setShowAddEntryModal(true);
168: };
169:
170: const confirmAddField = () => {
171:   if (!newEntryTitle.trim()) return;
172:
173:   const newKey = `entry_${Date.now()}`;

```

```

174:     const updated = {
175:       ...fields,
176:       [newKey]: {
177:         title: newEntryTitle.trim(),
178:         icon: "",
179:         sensitivity: "Medium",
180:       },
181:     };
182:     setFields(updated);
183:     updateData({ data: updated });
184:     setNewEntryTitle("");
185:     setShowAddEntryModal(false);
186:   };
187:
188:   useEffect(() => {
189:     updateData({ quote, description });
190:   }, [quote, description]);
191:
192:   return (
193:     <div className="min-h-screen bg-gray-50">
194:       <div className="max-w-8xl mx-auto space-y-6">
195:         {/* Header Section */}
196:         <Card className="shadow-lg border-0">
197:           <CardHeader className="bg-gradient-to-r from-red-600 via-red-500 to-orange-500 text-white rounded">
198:             <CardTitle className="text-lg sm:text-xl md:text-2xl font-bold uppercase flex items-center justify-center">
199:               Genes & Addiction Management
200:             </CardTitle>
201:           </CardHeader>
202:         </Card>
203:
204:         <div className="text-end mt-4">
205:           <Button onClick={onSave}>
206:             <Save className="w-4 h-4 mr-2" />
207:             Save Changes
208:           </Button>
209:         </div>
210:
211:         {/* Main Content Grid */}
212:         <div className="grid grid-cols-1 xl:grid-cols-6 gap-6">
213:           {/* Left Column - Content Settings */}
214:           <div className="xl:col-span-3 space-y-6">
215:             {/* Content Settings */}
216:             <Card className="shadow-md border-0 h-fit">
217:               <CardHeader className="bg-white border-b border-gray-100">
218:                 <CardTitle className="text-lg text-gray-800 flex items-center gap-2">
219:                   Content Settings
220:                 </CardTitle>
221:               </CardHeader>
222:               <CardContent className="p-6 space-y-6">
223:                 <div className="space-y-3">
224:                   <Label className="text-sm font-semibold text-gray-700">
225:                     Inspirational Quote
226:                   </Label>
227:                   <Textarea
228:                     value={quote}
229:                     onChange={e => setQuote(e.target.value)}
230:                     placeholder="Enter an inspiring quote about overcoming addiction..."
231:                     className="min-h-[100px] resize-none border-gray-200 focus:border-red-300 focus:ring-red-300"
232:                   />
233:                 </div>
234:                 <div className="space-y-3">
235:                   <Label className="text-sm font-semibold text-gray-700">
236:                     Section Description
237:                   </Label>
238:                   <Textarea
239:                     value={description}
240:                     onChange={e => setDescription(e.target.value)}
241:                     placeholder="Enter a detailed description of the genes and addiction section..."
242:                     className="min-h-[120px] resize-none border-gray-200 focus:border-red-300 focus:ring-red-300"
243:                   />
244:                 </div>
245:               </CardContent>
246:             </Card>

```

```

247:
248:     { /* Add New Entry Section */ }
249:     <Card className="shadow-md border-0">
250:         <CardContent className="p-6">
251:             <div className="text-center">
252:                 <Button
253:                     variant="outline"
254:                     onClick={handleAddField}
255:                     className="border-2 border-dashed border-red-300 text-red-600 hover:bg-red-50 hover:bor
256:                 >
257:                     <Plus className="w-5 h-5 mr-2" />
258:                     Add New Addiction Entry
259:                 </Button>
260:                 <p className="text-sm text-gray-500 mt-2">
261:                     Create a new addiction entry with custom icons and details
262:                 </p>
263:             </div>
264:         </CardContent>
265:     </Card>
266: </div>
267:
268: { /* Right Column - Image Management */ }
269: <div className="xl:col-span-3">
270:     <Card className="shadow-md border-0 sticky top-6 h-fit">
271:         <CardHeader className="bg-white border-b border-gray-100">
272:             <CardTitle className="text-lg text-gray-800 flex items-center gap-2">
273:                 <Upload className="w-5 h-5 text-red-600" />
274:                 Image Management
275:             </CardTitle>
276:         </CardHeader>
277:         <CardContent className="p-6 space-y-6">
278:             { /* Label Icon Upload */ }
279:             <div className="space-y-4">
280:                 <div className="flex items-center gap-2 pb-2 border-b border-gray-100">
281:                     <div className="w-2 h-2 bg-red-500 rounded-full"></div>
282:                     <Label className="text-sm font-semibold text-gray-700">
283:                         Label Icons
284:                     </Label>
285:                 </div>
286:                 <div className="space-y-3">
287:                     <Input
288:                         placeholder="Label (e.g., alcohol)"
289:                         value={labelIconLabel}
290:                         onChange={(e) => setLabelIconLabel(e.target.value)}
291:                         className="text-sm border-gray-200 focus:border-red-300 focus:ring-red-200"
292:                     />
293:                     <Input
294:                         type="file"
295:                         accept="image/*"
296:                         onChange={(e) =>
297:                             setLabelIconFile(e.target.files?.[0] || null)
298:                         }
299:                         className="text-sm border-gray-200 focus:border-red-300 focus:ring-red-200"
300:                     />
301:                     <Button
302:                         onClick={() => handleUploadIcon("icon")}
303:                         className="bg-red-600 hover:bg-red-700 text-white w-full shadow-sm"
304:                         disabled={!labelIconLabel.trim() || !labelIconFile}
305:                         size="sm"
306:                     >
307:                         <Upload className="w-4 h-4 mr-2" />
308:                         Upload Icon
309:                     </Button>
310:                 </div>
311:             </div>
312:
313:             { /* Sensitivity Icon Upload */ }
314:             <div className="space-y-4">
315:                 <div className="flex items-center gap-2 pb-2 border-b border-gray-100">
316:                     <div className="w-2 h-2 bg-orange-500 rounded-full"></div>
317:                     <Label className="text-sm font-semibold text-gray-700">
318:                         Sensitivity Icons
319:                     </Label>

```

```

320:         </div>
321:     <div className="space-y-3">
322:         <Input
323:             placeholder="Label (e.g., high_sensitivity)"
324:             value={sensitivityIconLabel}
325:             onChange={(e) => setSensitivityIconLabel(e.target.value)}
326:             className="text-sm border-gray-200 focus:border-red-300 focus:ring-red-200"
327:         />
328:         <Input
329:             type="file"
330:             accept="image/*"
331:             onChange={(e) =>
332:                 setSensitivityIconFile(e.target.files?.[0] || null)
333:             }
334:             className="text-sm border-gray-200 focus:border-red-300 focus:ring-red-200"
335:         />
336:         <Button
337:             onClick={() => handleUploadIcon("sensitivityIcon")}
338:             className="bg-red-600 hover:bg-red-700 text-white w-full shadow-sm"
339:             disabled={
340:                 !sensitivityIconLabel.trim() || !sensitivityIconFile
341:             }
342:             size="sm"
343:         >
344:             <Upload className="w-4 h-4 mr-2" />
345:             Upload Icon
346:         </Button>
347:     </div>
348: </div>
349:
350: { /* Available Icons Preview */ }
351: <div className="space-y-4">
352:     <div className="flex items-center justify-between pb-2 border-b border-gray-100">
353:         <div className="flex items-center gap-2">
354:             <div className="w-2 h-2 bg-blue-500 rounded-full"></div>
355:             <Label className="text-sm font-semibold text-gray-700">
356:                 Available Icons
357:             </Label>
358:         </div>
359:         <Button
360:             variant="outline"
361:             size="sm"
362:             onClick={() => setShowIconsModal(true)}
363:             className="text-xs px-3 py-1 h-auto border-blue-300 text-blue-600 hover:bg-blue-50"
364:         >
365:             View All (
366:                 {Object.keys(labelIcons).length +
367:                  Object.keys(sensitivityIcons).length}
368:             )
369:         </Button>
370:     </div>
371: </div>
372: </CardContent>
373: </Card>
374: </div>
375: </div>
376:
377: { /* Addition Entries Section */ }
378: <div className="space-y-6">
379:     {Object.entries(fields).length === 0 ? (
380:         <Card className="shadow-md border-0">
381:             <CardContent className="flex flex-col items-center justify-center py-16">
382:                 <div className="text-gray-300 mb-6">
383:                     <div className="w-20 h-20 border-4 border-dashed border-gray-200 rounded-full flex items-center justify-center">
384:                         <Plus className="w-8 h-8" />
385:                     </div>
386:                 </div>
387:                 <h3 className="text-xl font-semibold text-gray-400 mb-2">
388:                     No addition entries yet
389:                 </h3>
390:                 <p className="text-sm text-gray-400 text-center max-w-md">
391:                     Start building your addiction management content by adding
392:                     your first entry above

```

```

393:         </p>
394:     </CardContent>
395: </Card>
396: ) : (
397:     <div className="space-y-6">
398:         <div className="flex items-center justify-between">
399:             <h3 className="text-xl font-bold text-gray-800 flex items-center gap-2">
400:                 Addition Entries
401:             </h3>
402:             <div className="bg-red-100 text-red-700 px-3 py-1 rounded-full text-sm font-medium">
403:                 {Object.entries(fields).length} entries
404:             </div>
405:         </div>
406:
407:         <div className="grid grid-cols-1 lg:grid-cols-2 gap-6">
408:             {Object.entries(fields).map(([key, value], index) => (
409:                 <Card
410:                     key={key}
411:                     className="shadow-md border-0 border-l-4 border-l-red-500 hover:shadow-lg transition-al
412:                 >
413:                     <CardHeader className="bg-gradient-to-r from-gray-50 to-white border-b border-gray-100">
414:                         <div className="flex justify-between items-start">
415:                             <div className="flex items-center gap-3">
416:                                 <div className="w-8 h-8 bg-red-600 text-white rounded-full flex items-center just
417:                                     {index + 1}
418:                                 </div>
419:                             <div>
420:                                 <CardTitle className="text-lg text-gray-800">
421:                                     {value.title || "Untitled Entry"}
422:                                 </CardTitle>
423:                             </div>
424:                         </div>
425:                         <Button
426:                             size="sm"
427:                             variant="ghost"
428:                             onClick={() => handleDeleteField(key)}
429:                             className="text-red-600 hover:bg-red-50 hover:text-red-700"
430:                         >
431:                             <Trash2 className="w-4 h-4" />
432:                         </Button>
433:                     </div>
434:                 </CardHeader>
435:                 <CardContent className="p-6">
436:                     <div className="space-y-4">
437:                         { /* Title Field */ }
438:                         <div className="space-y-2">
439:                             <Label className="text-sm font-semibold text-gray-700">
440:                                 Entry Title
441:                             </Label>
442:                             <Input
443:                                 value={value.title}
444:                                 onChange={(e) =>
445:                                     handleFieldChange(key, "title", e.target.value)
446:                                 }
447:                                 placeholder="Enter addition title (e.g., Alcohol Sensitivity)"
448:                                 className="text-sm border-gray-200 focus:border-red-300 focus:ring-red-200"
449:                             />
450:                         </div>
451:
452:                         { /* Icon Selectors */ }
453:                         <div className="grid grid-cols-1 gap-4">
454:                             <div className="space-y-2">
455:                                 <Label className="text-sm font-semibold text-gray-700">
456:                                     Label Icon
457:                                 </Label>
458:                                 <Select
459:                                     value={value.icon}
460:                                     onChange={(val) =>
461:                                         handleFieldChange(key, "icon", val)
462:                                     }
463:                                 >
464:                                     <SelectTrigger className="text-sm border-gray-200 focus:border-red-300 focus:
465:                                         <SelectValue placeholder="Choose an icon" />

```

```

466:         </SelectTrigger>
467:         <SelectContent>
468:             {Object.entries(labelIcons).map(
469:                 ([label, url]) => (
470:                     <SelectItem key={label} value={label}>
471:                         <div className="flex items-center gap-3">
472:                             <Image
473:                                 src={url}
474:                                 alt={label}
475:                                 width={24}
476:                                 height={24}
477:                                 className="rounded"
478:                             />
479:                             <span className="capitalize font-medium">
480:                                 {label.replace(/_/g, " ")}
481:                             </span>
482:                         </div>
483:                     </SelectItem>
484:                 )
485:             )}
486:         </SelectContent>
487:     </Select>
488: </div>
489:
490: <div className="space-y-2">
491:     <Label className="text-sm font-semibold text-gray-700">
492:         Sensitivity Icon
493:     </Label>
494:     <Select
495:         value={value.sensitivityIcon}
496:         onValueChange={(val) =>
497:             handleFieldChange(key, "sensitivityIcon", val)
498:         }
499:     >
500:         <SelectTrigger className="text-sm border-gray-200 focus:border-red-300 focus:outline-none">
501:             <SelectValue placeholder="Choose sensitivity level" />
502:         </SelectTrigger>
503:         <SelectContent>
504:             {Object.entries(sensitivityIcons).map(
505:                 ([label, url]) => (
506:                     <SelectItem key={label} value={label}>
507:                         <div className="flex items-center gap-3">
508:                             <Image
509:                                 src={url}
510:                                 alt={label}
511:                                 width={24}
512:                                 height={24}
513:                                 className="rounded"
514:                             />
515:                             <span className="capitalize font-medium">
516:                                 {label.replace(/_/g, " ")}
517:                             </span>
518:                         </div>
519:                     </SelectItem>
520:                 )
521:             )}
522:         </SelectContent>
523:     </Select>
524: </div>
525: </div>
526: </div>
527: </CardContent>
528: </Card>
529:     )})
530: </div>
531: </div>
532: )}
533: </div>
534:
535: { /* Add Entry Modal */ }
536: {showAddEntryModal && (
537:     <div className="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center z-50 p-4">
538:         <div className="bg-white rounded-lg max-w-md w-full">

```

```

539:         <div className="flex items-center justify-between p-6 border-b border-gray-200">
540:             <h2 className="text-xl font-bold text-gray-800">
541:                 Add New Addiction Entry
542:             </h2>
543:             <Button
544:                 variant="ghost"
545:                 size="sm"
546:                 onClick={() => {
547:                     setShowAddEntryModal(false);
548:                     setNewEntryTitle("");
549:                 }}
550:                 className="text-gray-500 hover:text-gray-700"
551:             >
552:                 <X className="w-5 h-5" />
553:             </Button>
554:         </div>
555:
556:         <div className="p-6">
557:             <div className="space-y-4">
558:                 <div className="space-y-2">
559:                     <Label className="text-sm font-semibold text-gray-700">
560:                         Entry Title *
561:                     </Label>
562:                     <Input
563:                         value={newEntryTitle}
564:                         onChange={(e) => setNewEntryTitle(e.target.value)}
565:                         placeholder="Enter addiction title (e.g., Alcohol Sensitivity)"
566:                         className="border-gray-200 focus:border-red-300 focus:ring-red-200"
567:                         onKeyDown={(e) => {
568:                             if (e.key === "Enter" && newEntryTitle.trim()) {
569:                                 confirmAddField();
570:                             }
571:                         }}
572:                     />
573:                 </div>
574:                 <p className="text-xs text-gray-500">
575:                     You can configure icons and other settings after creating
576:                     the entry.
577:                 </p>
578:             </div>
579:         </div>
580:
581:         <div className="flex justify-end gap-3 p-6 border-t border-gray-200">
582:             <Button
583:                 variant="outline"
584:                 onClick={() => {
585:                     setShowAddEntryModal(false);
586:                     setNewEntryTitle("");
587:                 }}
588:                 className="text-gray-600 hover:text-gray-700"
589:             >
590:                 Cancel
591:             </Button>
592:             <Button
593:                 onClick={confirmAddField}
594:                 disabled={!newEntryTitle.trim()}
595:                 className="bg-red-600 hover:bg-red-700 text-white"
596:             >
597:                 <Plus className="w-4 h-4 mr-2" />
598:                 Create Entry
599:             </Button>
600:         </div>
601:     </div>
602: </div>
603: )}
604:
605: { /* Icons Modal */ }
606: {showIconsModal && (
607:     <div className="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center z-50 p-4">
608:         <div className="bg-white rounded-lg max-w-4xl w-full max-h-[80vh] overflow-hidden">
609:             <div className="flex items-center justify-between p-6 border-b border-gray-200">
610:                 <h2 className="text-xl font-bold text-gray-800">
611:                     Available Icons

```



```

612:         </h2>
613:         <Button
614:             variant="ghost"
615:             size="sm"
616:             onClick={() => setShowIconsModal(false)}
617:             className="text-gray-500 hover:text-gray-700"
618:         >
619:             <X className="w-5 h-5" />
620:         </Button>
621:     </div>
622:
623:     <div className="p-6 overflow-y-auto max-h-[calc(80vh-80px)]">
624:         <div className="space-y-8">
625:             <div className="space-y-6">
626:                 </* Label Icons List */>
627:                 {Object.keys(labelIcons).length > 0 && (
628:                     <div className="space-y-2">
629:                         <h3 className="text-md font-semibold text-gray-800">
630:                             Label Icons
631:                         </h3>
632:                         <ul className="divide-y border rounded-md">
633:                             {Object.entries(labelIcons).map(([label, url]) => {
634:                                 const fileName = url.split("/").pop();
635:                                 const folder = "addiction/icons";
636:
637:                                 return (
638:                                     <li
639:                                         key={label}
640:                                         className="flex items-center justify-between px-4 py-2 hover:bg-gray-50"
641:                                     >
642:                                         <div className="flex items-center gap-4">
643:                                             <Image
644:                                                 src={url}
645:                                                 alt={label}
646:                                                 width={32}
647:                                                 height={32}
648:                                                 className="rounded"
649:                                             />
650:                                             <span className="text-sm font-medium capitalize">
651:                                                 {label.replace(/_/g, " ")}
652:                                             </span>
653:                                         </div>
654:                                         <Button
655:                                             size="icon"
656:                                             variant="destructive"
657:                                             className="h-6 w-6"
658:                                             onClick={async () => {
659:                                                 const confirmed = confirm(
660:                                                     `Are you sure you want to delete "${label}"?`
661:                                                 );
662:                                                 if (!confirmed) return;
663:
664:                                                 try {
665:                                                     const res = await fetch(
666:                                                         `/api/addiction-images?file=${fileName}&folder=${folder}`,
667:                                                         { method: "DELETE" }
668:                                                     );
669:                                                     const result = await res.json();
670:                                                     if (result.success) {
671:                                                         const updated = { ...labelIcons };
672:                                                         delete updated[label];
673:                                                         setLabelIcons(updated);
674:                                                     } else {
675:                                                         alert("Delete failed");
676:                                                     }
677:                                                 } catch (err) {
678:                                                     console.error(
679:                                                         "Failed to delete icon:",
680:                                                         err
681:                                                     );
682:                                                     alert("Delete failed");
683:                                                 }
684:                                             }

```

```

685:         >
686:         <Trash2 className="w-4 h-4" />
687:     </Button>
688: </li>
689:     );
690:   })}
691: </ul>
692: </div>
693: )}
694:
695: /* Sensitivity Icons List */
696: {Object.keys(sensitivityIcons).length > 0 && (
697:   <div className="space-y-2">
698:     <h3 className="text-md font-semibold text-gray-800">
699:       Sensitivity Icons
700:     </h3>
701:     <ul className="divide-y border rounded-md">
702:       {Object.entries(sensitivityIcons).map(
703:         ([label, url]) => {
704:           const fileName = url.split("/").pop();
705:           const folder = "addiction/sensitivity";
706:
707:           return (
708:             <li
709:               key={label}
710:               className="flex items-center justify-between px-4 py-2 hover:bg-gray-50"
711:             >
712:               <div className="flex items-center gap-4">
713:                 <Image
714:                   src={url}
715:                   alt={label}
716:                   width={32}
717:                   height={32}
718:                   className="rounded"
719:                 />
720:                 <span className="text-sm font-medium capitalize">
721:                   {label.replace(/_/g, " ")}
722:                 </span>
723:               </div>
724:               <Button
725:                 size="icon"
726:                 variant="destructive"
727:                 className="h-6 w-6"
728:                 onClick={async () => {
729:                   const confirmed = confirm(
730:                     `Are you sure you want to delete "${label}"?`
731:                   );
732:                   if (!confirmed) return;
733:
734:                   try {
735:                     const res = await fetch(
736:                       `/api/addiction-images?file=${fileName}&folder=${folder}`,
737:                       { method: "DELETE" }
738:                     );
739:                     const result = await res.json();
740:                     if (result.success) {
741:                       const updated = {
742:                         ...sensitivityIcons,
743:                       };
744:                       delete updated[label];
745:                       setSensitivityIcons(updated);
746:                     } else {
747:                       alert("Delete failed");
748:                     }
749:                   } catch (err) {
750:                     console.error(
751:                       "Failed to delete icon:",
752:                       err
753:                     );
754:                     alert("Delete failed");
755:                   }
756:                 }}
757:               >

```

```

758:                 <Trash2 className="w-4 h-4" />
759:             </Button>
760:         </li>
761:     );
762:   }
763: })
764: </ul>
765: </div>
766: })
767: </div>
768: </div>
769: </div>
770: </div>
771: </div>
772: })
773: </div>
774: </div>
775: );
776: }

```

## ■ File: components\admin\genetic-parameters-admin.tsx

```

=====
1: "use client";
2:
3: import React, { JSX, useCallback, useEffect, useState } from "react";
4: import { Button } from "@/components/ui/button";
5: import { Input } from "@/components/ui/input";
6: import { Card, CardContent, CardHeader, CardTitle } from "@/components/ui/card";
7: import { Textarea } from "@/components/ui/textarea";
8: import {
9:   Select,
10:   SelectContent,
11:   SelectItem,
12:   SelectTrigger,
13:   SelectValue,
14: } from "@/components/ui/select";
15: import {
16:   AlertDialog,
17:   AlertDialogAction,
18:   AlertDialogCancel,
19:   AlertDialogContent,
20:   AlertDialogDescription,
21:   AlertDialogFooter,
22:   AlertDialogHeader,
23:   AlertDialogTitle,
24: } from "@/components/ui/alert-dialog";
25: import Image from "next/image";
26: import { toast } from "sonner";
27: import {
28:   X,
29:   Plus,
30:   Save,
31:   Upload,
32:   Edit3,
33:   FileImage,
34:   Trash2,
35:   AlertTriangle,
36: } from "lucide-react";
37: import type { GeneticCategory } from "@/types/report-types";
38:
39: // Custom hook for data fetching
40:
41: interface Props {
42:   categories: GeneticCategory[];
43:   onUpdateCategories: (updated: GeneticCategory[]) => void;
44:   onSave: () => void;
45:   onReset: () => void;
46: }
47:
48: export default function GeneticParametersAdmin({
49:   categories,

```

```

50:   onUpdateCategories,
51:   onSave,
52:   onReset,
53: }: Props): JSX.Element {
54:   const [imageErrors, setImageErrors] = useState<Set<string>>(new Set());
55:   const [selectedCategory, setSelectedCategory] = useState<string>("all");
56:   const [isEditing, setIsEditing] = useState(false);
57:   const [isSaving, setIsSaving] = useState(false);
58:
59:   // Delete Dialog state
60:   const [deleteDialog, setDeleteDialog] = useState<{
61:     open: boolean;
62:     type: "category" | "parameter";
63:     categoryId?: string;
64:     categoryName?: string;
65:     categoryIndex?: number;
66:     paramIndex?: number;
67:     paramName?: string;
68:   }>({
69:     open: false,
70:     type: "category",
71:   });
72:
73:   const handleCategoryChange = (
74:     categoryIndex: number,
75:     field: keyof GeneticCategory,
76:     value: string | boolean | number
77:   ): void => {
78:     const updated = [...categories];
79:     (updated[categoryIndex] as any)[field] = value;
80:     onUpdateCategories(updated);
81:   };
82:
83:   const handleParameterChange = (
84:     categoryIndex: number,
85:     paramIndex: number,
86:     value: string
87:   ): void => {
88:     const updated = [...categories];
89:     updated[categoryIndex].parameters[paramIndex] = value;
90:     onUpdateCategories(updated);
91:   };
92:
93:   const handleImageError = (categoryId: string): void => {
94:     const newErrors = new Set(imageErrors);
95:     newErrors.add(categoryId);
96:     setImageErrors(newErrors);
97:   };
98:
99:   const handleFileChange = async (
100:     categoryIndex: number,
101:     file: File
102:   ): Promise<void> => {
103:     const formData = new FormData();
104:     formData.append("file", file);
105:
106:     try {
107:       const res = await fetch("/api/upload-image", {
108:         method: "POST",
109:         body: formData,
110:       });
111:
112:       const result = await res.json();
113:
114:       if (result.success && result.url) {
115:         handleCategoryChange(categoryIndex, "imageUrl", result.url);
116:
117:         const categoryId = categories[categoryIndex].id;
118:         const newErrors = new Set(imageErrors);
119:         newErrors.delete(categoryId);
120:         setImageErrors(newErrors);
121:       } else {
122:         toast.error(result.error || "Upload failed");

```

```

123:     }
124:   } catch (err) {
125:     console.error("Upload error:", err);
126:     toast.error("Image upload failed");
127:   }
128: };
129:
130: const addCategory = (): void => {
131:   const newCategory: GeneticCategory = {
132:     id: `category_${Date.now()}`,
133:     category: "NEW CATEGORY",
134:     imageUrl: "",
135:     description: "",
136:     parameters: Array(15).fill(""),
137:     isActive: true,
138:     order: categories.length + 1,
139:   };
140:
141:   onUpdateCategories([...categories, newCategory]);
142: };
143:
144: const removeCategory = async (categoryId: string): Promise<void> => {
145:   const category = categories.find((cat) => cat.id === categoryId);
146:   setDeleteDialog({
147:     open: true,
148:     type: "category",
149:     categoryId,
150:     categoryName: category?.category || "Unknown Category",
151:   });
152: };
153:
154: const confirmRemoveCategory = async (): Promise<void> => {
155:   if (!deleteDialog.categoryId) return;
156:
157:   const updated = categories.filter(
158:     (cat) => cat.id !== deleteDialog.categoryId
159:   );
160:   onUpdateCategories(updated);
161:
162:   if (selectedCategory === deleteDialog.categoryId) {
163:     setSelectedCategory("all");
164:   }
165:
166:   setDeleteDialog({ open: false, type: "category" });
167:   toast.success("Category removed");
168: };
169:
170: const addParameter = (categoryIndex: number): void => {
171:   const updated = [...categories];
172:   updated[categoryIndex].parameters.push("");
173:   onUpdateCategories(updated);
174: };
175:
176: const removeParameter = (categoryIndex: number, paramIndex: number): void => {
177:   const category = categories[categoryIndex];
178:   const paramName = category.parameters[paramIndex] || "Empty Parameter";
179:
180:   setDeleteDialog({
181:     open: true,
182:     type: "parameter",
183:     categoryIndex,
184:     paramIndex,
185:     paramName,
186:     categoryName: category.category,
187:   });
188: };
189:
190: const confirmRemoveParameter = (): void => {
191:   if (
192:     deleteDialog.categoryIndex === undefined ||
193:     deleteDialog.paramIndex === undefined
194:   )
195:     return;

```

```

196:
197:     const updated = [...categories];
198:     updated[deleteDialog.categoryIndex].parameters.splice(
199:       deleteDialog.paramIndex,
200:       1
201:     );
202:     onUpdateCategories(updated);
203:
204:     setDeleteDialog({ open: false, type: "category" });
205:     toast.success("Parameter removed");
206:   };
207:
208:   const handleSave = async () => {
209:     setIsSaving(true);
210:     try {
211:       await onSave(); // delegate saving to parent
212:       toast.success("Changes saved successfully");
213:     } catch (error) {
214:       toast.error("Save failed");
215:     } finally {
216:       setIsSaving(false);
217:     }
218:   };
219:
220:   const filteredCategories =
221:     selectedCategory === "all"
222:       ? categories
223:       : categories.filter((cat) => cat.id === selectedCategory);
224:
225:   return (
226:     <div className="w-full max-w-8xl mx-auto ">
227:       <Card className="mb-4 sm:mb-6">
228:         <CardHeader className="bg-gradient-to-r from-blue-600 to-red-400 rounded-lg p-4 sm:p-6">
229:           <CardTitle className="text-base sm:text-xl md:text-2xl font-bold uppercase text-white text-center">
230:             Table of Contents
231:           </CardTitle>
232:         </CardHeader>
233:
234:         <CardHeader className="p-4 sm:p-6">
235:           <div className="flex flex-col space-y-3 sm:space-y-4">
236:             <div className="flex flex-col space-y-2 sm:space-y-0 sm:flex-row sm:items-center sm:justify-end">
237:               <Select
238:                 value={selectedCategory}
239:                 onChange={setSelectedCategory}
240:               >
241:                 <SelectTrigger className="w-full sm:w-48">
242:                   <SelectValue placeholder="Filter by category" />
243:                 </SelectTrigger>
244:                 <SelectContent>
245:                   <SelectItem value="all">All Categories</SelectItem>
246:                   {categories.map((cat) => (
247:                     <SelectItem key={cat.id} value={cat.id}>
248:                       {cat.category}
249:                     </SelectItem>
250:                   ))}
251:                 </SelectContent>
252:               </Select>
253:
254:               <div className="flex flex-col sm:flex-row gap-2 w-full sm:w-auto">
255:                 <Button
256:                   onClick={() => setIsEditing(!isEditing)}
257:                   variant="outline"
258:                   className="gap-2 text-xs sm:text-sm"
259:                 >
260:                   <Edit3 size={14} className="sm:size-4" />
261:                   <span className="hidden xs:inline">
262:                     {isEditing ? "View Mode" : "Edit Mode"}
263:                   </span>
264:                   <span className="xs:hidden">
265:                     {isEditing ? "View" : "Edit"}
266:                   </span>
267:                 </Button>
268:

```

```

269:         {isEditing && (
270:             <>
271:                 <Button
272:                     onClick={addCategory}
273:                     variant="outline"
274:                     className="gap-2 text-xs sm:text-sm"
275:                 >
276:                     <Plus size={14} className="sm:size-4" />
277:                     <span className="hidden xs:inline">Add Category</span>
278:                     <span className="xs:hidden">Add</span>
279:                 </Button>
280:                 <Button
281:                     onClick={handleSave}
282:                     disabled={isSaving}
283:                     className="gap-2 text-xs sm:text-sm"
284:                 >
285:                     <Save size={14} className="sm:size-4" />
286:                     <span className="hidden xs:inline">
287:                         {isSaving ? "Saving..." : "Save Changes"}
288:                     </span>
289:                     <span className="xs:hidden">
290:                         {isSaving ? "Saving..." : "Save"}
291:                     </span>
292:                 </Button>
293:             </>
294:         )}
295:     </div>
296: </div>
297: </div>
298: </CardHeader>
299:
300: <CardContent className="p-2 sm:p-6">
301:     {/* Category Table */}
302:     <div className="overflow-x-auto">
303:         <div className="min-w-full border border-gray-300 rounded-lg">
304:             {filteredCategories.map((category, categoryIndex) => {
305:                 const actualIndex = categories.findIndex(
306:                     (cat) => cat.id === category.id
307:                 );
308:                 return (
309:                     <div
310:                         key={category.id}
311:                         className="border-b border-black last:border-b-0"
312:                     >
313:                         {/* Category Row */}
314:                         <div className="border-b border-gray-300 last:border-b-0">
315:                             {/* Mobile Layout */}
316:                             <div className="block lg:hidden">
317:                                 {/* Category Header for Mobile */}
318:                                 <div className="border-b border-gray-300 p-3 sm:p-4 bg-gray-50">
319:                                     {isEditing ? (
320:                                         <div className="space-y-3">
321:                                             <Input
322:                                                 value={category.category}
323:                                                 onChange={(e) =>
324:                                                     handleCategoryChange(
325:                                                         actualIndex,
326:                                                         "category",
327:                                                         e.target.value
328:                                                     )
329:                                                 }
330:                                             className="text-center font-bold text-sm bg-white"
331:                                             placeholder="Category Name"
332:                                         </div>
333:                                         {/* <Textarea
334:                                             value={category.description || ""}
335:                                             onChange={(e) =>
336:                                                 handleCategoryChange(
337:                                                     actualIndex,
338:                                                     "description",
339:                                                     e.target.value
340:                                                 )
341:                                             }

```

```

342:     }
343:     className="text-xs bg-white resize-none h-16"
344:     placeholder="Category description..."
345:   /> */}
346:
347:   <div className="flex flex-col items-center space-y-2">
348:     {category.imageUrl &&
349:     !imageErrors.has(category.id) ? (
350:       <Image
351:         src={category.imageUrl}
352:         alt={category.category}
353:         width={50}
354:         height={50}
355:         className="rounded-full border object-cover"
356:         onError={() =>
357:           handleImageError(category.id)
358:         }
359:       />
360:     ) : (
361:       <div className="w-[50px] h-[50px] bg-white rounded-full border-2 border-d
362:         <Upload
363:           size={16}
364:           className="text-gray-400"
365:         />
366:       </div>
367:     )}
368:
369:     <label className="cursor-pointer">
370:       <input
371:         type="file"
372:         accept="image/*"
373:         className="hidden"
374:         onChange={(e) => {
375:           const file = e.target.files?.[0];
376:           if (file) {
377:             handleFileChange(actualIndex, file);
378:           }
379:         }}
380:       />
381:       <div className="text-xs text-blue-600 hover:text-blue-800 underline">
382:         Upload
383:       </div>
384:     </label>
385:   </div>
386:
387:   <Button
388:     variant="destructive"
389:     size="sm"
390:     onClick={() => removeCategory(category.id)}
391:     className="gap-1 w-full text-xs"
392:   >
393:     <Trash2 size={12} />
394:     Remove
395:   </Button>
396: </div>
397: ) : (
398:   <div className="text-center">
399:     <h2 className="text-sm sm:text-base font-bold text-gray-700 mb-2">
400:       {category.category}
401:     </h2>
402:
403:     {category.description && (
404:       <p className="text-xs text-gray-600 mb-3 leading-tight">
405:         {category.description}
406:       </p>
407:     )}
408:
409:     {category.imageUrl &&
410:     !imageErrors.has(category.id) ? (
411:       <Image
412:         src={category.imageUrl}
413:         alt={category.category}
414:         width={50}

```



```

415:             height={50}
416:             className="rounded-full border object-cover mx-auto"
417:             onError={() => handleImageError(category.id)}
418:         />
419:     ) : (
420:         <div className="w-[50px] h-[50px] bg-gray-100 rounded-full border-2 border-gray-400" >
421:             <Upload size={20} className="text-gray-400" />
422:         </div>
423:     )}
424: </div>
425: )}
426: </div>
427:
428: { /* Parameters for Mobile - Single Column Layout */ }
429: <div className="p-3 sm:p-4">
430:     <div className="grid grid-cols-1 sm:grid-cols-2 gap-2">
431:         {category.parameters.map(
432:             (parameter, paramIndex) => (
433:                 <div
434:                     key={paramIndex}
435:                     className="border border-gray-300 rounded p-2"
436:                 >
437:                     {isEditing ? (
438:                         <div className="flex items-center space-x-1">
439:                             <Input
440:                                 value={parameter}
441:                                 onChange={(e) =>
442:                                     handleParameterChange(
443:                                         actualIndex,
444:                                         paramIndex,
445:                                         e.target.value
446:                                     )
447:                             }
448:                             placeholder="Parameter name"
449:                             className="text-xs h-7 flex-1"
450:                         />
451:                         <Button
452:                             variant="ghost"
453:                             size="sm"
454:                             onClick={() =>
455:                                 removeParameter(
456:                                     actualIndex,
457:                                     paramIndex
458:                                 )
459:                         }
460:                         className="h-7 w-7 p-0 text-red-500 hover:text-red-700 hover:bg-red-100"
461:                     >
462:                         <X size={10} />
463:                     </Button>
464:                 </div>
465:             ) : (
466:                 parameter && (
467:                     <span className="text-xs text-gray-700">
468:                         {parameter}
469:                     </span>
470:                 )
471:             )}
472:         </div>
473:     )
474: )}
475: </div>
476:
477: {isEditing && (
478:     <Button
479:         variant="ghost"
480:         size="sm"
481:         onClick={() => addParameter(actualIndex)}
482:         className="mt-3 w-full text-xs text-gray-500 hover:text-gray-700"
483:     >
484:         <Plus size={12} className="mr-1" />
485:         Add Parameter
486:     </Button>
487: )}

```

```

488:         </div>
489:     </div>
490:
491:     { /* Desktop Layout */ }
492:     <div className="hidden lg:grid lg:grid-cols-4 min-h-[140px]">
493:         { /* Category Name & Image Column */ }
494:         <div className="border-r border-gray-300 p-4 flex flex-col items-center justify-cen
495:             {isEditing ? (
496:                 <div className="space-y-3 w-full">
497:                     <Input
498:                         value={category.category}
499:                         onChange={(e) =>
500:                             handleCategoryChange(
501:                                 actualIndex,
502:                                 "category",
503:                                 e.target.value
504:                             )
505:                     }
506:                     className="text-center font-bold text-sm bg-white"
507:                     placeholder="Category Name"
508:                 />
509:                 { /*
510:                 <Textarea
511:                     value={category.description || ""}
512:                     onChange={(e) =>
513:                         handleCategoryChange(
514:                             actualIndex,
515:                             "description",
516:                             e.target.value
517:                         )
518:                     }
519:                     className="text-xs bg-white resize-none h-16"
520:                     placeholder="Category description..."
521:                 /> */ }
522:
523:                 <div className="flex flex-col items-center space-y-2">
524:                     {category.imageUrl &&
525:                     !imageErrors.has(category.id) ? (
526:                         <Image
527:                             src={category.imageUrl}
528:                             alt={category.category}
529:                             width={60}
530:                             height={60}
531:                             className="rounded-full border object-cover"
532:                             onError={() =>
533:                                 handleImageError(category.id)
534:                             }
535:                         />
536:                     ) : (
537:                         <div className="w-[60px] h-[60px] bg-white rounded-full border-2 border-d
538:                             <Upload
539:                                 size={20}
540:                                 className="text-gray-400"
541:                             />
542:                         </div>
543:                     ) }
544:
545:                     <label className="cursor-pointer">
546:                         <input
547:                             type="file"
548:                             accept="image/*"
549:                             className="hidden"
550:                             onChange={(e) => {
551:                                 const file = e.target.files?.[0];
552:                                 if (file) {
553:                                     handleFileChange(actualIndex, file);
554:                                 }
555:                             } }
556:                         />
557:                     <div className="text-xs text-blue-600 hover:text-blue-800 underline">
558:                         Upload
559:                     </div>
560:                 </label>

```

```

561:     </div>
562:
563:     <Button
564:       variant="destructive"
565:       size="sm"
566:       onClick={() => removeCategory(category.id)}
567:       className="gap-1 w-full"
568:     >
569:       <Trash2 size={14} />
570:       Remove
571:     </Button>
572:   </div>
573: ) : (
574:   <div className="text-center">
575:     <h2 className="text-base font-bold text-gray-700 mb-2">
576:       {category.category}
577:     </h2>
578:
579:     {category.description && (
580:       <p className="text-xs text-gray-600 mb-3 leading-tight">
581:         {category.description}
582:       </p>
583:     )}
584:
585:     {category.imageUrl &&
586:       !imageErrors.has(category.id) ? (
587:         <Image
588:           src={category.imageUrl}
589:           alt={category.category}
590:           width={70}
591:           height={70}
592:           className="rounded-full border object-cover mx-auto"
593:           onError={() => handleImageError(category.id)}
594:         />
595:       ) : (
596:         <div className="w-[70px] h-[70px] bg-gray-100 rounded-full border-2 border-
597:           <Upload size={24} className="text-gray-400" />
598:         </div>
599:       )}
600:   </div>
601: )}
602: </div>
603:
604: {/* Parameters Columns */}
605: <div className="col-span-3">
606:   <table className="w-full table-fixed border border-gray-300">
607:     <tbody>
608:       {Array.from(
609:         length: Math.ceil(
610:           category.parameters.length / 3
611:         ),
612:       )}.map((_, rowIndex) => (
613:         <tr key={rowIndex}>
614:           {[0, 1, 2].map((colIndex) => {
615:             const paramIndex =
616:               colIndex *
617:                 Math.ceil(
618:                   category.parameters.length / 3
619:                 ) +
620:               rowIndex;
621:             const parameter =
622:               category.parameters[paramIndex] || "";
623:
624:             return (
625:               <td
626:                 key={colIndex}
627:                 className="border border-gray-300 p-1"
628:               >
629:                 {isEditing ? (
630:                   <div className="flex items-center space-x-1 w-full">
631:                     <Input
632:                       value={parameter}
633:                       onChange={(e) =>

```

```

634:         handleParameterChange(
635:             actualIndex,
636:             paramIndex,
637:             e.target.value
638:         )
639:     }
640:     placeholder="Parameter name"
641:     className="text-sm h-7 w-full"
642: />
643: {paramIndex <
644:     category.parameters.length && (
645:         <Button
646:             variant="ghost"
647:             size="sm"
648:             onClick={() =>
649:                 removeParameter(
650:                     actualIndex,
651:                     paramIndex
652:                 )
653:             }
654:             className="h-7 w-7 p-0 text-red-500"
655:         >
656:             <X size={12} />
657:         </Button>
658:     )}
659: </div>
660: ) : (
661:     parameter && (
662:         <span className="text-sm text-gray-700">
663:             {parameter}
664:         </span>
665:     )
666:     )}
667: </td>
668: );
669: }}}
670: </tr>
671: )}
672: </tbody>
673: </table>
674:
675: {isEditing && (
676:     <Button
677:         variant="ghost"
678:         size="sm"
679:         onClick={() => addParameter(actualIndex)}
680:         className="mt-2 w-full text-xs text-gray-500 hover:text-gray-700"
681:     >
682:         <Plus size={12} className="mr-1" />
683:         Add Parameter
684:     </Button>
685: )}
686: </div>
687: </div>
688: </div>
689: </div>
690: );
691: }}}
692: </div>
693: </div>
694:
695: {categories.length === 0 && (
696:     <div className="text-center py-8 sm:py-12">
697:         <div className="text-gray-400 mb-4">
698:             <FileImage size={36} className="sm:size-12 mx-auto" />
699:         </div>
700:         <h3 className="text-base sm:text-lg font-semibold text-gray-700 mb-2">
701:             No Categories Found
702:         </h3>
703:         <p className="text-sm sm:text-base text-gray-500 mb-4">
704:             Get started by adding your first category
705:         </p>
706:         <Button onClick={addCategory} className="gap-2 text-sm">

```

```

707:         <Plus size={14} className="sm:size-4" />
708:         Add First Category
709:     </Button>
710: </div>
711: )}
712: </CardContent>
713: </Card>
714:
715: { /* Delete Confirmation Dialog */}
716: <AlertDialog
717:   open={deleteDialog.open}
718:   onOpenChange={({open}) => setDeleteDialog({ ...deleteDialog, open })}}
719: >
720:   <AlertDialogContent className="w-[90vw] max-w-md mx-auto">
721:     <AlertDialogHeader className="space-y-3">
722:       <div className="flex items-start gap-3">
723:         <div className="flex-shrink-0 w-8 h-8 sm:w-10 sm:h-10 bg-red-100 rounded-full flex items-center justify-center">
724:           <AlertTriangle className="w-4 h-4 sm:w-5 sm:h-5 text-red-600" />
725:         </div>
726:         <div className="flex-1 min-w-0">
727:           <AlertDialogTitle className="text-base sm:text-lg font-semibold text-gray-900 leading-tight">
728:             {deleteDialog.type === "category"
729:               ? "Delete Category"
730:               : "Delete Parameter"}
731:           </AlertDialogTitle>
732:         </div>
733:       </div>
734:     </AlertDialogHeader>
735:
736:     <AlertDialogDescription className="text-sm sm:text-base text-gray-600 leading-relaxed px-2 sm:px-4">
737:       {deleteDialog.type === "category" ? (
738:         <>
739:           Are you sure you want to delete the category{" "}
740:           <span className="font-semibold text-gray-900">
741:             "{deleteDialog.categoryName}"
742:           </span>
743:           ?
744:           <br className="hidden sm:block" />
745:           <span className="sm:hidden"> </span>
746:           <span className="block sm:inline mt-2 sm:mt-0">
747:             This action will permanently remove the category and all its
748:             parameters. This cannot be undone.
749:           </span>
750:         </>
751:       ) : (
752:         <>
753:           Are you sure you want to delete the parameter{" "}
754:           <span className="font-semibold text-gray-900">
755:             "{deleteDialog.paramName}"
756:           </span>{" "}
757:           from{" "}
758:           <span className="font-semibold text-gray-900">
759:             "{deleteDialog.categoryName}"
760:           </span>
761:           ?
762:           <br className="hidden sm:block" />
763:           <span className="sm:hidden"> </span>
764:           <span className="block sm:inline mt-2 sm:mt-0">
765:             This action cannot be undone.
766:           </span>
767:         </>
768:       )}
769:     </AlertDialogDescription>
770:
771:     <AlertDialogFooter className="flex-col-reverse sm:flex-row gap-2 sm:gap-0 mt-4 sm:mt-6">
772:       <AlertDialogCancel className="w-full sm:w-auto order-2 sm:order-1">
773:         Cancel
774:       </AlertDialogCancel>
775:       <AlertDialogAction
776:         onClick={
777:           deleteDialog.type === "category"
778:             ? confirmRemoveCategory
779:             : confirmRemoveParameter

```

```

780:         }
781:         className="w-full sm:w-auto order-1 sm:order-2 bg-red-600 hover:bg-red-700 focus:ring-red-500
782:     >
783:     <Trash2 size={14} className="mr-2" />
784:     Delete{ " "}
785:     {deleteDialog.type === "category" ? "Category" : "Parameter"}
786:   </AlertDialogAction>
787: </AlertDialogFooter>
788: </AlertDialogContent>
789: </AlertDialog>
790: </div>
791: );
792: }

```

## File: components\admin\health-summary-admin.tsx

```

=====
1: "use client";
2:
3: import { Input } from "@components/ui/input";
4: import { Textarea } from "@components/ui/textarea";
5: import { Button } from "@components/ui/button";
6: import { HealthSummary, HealthSummaryEntry } from "@types/report-types";
7: import { Card, CardContent, CardHeader, CardTitle } from "@components/ui/card";
8: import { Trash2 } from "lucide-react";
9:
10: interface HealthSummaryAdminProps {
11:   healthSummary: HealthSummary;
12:   onDescriptionUpdate: (value: string) => void;
13:   onUpdate: (
14:     index: number,
15:     field: keyof HealthSummaryEntry,
16:     value: string
17:   ) => void;
18:   onAdd: () => void;
19:   onRemove: (index: number) => void;
20:   onSave: () => void;
21:   isLoading?: boolean;
22: }
23:
24: export default function HealthSummaryAdmin({
25:   healthSummary,
26:   onDescriptionUpdate,
27:   onUpdate,
28:   onAdd,
29:   onRemove,
30:   onSave,
31:   isLoading = false,
32: }: HealthSummaryAdminProps) {
33:   const { description = "", data = [] } = healthSummary || {};
34:
35:   return (
36:     <div className="space-y-6">
37:       {/* Section Header */}
38:       <Card className="border-0 shadow-lg">
39:         <CardHeader className="bg-gradient-to-r from-red-500 to-orange-500 rounded-lg p-4 sm:p-6 uppercase">
40:           <CardTitle className="text-lg sm:text-xl md:text-2xl font-bold flex items-center justify-center">
41:             Health Summary Management
42:           </CardTitle>
43:         </CardHeader>
44:       </Card>
45:
46:       {/* Top Actions */}
47:       <div className="flex flex-row flex-wrap justify-end gap-2">
48:         <Button
49:           onClick={onSave}
50:           disabled={isLoading}
51:           className="flex-1 sm:flex-initial min-w-[140px]"
52:         >
53:           {isLoading ? "Saving..." : "Save All Changes"}
54:         </Button>
55:         <Button

```

```

56:         onClick={onAdd}
57:         disabled={isLoading}
58:         className="flex-1 sm:flex-initial min-w-[140px]"
59:     >
60:         Add Summary
61:     </Button>
62: </div>
63:
64: { /* Section Description */}
65: <div>
66:     <label className="block text-sm font-medium text-gray-700 mb-2">
67:         Health Summary Description
68:     </label>
69:     <Input
70:         placeholder="Enter description for the health summary section..."
71:         value={description}
72:         onChange={(e) => onDescriptionUpdate(e.target.value)}
73:         disabled={isLoading}
74:         className="w-full"
75:     />
76: </div>
77:
78: { /* Empty State */}
79: {data.length === 0 && (
80:     <div className="text-center py-10 px-4 text-gray-500 border border-dashed rounded-lg bg-gray-50">
81:         <p className="text-lg font-medium mb-2">
82:             No health summary entries yet
83:         </p>
84:         <p className="text-sm">
85:             Click <strong>"Add Summary"</strong> to get started.
86:         </p>
87:     </div>
88: )}
89:
90: { /* Entry List */}
91: <div className="grid gap-6 md:grid-cols-3">
92:     {data.map((entry, index) => (
93:         <Card
94:             key={`health-summary-entry-${index}`}
95:             className="border rounded-xl shadow-sm hover:shadow-md border-gray-400 transition-shadow"
96:         >
97:             <CardHeader className="flex flex-row items-center justify-between p-4 bg-gray-50 rounded-t-xl">
98:                 <span className="bg-red-600 text-sm font-medium text-white px-3 py-1 rounded-full">
99:                     {index + 1}
100:                 </span>
101:                 <Button
102:                     variant="destructive"
103:                     size="icon"
104:                     onClick={() => onRemove(index)}
105:                     disabled={isLoading}
106:                 >
107:                     <Trash2 className="h-4 w-4" />
108:                 </Button>
109:             </CardHeader>
110:
111:             <CardContent className="space-y-4 p-4">
112:                 { /* Title */}
113:                 <div>
114:                     <label className="block text-sm font-medium text-gray-700 mb-2">
115:                         Title
116:                     </label>
117:                     <Input
118:                         placeholder="Enter entry title..."
119:                         value={entry.title}
120:                         onChange={(e) => onUpdate(index, "title", e.target.value)}
121:                         disabled={isLoading}
122:                     />
123:                 </div>
124:
125:                 { /* Description */}
126:                 <div>
127:                     <label className="block text-sm font-medium text-gray-700 mb-2">
128:                         Description

```

```

129:         </label>
130:         <Textarea
131:             placeholder="Enter detailed description..."
132:             value={entry.description}
133:             onChange={(e) =>
134:                 onUpdate(index, "description", e.target.value)
135:             }
136:             disabled={isLoading}
137:             className="w-full min-h-[120px] resize-none"
138:         />
139:     </div>
140: </CardContent>
141: </Card>
142:     )}
143: </div>
144: </div>
145: );
146: }

```

## ■ File: components\admin\import-export-admin.tsx

```

1: "use client";
2:
3: import React, { useState, useCallback } from "react";
4: import { Button } from "@/components/ui/button";
5: import {
6:   Card,
7:   CardContent,
8:   CardDescription,
9:   CardHeader,
10:  CardTitle,
11: } from "@/components/ui/card";
12: import { Label } from "@/components/ui/label";
13: import {
14:   Download,
15:   Upload,
16:   FileText,
17:   CheckCircle,
18:   AlertCircle,
19:   Clock,
20:   RefreshCw,
21:   Database,
22:   Shield,
23: } from "lucide-react";
24: import type { ComprehensiveReportData } from "@/types/report-types";
25:
26: interface ImportExportAdminProps {
27:   reportData: ComprehensiveReportData;
28:   setReportData: (data: ComprehensiveReportData) => void;
29: }
30:
31: // Mock toast hook for demonstration
32: const useToast = () => ({
33:   toast: ({ title, description, variant }: any) => {
34:     console.log(`Toast: ${title} - ${description} (${variant})`);
35:   },
36: });
37:
38: export default function ImportExportAdmin({
39:   reportData,
40:   setReportData,
41: }: ImportExportAdminProps) {
42:   const { toast } = useToast();
43:   const [isExporting, setIsExporting] = useState(false);
44:   const [isImporting, setIsImporting] = useState(false);
45:   const [dragActive, setDragActive] = useState(false);
46:   const [fileInfo, setFileInfo] = useState<{
47:     name: string;
48:     size: number;
49:     lastModified: number;
50:   } | null>(null);

```



```

51:
52: const formatFileSize = (bytes: number): string => {
53:     if (bytes === 0) return "0 Bytes";
54:     const k = 1024;
55:     const sizes = ["Bytes", "KB", "MB", "GB"];
56:     const i = Math.floor(Math.log(bytes) / Math.log(k));
57:     return parseFloat((bytes / Math.pow(k, i)).toFixed(2)) + " " + sizes[i];
58: };
59:
60: const validateImportData = (data: any): boolean => {
61:     if (!data || typeof data !== "object") return false;
62:
63:     // Check for required top-level properties
64:     if (!data.patientInfo) return false;
65:     if (!data.id || !data.content || !data.settings) return false;
66:
67:     // Check patientInfo structure
68:     if (!data.patientInfo.name || !data.patientInfo.sampleCode) return false;
69:
70:     // Check for essential report properties
71:     if (!data.content || typeof data.content !== "object") return false;
72:     if (!data.settings || typeof data.settings !== "object") return false;
73:
74:     return true;
75: };
76:
77: const exportData = async () => {
78:     setIsExporting(true);
79:     try {
80:         // Add metadata to export
81:         const exportData: ComprehensiveReportData & { metadata?: any } = {
82:             ...reportData,
83:             metadata: {
84:                 version: "2.0",
85:                 exportDate: new Date().toISOString(),
86:                 dataVersion: "1.0",
87:                 exportedBy: "Genomics Report System",
88:             },
89:         };
90:
91:         const dataStr = JSON.stringify(exportData, null, 2);
92:         const dataBlob = new Blob([dataStr], { type: "application/json" });
93:         const url = URL.createObjectURL(dataBlob);
94:
95:         const timestamp = new Date().toISOString().split("T")[0];
96:         const patientId = reportData.patientInfo?.sampleCode || "unknown";
97:         const filename = `genomics-report-${patientId}-${timestamp}.json`;
98:
99:         const link = document.createElement("a");
100:         link.href = url;
101:         link.download = filename;
102:         document.body.appendChild(link);
103:         link.click();
104:         document.body.removeChild(link);
105:         URL.revokeObjectURL(url);
106:
107:         toast({
108:             title: "Export Successful",
109:             description: `Report data exported as ${filename}`,
110:             variant: "success",
111:             duration: 3000,
112:         });
113:     } catch (error) {
114:         console.error("Export error:", error);
115:         toast({
116:             title: "Export Failed",
117:             description: "Unable to export data. Please try again.",
118:             variant: "destructive",
119:             duration: 4000,
120:         });
121:     } finally {
122:         setIsExporting(false);
123:     }

```

```

124: };
125:
126: const processImportFile = useCallback(
127:   async (file: File) => {
128:     setIsImporting(true);
129:     setFileInfo({
130:       name: file.name,
131:       size: file.size,
132:       lastModified: file.lastModified,
133:     });
134:
135:     try {
136:       // Check file size (max 10MB)
137:       if (file.size > 10 * 1024 * 1024) {
138:         throw new Error("File size too large. Maximum size is 10MB.");
139:       }
140:
141:       // Check file type
142:       if (!file.name.toLowerCase().endsWith(".json")) {
143:         throw new Error("Invalid file type. Please select a JSON file.");
144:       }
145:
146:       const fileText = await file.text();
147:       const importedData = JSON.parse(fileText);
148:
149:       // Validate data structure
150:       if (!validateImportData(importedData)) {
151:         throw new Error(
152:           "Invalid file format. Missing required patient info or report data."
153:         );
154:       }
155:
156:       // Check if it's an exported file from this system
157:       if (importedData.metadata?.exportedBy !== "Genomics Report System") {
158:         console.warn(
159:           "File may not be from this system, proceeding with caution..."
160:         );
161:       }
162:
163:       // Remove metadata before setting data (clean the imported data)
164:       const { metadata, ...cleanImportedData } = importedData;
165:
166:       setReportData(cleanImportedData as ComprehensiveReportData);
167:
168:       toast({
169:         title: "Import Successful",
170:         description: `Successfully imported data for patient: ${importedData.patientInfo.name}`,
171:         variant: "success",
172:         duration: 3000,
173:       });
174:     } catch (error: any) {
175:       console.error("Import error:", error);
176:       setFileInfo(null);
177:
178:       let errorMessage = "Unknown error occurred";
179:       if (error instanceof SyntaxError) {
180:         errorMessage = "Invalid JSON format";
181:       } else if (error.message) {
182:         errorMessage = error.message;
183:       }
184:
185:       toast({
186:         title: "Import Failed",
187:         description: errorMessage,
188:         variant: "destructive",
189:         duration: 5000,
190:       });
191:     } finally {
192:       setIsImporting(false);
193:     }
194:   },
195:   [setReportData, toast]
196: );

```

```

197:
198: const handleFileInput = (event: React.ChangeEvent<HTMLInputElement>) => {
199:   const file = event.target.files?.[0];
200:   if (file) {
201:     processImportFile(file);
202:   }
203: };
204:
205: const handleDrag = useCallback((e: React.DragEvent) => {
206:   e.preventDefault();
207:   e.stopPropagation();
208: }, []);
209:
210: const handleDragIn = useCallback((e: React.DragEvent) => {
211:   e.preventDefault();
212:   e.stopPropagation();
213:   setDragActive(true);
214: }, []);
215:
216: const handleDragOut = useCallback((e: React.DragEvent) => {
217:   e.preventDefault();
218:   e.stopPropagation();
219:   setDragActive(false);
220: }, []);
221:
222: const handleDrop = useCallback(
223:   (e: React.DragEvent) => {
224:     e.preventDefault();
225:     e.stopPropagation();
226:     setDragActive(false);
227:
228:     const files = e.dataTransfer.files;
229:     if (files?.[0]) {
230:       processImportFile(files[0]);
231:     }
232:   },
233:   [processImportFile]
234: );
235:
236: return (
237:   <div className="max-w-8xl mx-auto">
238:     <Card className="shadow-2xl border-0 bg-gradient-to-br from-white to-gray-50 overflow-hidden">
239:       <CardHeader className="bg-gradient-to-r from-indigo-600 via-purple-600 to-blue-600 text-white rounded">
240:         <CardTitle className="text-lg sm:text-xl md:text-2xl font-bold uppercase flex items-center justify">
241:           Data Management Hub
242:         </CardTitle>
243:       </CardHeader>
244:
245:       <CardContent className="p-8">
246:         <div className="grid lg:grid-cols-2 gap-10">
247:           <div>
248:             <Card className="border-2 border-green-200 bg-green-50/50 hover:shadow-lg transition-all duration">
249:               <CardHeader className="pb-4">
250:                 <div className="flex items-center gap-3 mb-2">
251:                   <div className="p-2 bg-green-100 rounded-full">
252:                     <Download className="h-6 w-6 text-green-600" />
253:                   </div>
254:                   <h3 className="text-2xl font-bold text-green-800">
255:                     Export Data
256:                   </h3>
257:                 </div>
258:                 <p className="text-green-700 leading-relaxed">
259:                   Create a secure backup of your current report configuration.
260:                   Includes all patient data, analysis results, and metadata.
261:                 </p>
262:               </CardHeader>
263:
264:               <CardContent className="space-y-4">
265:                 <div className="bg-white rounded-lg p-4 border border-green-200">
266:                   <div className="flex items-center gap-2 text-sm text-gray-600 mb-2">
267:                     <FileText className="h-4 w-4" />
268:                     <span>Current Data Summary</span>
269:                   </div>

```

```

270:         <div className="text-sm space-y-1 text-gray-700">
271:             <div>
272:                 Patient:{ " "}
273:                 <span className="font-medium">
274:                     {reportData.patientInfo?.name || "No name"}
275:                 </span>
276:             </div>
277:             <div>
278:                 Patient ID:{ " "}
279:                 <span className="font-medium">
280:                     {reportData.patientInfo?.sampleCode || "No ID"}
281:                 </span>
282:             </div>
283:             <div>
284:                 Data Size:{ " "}
285:                 <span className="font-medium">
286:                     {formatFileSize(JSON.stringify(reportData).length)}
287:                 </span>
288:             </div>
289:         </div>
290:     </div>
291:
292:     <Button
293:         onClick={exportData}
294:         disabled={isExporting}
295:         className="w-full bg-green-600 hover:bg-green-700 text-white py-3 text-lg font-medium tra
296:     >
297:         {isExporting ? (
298:             <>
299:                 <RefreshCw className="h-5 w-5 mr-2 animate-spin" />
300:                 Exporting...
301:             </>
302:         ) : (
303:             <>
304:                 <Download className="h-5 w-5 mr-2" />
305:                 Export Current Data
306:             </>
307:         )}
308:     </Button>
309: </CardContent>
310: </Card>
311:
312: { /* Import Section */}
313: <Card className="border-2 border-blue-200 bg-blue-50/50 hover:shadow-lg transition-all duration
314:     <CardHeader className="pb-4">
315:         <div className="flex items-center gap-3 mb-2">
316:             <div className="p-2 bg-blue-100 rounded-full">
317:                 <Upload className="h-6 w-6 text-blue-600" />
318:             </div>
319:             <h3 className="text-2xl font-bold text-blue-800">
320:                 Import Data
321:             </h3>
322:         </div>
323:         <p className="text-blue-700 leading-relaxed">
324:             Restore previously exported data or load a new report
325:             configuration. Supports drag & drop for easy file selection.
326:         </p>
327:     </CardHeader>
328:
329:     <CardContent className="space-y-4">
330:         <div
331:             className={`
332:                 border-3 border-dashed rounded-xl p-8 text-center transition-all duration-300 cursor-po
333:             ${
334:                 dragActive
335:                 ? "border-blue-500 bg-blue-100/70 scale-105"
336:                 : "border-blue-300 bg-white hover:border-blue-400 hover:bg-blue-50/50"
337:             }
338:             ${isImporting ? "pointer-events-none opacity-60" : ""}
339:         `>
340:             onDragEnter={handleDragIn}
341:             onDragLeave={handleDragOut}
342:             onDragOver={handleDrag}

```

```

343:         onDrop={handleDrop}
344:     >
345:         <input
346:             type="file"
347:             accept=".json"
348:             onChange={handleFileInput}
349:             className="hidden"
350:             id="import-file"
351:             disabled={isImporting}
352:         />
353:     <Label htmlFor="import-file" className="cursor-pointer block">
354:         {isImporting ? (
355:             <div className="space-y-3">
356:                 <RefreshCw className="h-12 w-12 mx-auto text-blue-500 animate-spin" />
357:                 <div className="text-lg font-medium text-blue-700">
358:                     Processing File...
359:                 </div>
360:                 {fileInfo && (
361:                     <div className="text-sm text-blue-600">
362:                         {fileInfo.name} ({formatFileSize(fileInfo.size)})
363:                     </div>
364:                 )}
365:             </div>
366:         ) : dragActive ? (
367:             <div className="space-y-3">
368:                 <Upload className="h-12 w-12 mx-auto text-blue-500" />
369:                 <div className="text-lg font-medium text-blue-700">
370:                     Drop your JSON file here!
371:                 </div>
372:             </div>
373:         ) : (
374:             <div className="space-y-3">
375:                 <Upload className="h-12 w-12 mx-auto text-blue-400" />
376:                 <div className="text-lg font-medium text-blue-700">
377:                     Click to select or drag & drop
378:                 </div>
379:                 <div className="text-sm text-blue-600">
380:                     Supports JSON files up to 10MB
381:                 </div>
382:             </div>
383:         )}
384:     </Label>
385: </div>
386:
387: {fileInfo && !isImporting && (
388:     <div className="bg-white rounded-lg p-4 border border-blue-200">
389:         <div className="flex items-center gap-2 text-sm text-gray-600 mb-2">
390:             <CheckCircle className="h-4 w-4 text-green-500" />
391:             <span>File Ready for Import</span>
392:         </div>
393:         <div className="text-sm space-y-1 text-gray-700">
394:             <div>
395:                 File:{" "}
396:                 <span className="font-medium">{fileInfo.name}</span>
397:             </div>
398:             <div>
399:                 Size:{" "}
400:                 <span className="font-medium">
401:                     {formatFileSize(fileInfo.size)}
402:                 </span>
403:             </div>
404:             <div>
405:                 Modified:{" "}
406:                 <span className="font-medium">
407:                     {new Date(fileInfo.lastModified).toLocaleDateString()}
408:                 </span>
409:             </div>
410:         </div>
411:     </div>
412: )}
413: </CardContent>
414: </Card>
415: </div>

```

```

416:
417:     { /* Security & Guidelines Section */ }
418:     <Card className="mt-8 border-2 border-amber-200 bg-amber-50/70">
419:         <CardHeader className="pb-4">
420:             <div className="flex items-center gap-3"></div>
421:         </CardHeader>
422:         <CardContent>
423:             <div className="grid md:grid-cols-2 gap-6">
424:                 <div className="space-y-3">
425:                     <ul className="text-sm text-amber-700 space-y-2 ml-6">
426:                         <li className="flex items-start gap-2">
427:                             <span className="text-amber-500 mt-1">?</span>
428:                             Always export current data as backup first
429:                         </li>
430:                         <li className="flex items-start gap-2">
431:                             <span className="text-amber-500 mt-1">?</span>
432:                             Verify file source and authenticity
433:                         </li>
434:                         <li className="flex items-start gap-2">
435:                             <span className="text-amber-500 mt-1">?</span>
436:                             Check file size and modification date
437:                         </li>
438:                     </ul>
439:                 </div>
440:
441:                 <div className="space-y-3">
442:                     <ul className="text-sm text-amber-700 space-y-2 ml-6">
443:                         <li className="flex items-start gap-2">
444:                             <span className="text-amber-500 mt-1">?</span>
445:                             Imports completely replace current configuration
446:                         </li>
447:                         <li className="flex items-start gap-2">
448:                             <span className="text-amber-500 mt-1">?</span>
449:                             Only use JSON files from this system
450:                         </li>
451:                         <li className="flex items-start gap-2">
452:                             <span className="text-amber-500 mt-1">?</span>
453:                             Save changes immediately after importing
454:                         </li>
455:                     </ul>
456:                 </div>
457:             </div>
458:         </CardContent>
459:     </Card>
460: </CardContent>
461: </Card>
462: </div>
463: );
464: }

```

## **File: components\admin\lifestyle-conditions-admin.tsx**

```

=====
1: "use client";
2:
3: import React, { useState, useEffect } from "react";
4: import {
5:   Trash2,
6:   Upload,
7:   Plus,
8:   Save,
9:   ImageIcon,
10:  Edit3,
11:  Check,
12:  FileText,
13:  Activity,
14:  AlertTriangle,
15: } from "lucide-react";
16:
17: import { Label } from "@/components/ui/label";
18: import {
19:   Select,

```

```

20:   SelectContent,
21:   SelectItem,
22:   SelectTrigger,
23:   SelectValue,
24: } from "@components/ui/select";
25: import { Button } from "@components/ui/button";
26: import {
27:   Card,
28:   CardContent,
29:   CardHeader,
30:   CardTitle,
31:   CardDescription,
32: } from "@components/ui/card";
33: import { Input } from "@components/ui/input";
34: import { Textarea } from "@components/ui/textarea";
35:
36: // Types (these would be imported from your types file)
37: interface HealthConditionStatus {
38:   status: "strength" | "improvement";
39:   description?: string;
40:   sensitivity?: "low" | "medium" | "high";
41:   avoid?: string[];
42:   follow?: string[];
43:   consume?: string[];
44:   monitor?: string[];
45:   avoidLabel?: string;
46:   followLabel?: string;
47:   consumeLabel?: string;
48:   monitorLabel?: string;
49: }
50:
51: interface LifestyleConditions {
52:   quote?: string;
53:   description?: string;
54:   [categoryId: string]:
55:     | Record<string, HealthConditionStatus>
56:     | string
57:     | undefined;
58: }
59:
60: interface LifestyleCategoryImages {
61:   [categoryId: string]: string;
62: }
63:
64: type EditableLabelKey =
65:   | "avoidLabel"
66:   | "followLabel"
67:   | "consumeLabel"
68:   | "monitorLabel";
69:
70: interface Props {
71:   lifestyleConditions: LifestyleConditions;
72:   lifestyleCategoryImages: LifestyleCategoryImages;
73:   addCategory: (categoryId: string) => void;
74:   addField: (
75:     categoryId: string,
76:     fieldName: string,
77:     status: HealthConditionStatus["status"]
78:   ) => void;
79:   updateFieldStatus: (
80:     categoryId: string,
81:     fieldName: string,
82:     updated: Partial<HealthConditionStatus>
83:   ) => void;
84:   deleteField: (categoryId: string, fieldName: string) => void;
85:   deleteCategory: (categoryId: string) => void;
86:   updateQuoteAndDescription: (quote: string, description: string) => void;
87:   updateCategoryImage: (categoryId: string, imageUrl: string) => void;
88:   onSave: () => void;
89:   onReset: () => void;
90: }
91:
92: export default function LifestyleConditionsAdmin({

```

```

93:   lifestyleConditions,
94:   lifestyleCategoryImages,
95:   updateQuoteAndDescription,
96:   updateCategoryImage,
97:   addCategory,
98:   addField,
99:   updateFieldStatus,
100:  deleteCategory,
101:  deleteField,
102:  onSave,
103:  onReset,
104: } : Props) {
105:   const [newCategoryName, setNewCategoryName] = useState("");
106:   const [newField, setNewField] = useState<Record<string, string>>({});
107:   const [availableImages, setAvailableImages] = useState<
108:     { label: string; url: string }[]
109:   >([]);
110:   const [newImageFile, setNewImageFile] = useState<File | null>(null);
111:   const [newImageLabel, setNewImageLabel] = useState("");
112:   const [showSensitivityDropdown, setShowSensitivityDropdown] = useState<{
113:     [key: string]: boolean;
114:   }>({});
115:
116:   // Confirmation popup states
117:   const [deleteConfirmation, setDeleteConfirmation] = useState<{
118:     type: "category" | "field" | null;
119:     categoryId?: string;
120:     fieldName?: string;
121:     categoryName?: string;
122:   }>({ type: null });
123:
124:   const quote = lifestyleConditions.quote ?? "";
125:   const description = lifestyleConditions.description ?? "";
126:
127:   const [editingLabels, setEditingLabels] = useState<{
128:     [categoryId: string]: {
129:       [field: string]: {
130:         [key in EditableLabelKey]?: boolean;
131:       };
132:     };
133:   }>({});
134:
135:   const toggleEditLabel = (
136:     categoryId: string,
137:     field: string,
138:     key: EditableLabelKey
139:   ) => {
140:     setEditingLabels((prev) => ({
141:       ...prev,
142:       [categoryId]: {
143:         ...prev[categoryId],
144:         [field]: {
145:           ...prev[categoryId]?.[field],
146:           [key]: !prev[categoryId]?.[field]?.[key],
147:         },
148:       },
149:     }));
150:   };
151:
152:   const loadAvailableImages = async () => {
153:     try {
154:       const res = await fetch("/api/lifestyle-images?folder=lifestyle");
155:       const data = await res.json();
156:       if (data.success) {
157:         setAvailableImages(data.images);
158:       }
159:     } catch (err) {
160:       console.error("Failed to load lifestyle images:", err);
161:     }
162:   };
163:
164:   useEffect(() => {
165:     loadAvailableImages();

```



```

166:   }, []);
167:
168:   const handleImageUpload = async () => {
169:     if (!newImageFile || !newImageLabel) return;
170:
171:     const formData = new FormData();
172:     formData.append("file", newImageFile);
173:     formData.append("label", newImageLabel);
174:     formData.append("folder", "lifestyle");
175:
176:     try {
177:       const response = await fetch("/api/lifestyle-images", {
178:         method: "POST",
179:         body: formData,
180:       });
181:
182:       const result = await response.json();
183:       if (result.success) {
184:         setNewImageFile(null);
185:         setNewImageLabel("");
186:         loadAvailableImages(); // refresh list
187:         alert("Image uploaded successfully!");
188:       }
189:     } catch (error) {
190:       console.error("Failed to upload image:", error);
191:     }
192:   };
193:
194:   const handleDeleteCategory = (categoryId: string) => {
195:     setDeleteConfirmation({
196:       type: "category",
197:       categoryId,
198:       categoryName: categoryId.replace(/[A-Z]/g, " $1"),
199:     });
200:   };
201:
202:   const handleDeleteField = (categoryId: string, fieldName: string) => {
203:     setDeleteConfirmation({
204:       type: "field",
205:       categoryId,
206:       fieldName,
207:       categoryName: categoryId.replace(/[A-Z]/g, " $1"),
208:     });
209:   };
210:
211:   const confirmDelete = () => {
212:     if (
213:       deleteConfirmation.type === "category" &&
214:       deleteConfirmation.categoryId
215:     ) {
216:       deleteCategory(deleteConfirmation.categoryId);
217:     } else if (
218:       deleteConfirmation.type === "field" &&
219:       deleteConfirmation.categoryId &&
220:       deleteConfirmation.fieldName
221:     ) {
222:       deleteField(deleteConfirmation.categoryId, deleteConfirmation.fieldName);
223:     }
224:     setDeleteConfirmation({ type: null });
225:   };
226:
227:   const cancelDelete = () => {
228:     setDeleteConfirmation({ type: null });
229:   };
230:
231:   return (
232:     <div className="min-h-screen bg-gradient-to-br from-slate-50 to-blue-50 ">
233:       <div className="max-w-8xl mx-auto space-y-8">
234:         {/* Header Section */}
235:         <Card className="shadow-lg border-0">
236:           <CardHeader className="bg-gradient-to-r from-blue-600 to-indigo-700 text-white rounded-lg p-4 sm:p-6">
237:             <CardTitle className="text-base sm:text-xl md:text-2xl font-bold uppercase flex items-center justify-center">
238:               Lifestyle Conditions

```

```

239:         </CardTitle>
240:     </CardHeader>
241:
242:     <div className="text-end mt-4">
243:         <Button onClick={onSave}>
244:             <Save className="h-3 w-3 mr-2" />
245:             Save Changes
246:         </Button>
247:     </div>
248:
249:     <CardContent className="p-6 space-y-6">
250:         {/* Quote and Description */}
251:         <div className="grid md:grid-cols-2 gap-6">
252:             <div className="space-y-3">
253:                 <div className="flex items-center gap-2">
254:                     <Label className="font-semibold text-gray-800">
255:                         Motivational Quote
256:                     </Label>
257:                 </div>
258:                 <Input
259:                     value={quote}
260:                     onChange={(e) =>
261:                         updateQuoteAndDescription(e.target.value, description)
262:                     }
263:                     placeholder="Enter an inspiring quote..."
264:                     className="border-2 border-gray-200 focus:border-blue-500"
265:                 />
266:             </div>
267:
268:             <div className="space-y-3">
269:                 <div className="flex items-center gap-2">
270:                     <Label className="font-semibold text-gray-800">
271:                         Add New Category
272:                     </Label>
273:                 </div>
274:                 <div className="flex gap-3">
275:                     <Input
276:                         placeholder="Category name (e.g., Nutrition, Exercise)..."
277:                         value={newCategoryName}
278:                         onChange={(e) => setNewCategoryName(e.target.value)}
279:                         className="border-2 border-gray-200 focus:border-green-500"
280:                     />
281:                     <Button
282:                         onClick={() => {
283:                             if (newCategoryName.trim()) {
284:                                 addCategory(newCategoryName.trim());
285:                                 setNewCategoryName("");
286:                             }
287:                         }}
288:                         className="bg-green-600 hover:bg-green-700"
289:                     >
290:                         <Plus className="w-4 h-4 mr-1" />
291:                         Add
292:                     </Button>
293:                 </div>
294:             </div>
295:         </div>
296:
297:         <div className="space-y-3">
298:             <div className="flex items-center gap-2">
299:                 <Label className="font-semibold text-gray-800">
300:                     Introduction Description
301:                 </Label>
302:             </div>
303:             <Textarea
304:                 value={description}
305:                 onChange={(e) =>
306:                     updateQuoteAndDescription(quote, e.target.value)
307:                 }
308:                 placeholder="Write a comprehensive introduction..."
309:                 rows={4}
310:                 className="border-2 border-gray-200 focus:border-purple-500 resize-none"
311:             />

```

```

312:         </div>
313:     </CardContent>
314: </Card>
315:
316: { /* Image Upload Section */}
317: <Card className="shadow-lg border-0">
318:     <CardHeader>
319:         <CardTitle className="flex items-center gap-2">
320:             Image Library Management
321:         </CardTitle>
322:     </CardHeader>
323:     <CardContent className="space-y-4">
324:         <div className="grid md:grid-cols-3 gap-4 items-end">
325:             <div>
326:                 <Label htmlFor="imageLabel">Image Label</Label>
327:                 <Input
328:                     id="imageLabel"
329:                     placeholder="e.g., sleep, nutrition, exercise"
330:                     value={newImageLabel}
331:                     onChange={(e) => setNewImageLabel(e.target.value)}
332:                 />
333:             </div>
334:             <div>
335:                 <Label htmlFor="imageFile">Select Image</Label>
336:                 <Input
337:                     id="imageFile"
338:                     type="file"
339:                     accept="image/*"
340:                     onChange={(e) => setNewImageFile(e.target.files?.[0] || null)}
341:                 />
342:             </div>
343:             <Button
344:                 disabled={!newImageLabel || !newImageFile}
345:                 onClick={handleImageUpload}
346:                 className="bg-blue-600 hover:bg-blue-700"
347:             >
348:                 <Upload className="w-4 h-4 mr-2" />
349:                 Upload Image
350:             </Button>
351:         </div>
352:     </CardContent>
353: </Card>
354:
355: { /* Categories Grid */}
356: <div className="space-y-6">
357:     {Object.entries(lifestyleConditions)
358:         .filter(
359:             ([categoryId]) => !["quote", "description"].includes(categoryId)
360:         )
361:         .map(([categoryId, conditions]) => {
362:             const fieldMap = conditions as Record<
363:                 string,
364:                 HealthConditionStatus
365:             >;
366:
367:             return (
368:                 <Card
369:                     key={categoryId}
370:                     className="shadow-lg rounded-lg border-0"
371:                 >
372:                     <CardHeader>
373:                         <div className="flex flex-col sm:flex-row justify-between items-start sm:items-center g
374:                         <div className="flex items-center gap-4">
375:                             { /* Category Image Preview */}
376:                             {lifestyleCategoryImages[categoryId] && (
377:                                 <div className="w-16 h-16 rounded-lg overflow-hidden">
378:                                     <img
379:                                         src={lifestyleCategoryImages[categoryId]}
380:                                         alt={categoryId}
381:                                         className="w-full h-full object-cover"
382:                                     />
383:                                 </div>
384:                             )}

```

```

385:         <div>
386:             <h3 className="text-xl font-bold text-gray-800 capitalize">
387:                 {categoryId
388:                     .replace(/[a-z]([A-Z])/g, "$1 $2")
389:                     .replace(/[A-Z]([A-Z][a-z])/g, "$1 $2")
390:                     .replace(/&/g, " & ")
391:                     .replace(/\\s+/g, " ")
392:                     .trim()}
393:             </h3>
394:             <p className="text-sm text-gray-600">
395:                 {Object.keys(fieldMap).length} conditions managed
396:             </p>
397:         </div>
398:     </div>
399:
400:     <div className="flex items-center gap-3">
401:         { /* Image Selection */ }
402:         <div className="min-w-[200px]">
403:             <Select
404:                 value={lifestyleCategoryImages[categoryId] || ""}
405:                 onChange={(val) =>
406:                     updateCategoryImage(categoryId, val)
407:                 }
408:             >
409:                 <SelectTrigger>
410:                     <SelectValue placeholder="Select image" />
411:                 </SelectTrigger>
412:                 <SelectContent>
413:                     {availableImages.map((img) => (
414:                         <SelectItem value={img.url} key={img.url}>
415:                             <div className="flex items-center gap-2">
416:                                 <img
417:                                     src={img.url}
418:                                     alt={img.label}
419:                                     className="w-6 h-6 rounded object-cover"
420:                                 />
421:                                 <span>{img.label}</span>
422:                             </div>
423:                             </SelectItem>
424:                         )))
425:                     </SelectContent>
426:                 </Select>
427:             </div>
428:
429:             <Button
430:                 variant="ghost"
431:                 size="icon"
432:                 onClick={() => handleDeleteCategory(categoryId)}
433:                 className="text-red-600 hover:bg-red-50"
434:             >
435:                 <Trash2 className="w-5 h-5" />
436:             </Button>
437:         </div>
438:     </div>
439: </CardHeader>
440:
441: <CardContent className="p-6 space-y-6">
442:     { /* Add New Condition */ }
443:     <div className="bg-gradient-to-r from-blue-50 to-indigo-50 rounded-lg p-4">
444:         <div className="flex gap-3">
445:             <Input
446:                 placeholder="Enter new condition name..."
447:                 value={newField[categoryId] ?? ""}
448:                 onChange={(e) =>
449:                     setNewField((prev) => ({
450:                         ...prev,
451:                         [categoryId]: e.target.value,
452:                     }))
453:                 }
454:                 className="flex-1 border-2 border-blue-200 focus:border-blue-400"
455:             />
456:             <Button
457:                 onClick={() => {

```

```

458:         const fieldName = newField[categoryId]?.trim();
459:         if (fieldName) {
460:             addField(categoryId, fieldName, "strength");
461:             setNewField((prev) => ({
462:                 ...prev,
463:                 [categoryId]: "",
464:             }));
465:         }
466:     }}
467:     className="bg-blue-600 hover:bg-blue-700"
468: >
469:     <Plus className="w-4 h-4 mr-2" />
470:     Add Condition
471: </Button>
472: </div>
473: </div>
474:
475: { /* Conditions List */}
476: <div className="space-y-6">
477:     {Object.entries(fieldMap).map(([field, fieldData]) => (
478:         <div
479:             key={field}
480:             className="bg-white border-2 border-gray-100 rounded-lg p-4 sm:p-6"
481:         >
482:             { /* Header */}
483:             <div className="flex flex-col sm:flex-row sm:items-center justify-between gap-3 ml">
484:                 <h4 className="font-semibold text-gray-800 text-base sm:text-lg">
485:                     {field}
486:                 </h4>
487:                 <div className="flex flex-wrap items-center gap-3">
488:                     { /* Radio group */}
489:                     <div className="flex flex-wrap gap-3">
490:                         <label className="flex items-center gap-2 cursor-pointer">
491:                             <input
492:                                 type="radio"
493:                                 name={` ${categoryId} - ${field} `}
494:                                 checked={fieldData.status === "strength"}
495:                                 onChange={() =>
496:                                     updateFieldStatus(categoryId, field, {
497:                                         status: "strength",
498:                                     })
499:                                 }
500:                                 className="w-4 h-4 text-green-600"
501:                             />
502:                             <span className="text-green-700 font-medium text-sm sm:text-base">
503:                                 Strength
504:                             </span>
505:                         </label>
506:                         <label className="flex items-center gap-2 cursor-pointer">
507:                             <input
508:                                 type="radio"
509:                                 name={` ${categoryId} - ${field} `}
510:                                 checked={fieldData.status === "improvement"}
511:                                 onChange={() =>
512:                                     updateFieldStatus(categoryId, field, {
513:                                         status: "improvement",
514:                                     })
515:                                 }
516:                                 className="w-4 h-4 text-red-600"
517:                             />
518:                             <span className="text-red-700 font-medium text-sm sm:text-base">
519:                                 Needs Improvement
520:                             </span>
521:                         </label>
522:                     </div>
523:                     { /* Delete Button */}
524:                     <Button
525:                         variant="ghost"
526:                         size="icon"
527:                         onClick={() =>
528:                             handleDeleteField(categoryId, field)
529:                         }
530:                         className="text-red-600 hover:bg-red-50"

```

```

531:         >
532:         <Trash2 className="w-4 h-4" />
533:     </Button>
534: </div>
535: </div>
536:
537: { /* Description */}
538: <div className="space-y-2 mb-4">
539:     <Label className="text-sm font-semibold text-gray-700">
540:         Description
541:     </Label>
542:     <Textarea
543:         rows={4}
544:         placeholder="Describe this condition..."
545:         value={fieldData.description || ""}
546:         onChange={(e) =>
547:             updateFieldStatus(categoryId, field, {
548:                 description: e.target.value,
549:             })
550:         }
551:         className="border-2 border-gray-200 focus:border-blue-500 resize-y text-sm"
552:     />
553: </div>
554:
555: { /* Sensitivity */}
556: <div className="space-y-2 mb-6">
557:     <Label className="text-sm font-semibold text-gray-700">
558:         Sensitivity Level
559:     </Label>
560:     <div className="grid grid-cols-3 gap-3 sm:flex sm:gap-6">
561:         {[["low", "medium", "high"] as const].map(
562:             (level) => (
563:                 <div
564:                     key={level}
565:                     className={`cursor-pointer border-2 p-3 rounded-lg transition-all flex :
566:                         fieldData.sensitivity === level
567:                         ? "border-blue-500 bg-blue-50"
568:                         : "border-gray-200 hover:border-gray-300"
569:                     }`
570:                     onClick={() =>
571:                         updateFieldStatus(categoryId, field, {
572:                             sensitivity: level,
573:                         })
574:                     }
575:                 >
576:                     <img
577:                         src={
578:                             level === "low"
579:                                 ? "/lifestylesensetivelevel/1.png"
580:                                 : level === "medium"
581:                                 ? "/lifestylesensetivelevel/2.png"
582:                                 : "/lifestylesensetivelevel/3.png"
583:                         }
584:                         alt={` ${level} sensitivity`}
585:                         className="w-14 h-14 sm:w-16 sm:h-16 object-contain"
586:                     />
587:                     <p className="text-xs sm:text-sm mt-1 capitalize">
588:                         {level}
589:                     </p>
590:                 </div>
591:             )
592:         )}
593:     </div>
594: </div>
595:
596: { /* Recommendations */}
597: <div className="grid grid-cols-1 md:grid-cols-2 xl:grid-cols-4 gap-4">
598:     {
599:         [
600:             "avoid", "follow", "consume", "monitor"
601:         ].map((key) => {
602:             const labelKey =
603:                 `${key}Label` as EditableLabelKey;
604:             const colors = {

```

```

604:         avoid: "red",
605:         follow: "green",
606:         consume: "blue",
607:         monitor: "yellow",
608:     };
609:
610:     return (
611:         <div key={key} className="space-y-2">
612:             <div className="flex items-center gap-2">
613:                 {editingLabels?.[categoryId]?.[field]?.[
614:                     labelKey
615:                 ] ? (
616:                     <>
617:                         <Input
618:                             className="flex-1 text-sm"
619:                             value={fieldData[labelKey] || ""}
620:                             onChange={(e) =>
621:                                 updateFieldStatus(
622:                                     categoryId,
623:                                     field,
624:                                     {
625:                                         [labelKey]: e.target.value,
626:                                     }
627:                                 )
628:                             }
629:                             placeholder={`Label for ${key}`}
630:                         />
631:                         <Button
632:                             variant="ghost"
633:                             size="icon"
634:                             onClick={() =>
635:                                 toggleEditLabel(
636:                                     categoryId,
637:                                     field,
638:                                     labelKey
639:                                 )
640:                             }
641:                             className="h-8 w-8"
642:                         >
643:                             <Check className="w-3 h-3" />
644:                         </Button>
645:                     </>
646:                 ) : (
647:                     <>
648:                         <Label
649:                             className={`flex-1 font-semibold text-${colors[key]}-700 text-xs`}
650:                         >
651:                             {fieldData[labelKey] ||
652:                                 key.toUpperCase()}
653:                         </Label>
654:                         <Button
655:                             variant="ghost"
656:                             size="icon"
657:                             onClick={() =>
658:                                 toggleEditLabel(
659:                                     categoryId,
660:                                     field,
661:                                     labelKey
662:                                 )
663:                             }
664:                             className="h-8 w-8"
665:                         >
666:                             <Edit3 className="w-3 h-3" />
667:                         </Button>
668:                     </>
669:                 )}
670:             </div>
671:             <Textarea
672:                 rows={4}
673:                 placeholder={`Enter ${key} recommendations (one per line)...`}
674:                 defaultValue={{
675:                     (fieldData[key] as string[]) || []
676:                 }.join("\n")}

```

```

677:                                     onBlur={(e) => {
678:                                         const items = e.target.value
679:                                             .split("\n")
680:                                             .map((v) => v.trim())
681:                                             .filter(Boolean);
682:                                         updateFieldStatus(categoryId, field, {
683:                                             [key]: items,
684:                                         });
685:                                     }}
686:                                     onKeyDown={(e) => e.stopPropagation()}
687:                                     className="text-sm resize-y min-h-[80px]"
688:                                 />
689:                             </div>
690:                         );
691:                     }}}
692:                 </div>
693:             </div>
694:         )}}
695:     </div>
696: </CardContent>
697: </Card>
698: );
699: }}}
700: </div>
701: </div>
702:
703: { /* Delete Confirmation Modal */ }
704: {deleteConfirmation.type && (
705:     <div className="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center z-50 p-4">
706:         <div className="bg-white rounded-2xl p-6 max-w-md w-full shadow-2xl">
707:             <div className="flex items-center gap-4 mb-4">
708:                 <div className="w-12 h-12 bg-red-100 rounded-full flex items-center justify-center">
709:                     <AlertTriangle className="w-6 h-6 text-red-600" />
710:                 </div>
711:                 <div>
712:                     <h3 className="text-lg font-bold text-gray-900">
713:                         Confirm Deletion
714:                     </h3>
715:                     <p className="text-sm text-gray-600">
716:                         This action cannot be undone
717:                     </p>
718:                 </div>
719:             </div>
720:
721:             <div className="mb-6">
722:                 {deleteConfirmation.type === "category" ? (
723:                     <p className="text-gray-700">
724:                         Are you sure you want to delete the{" "}
725:                         <strong>{deleteConfirmation.categoryName}</strong> category?
726:                         This will permanently remove all conditions and data.
727:                     </p>
728:                 ) : (
729:                     <p className="text-gray-700">
730:                         Are you sure you want to delete{" "}
731:                         <strong>{deleteConfirmation.fieldName}</strong>
732:                         from <strong>{deleteConfirmation.categoryName}</strong>? All
733:                         recommendations will be permanently removed.
734:                     </p>
735:                 )}
736:             </div>
737:
738:             <div className="flex justify-end gap-3">
739:                 <Button variant="outline" onClick={cancelDelete}>
740:                     Cancel
741:                 </Button>
742:                 <Button
743:                     onClick={confirmDelete}
744:                     className="bg-red-600 hover:bg-red-700 text-white"
745:                 >
746:                     <Trash2 className="w-4 h-4 mr-2" />
747:                     Delete
748:                 </Button>
749:             </div>

```



```

750:         </div>
751:     </div>
752:   )}
753: </div>
754: );
755: }

```

## File: components\admin\metabolic-core-admin.tsx

```

=====
1: "use client";
2:
3: import {
4:   Card,
5:   CardContent,
6:   CardDescription,
7:   CardHeader,
8:   CardTitle,
9: } from "@components/ui/card";
10: import { Input } from "@components/ui/input";
11: import { Label } from "@components/ui/label";
12: import { Textarea } from "@components/ui/textarea";
13: import { Button } from "@components/ui/button";
14: import { Badge } from "@components/ui/badge";
15: import {
16:   Select,
17:   SelectTrigger,
18:   SelectValue,
19:   SelectContent,
20:   SelectItem,
21: } from "@components/ui/select";
22: import {
23:   Save,
24:   RotateCcw,
25:   Plus,
26:   Trash2,
27:   AlertCircle,
28:   CheckCircle,
29:   Minus,
30: } from "lucide-react";
31: import { useState } from "react";
32: import type { MetabolicCore, MetabolicGeneEntry } from "@types/report-types";
33:
34: interface MetabolicCoreAdminProps {
35:   metabolicCore: MetabolicCore;
36:   setMetabolicCore: React.Dispatch<React.SetStateAction<MetabolicCore>>;
37:   updateMetabolicCore: (field: string, data: any) => void;
38:   onSave: () => void;
39:   onReset: () => void;
40: }
41:
42: export default function MetabolicCoreAdmin({
43:   metabolicCore,
44:   setMetabolicCore,
45:   updateMetabolicCore,
46:   onSave,
47:   onReset,
48: }: MetabolicCoreAdminProps) {
49:   const metabolicAreas = Object.keys(metabolicCore).filter(
50:     (key) => key !== "quote" && key !== "description"
51:   );
52:
53:   const [showAddModal, setShowAddModal] = useState(false);
54:   const [newAreaName, setNewAreaName] = useState("");
55:   const [showDeleteModal, setShowDeleteModal] = useState(false);
56:   const [deleteTarget, setDeleteTarget] = useState<{
57:     area: string;
58:     geneIndex?: number;
59:   } | null>(null);
60:
61:   const handleConfirmAddArea = () => {
62:     const trimmedName = newAreaName.trim();

```

```

63:     if (!trimmedName || metabolicCore[trimmedName]) {
64:         alert("Invalid or duplicate name.");
65:         return;
66:     }
67:
68:     const { quote, description, ...restAreas } = metabolicCore;
69:
70:     // New object with quote & description first, new area next, rest after
71:     const reorderedCore: MetabolicCore = {
72:         quote: quote || "",
73:         description: description || "",
74:         [trimmedName]: { impact: "", advice: "", genes: [] },
75:         ...restAreas,
76:     };
77:
78:     setMetabolicCore(reorderedCore);
79:     setNewAreaName("");
80:     setShowAddModal(false);
81: };
82:
83: const deleteArea = (area: string) => {
84:     setDeleteTarget({ area });
85:     setShowDeleteModal(true);
86: };
87:
88: const addGene = (area: string) => {
89:     const currentGenes = metabolicCore[area]?.genes ?? [];
90:     updateMetabolicCore(area, {
91:         genes: [...currentGenes, { name: "", genotype: "", impact: "" }],
92:     });
93: };
94:
95: const removeGene = (area: string, index: number) => {
96:     const updatedGenes = [...metabolicCore[area].genes];
97:     updatedGenes.splice(index, 1);
98:     updateMetabolicCore(area, { genes: updatedGenes });
99: };
100:
101: const updateGene = (
102:     area: string,
103:     index: number,
104:     updatedGene: MetabolicGeneEntry
105: ) => {
106:     const genes = [...metabolicCore[area].genes];
107:     genes[index] = updatedGene;
108:     updateMetabolicCore(area, { genes });
109: };
110:
111: const updateAreaAdvice = (area: string, advice: string) => {
112:     updateMetabolicCore(area, { advice });
113: };
114:
115: return (
116:     <div className="min-h-screen bg-gradient-to-br from-slate-50 to-blue-50/30 ">
117:         <div className="max-w-8xl mx-auto">
118:             <Card className="shadow-xl border-0 bg-white/95 backdrop-blur-sm">
119:                 <CardHeader className="bg-gradient-to-r from-indigo-600 via-purple-600 to-blue-600 text-white rounded">
120:                     <CardTitle className="text-base sm:text-xl md:text-2xl lg:text-3xl font-bold uppercase flex items-center">
121:                         Metabolic Core Analysis
122:                     </CardTitle>
123:                 </CardHeader>
124:
125:                 <div className="text-right p-3 sm:p-4">
126:                     <Button onClick={onSave} size="sm" className="sm:w-auto">
127:                         <Save className="h-4 w-4 mr-2" /> Save Changes
128:                     </Button>
129:                 </div>
130:
131:                 <CardContent className="p-3 sm:p-4 lg:p-6 space-y-6">
132:                     { /* General Information Section */ }
133:                     <div className="space-y-4">
134:                         <div className="border-b border-slate-200 pb-4">
135:                             <h3 className="text-base sm:text-lg font-semibold text-slate-800 mb-4">

```

```

136:         General Information
137:     </h3>
138:     <div className="grid gap-4">
139:         <div className="space-y-2">
140:             <Label className="text-sm font-medium text-slate-700">
141:                 Quote
142:             </Label>
143:             <Input
144:                 value={metabolicCore.quote || ""}
145:                 onChange={(e) =>
146:                     updateMetabolicCore("quote", e.target.value)
147:                 }
148:                 placeholder="Enter a brief summary quote about the metabolic analysis..."
149:                 className="border-slate-200 focus:border-indigo-500 focus:ring-indigo-500 text-sm"
150:             />
151:         </div>
152:         <div className="space-y-2">
153:             <Label className="text-sm font-medium text-slate-700">
154:                 Overall Description
155:             </Label>
156:             <Textarea
157:                 value={metabolicCore.description || ""}
158:                 onChange={(e) =>
159:                     updateMetabolicCore("description", e.target.value)
160:                 }
161:                 placeholder="Provide an overall description of the metabolic analysis findings..."
162:                 className="border-slate-200 focus:border-indigo-500 focus:ring-indigo-500 min-h-[80px"
163:             />
164:         </div>
165:     </div>
166: </div>
167: </div>
168:
169: { /* Metabolic Areas Section */}
170: <div className="space-y-4">
171:     <div className="flex flex-col sm:flex-row sm:items-center sm:justify-between gap-3 border-b b"
172:     <h3 className="text-base sm:text-lg font-semibold text-slate-800">
173:         Metabolic Areas ({metabolicAreas.length})
174:     </h3>
175:     <Button
176:         onClick={() => setShowAddModal(true)}
177:         size="sm"
178:         className="bg-indigo-600 hover:bg-indigo-700 w-full sm:w-auto"
179:     >
180:         <Plus className="h-4 w-4 mr-2" /> Add New Area
181:     </Button>
182: </div>
183:
184: <div className="grid gap-4">
185:     {metabolicAreas.map((area) => {
186:         const section = metabolicCore[area];
187:         return (
188:             <Card
189:                 key={area}
190:                 className="border border-slate-200 bg-slate-50/50 hover:shadow-md transition-shadow d"
191:             >
192:                 <CardHeader className="p-3 sm:p-4 pb-2 sm:pb-3">
193:                     <div className="flex flex-col sm:flex-row sm:items-start sm:justify-between gap-3">
194:                         <CardTitle className="text-base sm:text-lg lg:text-xl text-slate-800 capitalize">
195:                             {area
196:                                 .replace(/([a-z])([A-Z])/g, "$1 $2")
197:                                 .replace(/[A-Z]([A-Z][a-z])/g, "$1 $2")
198:                                 .replace(/&/g, " & ")
199:                                 .replace(/\\s+/g, " ")
200:                                 .trim()}
201:                             </CardTitle>
202:                             <Button
203:                                 onClick={() => {
204:                                     setDeleteTarget({ area });
205:                                     setShowDeleteModal(true);
206:                                 }}
207:                                 size="sm"
208:                                 variant="outline"

```

```

209:         className="text-red-600 border-red-200 hover:bg-red-50 hover:border-red-300 w-f
210:     >
211:         <Trash2 className="h-4 w-4 mr-2 sm:mr-0" />
212:         <span className="sm:hidden">Delete Area</span>
213:     </Button>
214: </div>
215: </CardHeader>
216:
217: <CardContent className="p-3 sm:p-4 pt-0 space-y-4">
218:     { /* Area Advice */ }
219:     <div className="space-y-2">
220:         <Label className="text-sm font-medium text-slate-700">
221:             Area Advice
222:         </Label>
223:         <Textarea
224:             value={section.advice || ""}
225:             onChange={(e) =>
226:                 updateAreaAdvice(area, e.target.value)
227:             }
228:             rows={5}
229:             placeholder={`Enter advice for ${area} area...`}
230:             className="border-slate-200 focus:border-indigo-500 focus:ring-indigo-500 min-h
231:         />
232:     </div>
233:
234:     <div className="flex flex-col sm:flex-row sm:items-center sm:justify-between gap-3">
235:         <h4 className="text-sm font-medium text-slate-700">
236:             Genes ({section.genes.length})
237:         </h4>
238:         <Button
239:             onClick={() => addGene(area)}
240:             size="sm"
241:             variant="outline"
242:             className="text-indigo-600 border-indigo-200 hover:bg-indigo-50 w-full sm:w-aut
243:         >
244:             <Plus className="w-4 h-4 mr-1" /> Add Gene
245:         </Button>
246:     </div>
247:
248:     {section.genes.length === 0 ? (
249:         <div className="text-center py-6 text-slate-500 text-sm">
250:             No genes added yet. Click "Add Gene" to get started.
251:         </div>
252:     ) : (
253:         <div className="space-y-3">
254:             { /* Mobile Layout - Stacked Cards */ }
255:             <div className="block sm:hidden space-y-3">
256:                 {section.genes.map(
257:                     (gene: MetabolicGeneEntry, index: number) => (
258:                         <div
259:                             key={index}
260:                             className="bg-white p-3 rounded-lg border border-slate-200 space-y-3"
261:                         >
262:                             <div className="space-y-2">
263:                                 <Label className="text-xs font-medium text-slate-600">
264:                                     GENE (ALLELES)
265:                                 </Label>
266:                                 <Input
267:                                     placeholder="e.g., MTHFR (A>C)"
268:                                     value={gene.name}
269:                                     onChange={(e) =>
270:                                         updateGene(area, index, {
271:                                             ...gene,
272:                                             name: e.target.value,
273:                                         })
274:                                     }
275:                                 </div>
276:                                 <div className="text-sm" />
277:                             </div>
278:
279:                             <div className="grid grid-cols-2 gap-3">
280:                                 <div className="space-y-2">
281:                                     <Label className="text-xs font-medium text-slate-600">

```

```

282:         YOUR GENOTYPE
283:     </Label>
284:     <Input
285:         placeholder="TT"
286:         value={gene.genotype}
287:         onChange={(e) =>
288:             updateGene(area, index, {
289:                 ...gene,
290:                 genotype: e.target.value,
291:             })
292:         }
293:         className="text-center text-sm font-mono"
294:     />
295: </div>
296:
297: <div className="space-y-2">
298:     <Label className="text-xs font-medium text-slate-600">
299:         GENETIC IMPACT
300:     </Label>
301:     <div className="flex items-center gap-2">
302:         <div
303:             className={`w-6 h-6 rounded-full flex-shrink-0 ${
304:                 gene.impact
305:                     ?.toLowerCase()
306:                     .includes("high") ||
307:                 gene.impact
308:                     ?.toLowerCase()
309:                     .includes("increased")
310:                 ? "bg-red-500"
311:                 : gene.impact
312:                     ?.toLowerCase()
313:                     .includes("normal") ||
314:                 gene.impact
315:                     ?.toLowerCase()
316:                     .includes("average")
317:                 ? "bg-green-500"
318:                 : gene.impact
319:                     ?.toLowerCase()
320:                     .includes("moderate")
321:                 ? "bg-yellow-500"
322:                 : "bg-slate-300"
323:             }`}
324:         ></div>
325:         <Select
326:             value={gene.impact}
327:             onValueChange={(val) =>
328:                 updateGene(area, index, {
329:                     ...gene,
330:                     impact: val,
331:                 })
332:             }
333:         >
334:             <SelectTrigger className="flex-1 text-sm">
335:                 <SelectValue placeholder="Select Impact" />
336:             </SelectTrigger>
337:             <SelectContent>
338:                 <SelectItem value="High">
339:                     High
340:                 </SelectItem>
341:                 <SelectItem value="Moderate">
342:                     Moderate
343:                 </SelectItem>
344:                 <SelectItem value="Normal">
345:                     Normal
346:                 </SelectItem>
347:                 <SelectItem value="Low">
348:                     Low
349:                 </SelectItem>
350:             </SelectContent>
351:         </Select>
352:     </div>
353: </div>
354: </div>

```

```

355:
356:         <Button
357:             onClick={() => {
358:                 setDeleteTarget({
359:                     area,
360:                     geneIndex: index,
361:                 });
362:                 setShowDeleteModal(true);
363:             }}
364:             variant="outline"
365:             size="sm"
366:             className="w-full text-red-600 border-red-200 hover:bg-red-50"
367:         >
368:             <Trash2 className="w-4 h-4 mr-2" />
369:             Delete Gene
370:         </Button>
371:     </div>
372: )
373: }}
374: </div>
375:
376: { /* Desktop Layout - Table */}
377: <div className="hidden sm:block overflow-x-auto">
378:     <div className="min-w-full bg-white rounded-lg border border-slate-200">
379:         { /* Table Header */}
380:         <div className="bg-blue-500 text-white grid grid-cols-8 gap-4 p-4 rounded-t">
381:             <div className="col-span-3 text-center">
382:                 GENE (ALLELES)
383:             </div>
384:             <div className="col-span-2 text-center">
385:                 YOUR GENOTYPE
386:             </div>
387:             <div className="col-span-2 text-center">
388:                 GENETIC IMPACT
389:             </div>
390:             <div className="col-span-1 text-center">
391:                 ACTION
392:             </div>
393:         </div>
394:
395:         { /* Table Rows */}
396:         {section.genes.map(
397:             (gene: MetabolicGeneEntry, index: number) => (
398:                 <div
399:                     key={index}
400:                     className="grid grid-cols-8 gap-4 p-4 border-b border-slate-200 items
401:                 >
402:                     <div className="col-span-3">
403:                         <Input
404:                             placeholder="e.g., MTHFR (A>C)"
405:                             value={gene.name}
406:                             onChange={(e) =>
407:                                 updateGene(area, index, {
408:                                     ...gene,
409:                                     name: e.target.value,
410:                                 })
411:                             }
412:                             className="border-slate-200 focus:border-indigo-500 text-center t
413:                         />
414:                     </div>
415:                     <div className="col-span-2">
416:                         <Input
417:                             placeholder="TT"
418:                             value={gene.genotype}
419:                             onChange={(e) =>
420:                                 updateGene(area, index, {
421:                                     ...gene,
422:                                     genotype: e.target.value,
423:                                 })
424:                             }
425:                             className="border-slate-200 focus:border-indigo-500 text-center t
426:                         />
427:                     </div>

```

```

428: <div className="col-span-2 flex justify-center">
429:   <div className="flex items-center gap-2">
430:     <div
431:       className={`w-7 h-7 rounded-full ${
432:         gene.impact
433:           ?.toLowerCase()
434:             .includes("high") ||
435:         gene.impact
436:           ?.toLowerCase()
437:             .includes("increased")
438:           ? "bg-red-500"
439:           : gene.impact
440:             ?.toLowerCase()
441:               .includes("normal") ||
442:         gene.impact
443:           ?.toLowerCase()
444:             .includes("average")
445:           ? "bg-green-500"
446:           : gene.impact
447:             ?.toLowerCase()
448:               .includes("moderate")
449:           ? "bg-yellow-500"
450:           : "bg-slate-300"
451:         }`}
452:     ></div>
453:     <Select
454:       value={gene.impact}
455:       onChange={(val) =>
456:         updateGene(area, index, {
457:           ...gene,
458:           impact: val,
459:         })
460:     }
461:   >
462:     <SelectTrigger className="w-[160px] border border-slate-200 bg-
463:       <SelectValue placeholder="Select Impact" />
464:     </SelectTrigger>
465:     <SelectContent>
466:       <SelectItem value="High">
467:         High
468:       </SelectItem>
469:       <SelectItem value="Moderate">
470:         Moderate
471:       </SelectItem>
472:       <SelectItem value="Normal">
473:         Normal
474:       </SelectItem>
475:       <SelectItem value="Low">
476:         Low
477:       </SelectItem>
478:     </SelectContent>
479:   </Select>
480: </div>
481: </div>
482:
483: <div className="col-span-1 flex justify-center">
484:   <Button
485:     onClick={() => {
486:       setDeleteTarget({
487:         area,
488:         geneIndex: index,
489:       });
490:       setShowDeleteModal(true);
491:     }}
492:     variant="outline"
493:     size="sm"
494:     className="text-red-600 border-red-200 hover:bg-red-50"
495:   >
496:     <Trash2 className="w-4 h-4" />
497:   </Button>
498: </div>
499: </div>
500: )

```

```

501:         }}
502:     </div>
503: </div>
504: </div>
505:     }}
506: </CardContent>
507: </Card>
508: );
509: }}}
510:
511: {metabolicAreas.length === 0 && (
512:     <div className="text-center py-8 sm:py-12 text-slate-500">
513:         <div className="mb-4">
514:             <Plus className="h-8 w-8 sm:h-12 sm:w-12 mx-auto text-slate-300" />
515:         </div>
516:         <p className="text-base sm:text-lg mb-2">
517:             No metabolic areas defined
518:         </p>
519:         <p className="text-xs sm:text-sm">
520:             Click "Add New Area" to create your first metabolic
521:             analysis section.
522:         </p>
523:     </div>
524:     )}
525: </div>
526: </div>
527: </CardContent>
528: </Card>
529: </div>
530:
531: {showAddModal && (
532:     <div className="fixed inset-0 z-50 flex items-center justify-center bg-black/40 p-4">
533:         <div className="bg-white rounded-lg shadow-lg p-4 sm:p-6 w-full max-w-sm space-y-4 border border-
534:             <h3 className="text-base sm:text-lg font-semibold text-slate-800">
535:                 Add New Metabolic Area
536:             </h3>
537:             <Input
538:                 value={newAreaName}
539:                 onChange={(e) => setNewAreaName(e.target.value)}
540:                 placeholder="e.g., Detox Pathways"
541:                 className="border-slate-200 focus:ring-indigo-500 text-sm"
542:             />
543:             <div className="flex flex-col sm:flex-row justify-end gap-2 pt-2">
544:                 <Button
545:                     variant="outline"
546:                     onClick={() => {
547:                         setNewAreaName("");
548:                         setShowAddModal(false);
549:                     }}
550:                     className="w-full sm:w-auto order-2 sm:order-1"
551:                 >
552:                     Cancel
553:                 </Button>
554:                 <Button
555:                     onClick={handleConfirmAddArea}
556:                     className="bg-indigo-600 text-white hover:bg-indigo-700 w-full sm:w-auto order-1 sm:order-2"
557:                 >
558:                     Add
559:                 </Button>
560:             </div>
561:         </div>
562:     </div>
563: )}
564:
565: {showDeleteModal && deleteTarget && (
566:     <div className="fixed inset-0 z-50 flex items-center justify-center bg-black/40 p-4">
567:         <div className="bg-white rounded-lg shadow-xl p-4 sm:p-6 w-full max-w-sm space-y-4 border border-
568:             <h3 className="text-base sm:text-lg font-semibold text-slate-800">
569:                 Confirm Deletion
570:             </h3>
571:             <p className="text-sm sm:text-base text-slate-600">
572:                 Are you sure you want to delete{" "}
573:                 {deleteTarget.geneIndex !== undefined

```



```

574:         ? `this gene from ${deleteTarget.area}?`
575:         : `the "${deleteTarget.area}" area?`
576:     </p>
577:     <div className="flex flex-col sm:flex-row justify-end gap-2 pt-2">
578:         <Button
579:             variant="outline"
580:             onClick={() => {
581:                 setShowDeleteModal(false);
582:                 setDeleteTarget(null);
583:             }}
584:             className="w-full sm:w-auto order-2 sm:order-1"
585:         >
586:             Cancel
587:         </Button>
588:         <Button
589:             onClick={() => {
590:                 if (deleteTarget.geneIndex !== undefined) {
591:                     removeGene(deleteTarget.area, deleteTarget.geneIndex);
592:                 } else {
593:                     const updated = { ...metabolicCore };
594:                     delete updated[deleteTarget.area];
595:                     setMetabolicCore(updated);
596:                 }
597:                 setShowDeleteModal(false);
598:                 setDeleteTarget(null);
599:             }}
600:             className="bg-red-600 text-white hover:bg-red-700 w-full sm:w-auto order-1 sm:order-2"
601:         >
602:             Confirm Delete
603:         </Button>
604:     </div>
605: </div>
606: </div>
607:     )}
608: </div>
609: );
610: }

```

## File: components\admin\nutrition-admin.tsx

```

=====
1: "use client";
2:
3: import { Card, CardContent, CardHeader, CardTitle } from "@/components/ui/card";
4: import { Input } from "@/components/ui/input";
5: import { Label } from "@/components/ui/label";
6: import {
7:   Select,
8:   SelectContent,
9:   SelectItem,
10:  SelectTrigger,
11:  SelectValue,
12: } from "@/components/ui/select";
13: import { Button } from "@/components/ui/button";
14: import { Badge } from "@/components/ui/badge";
15: import {
16:   Dialog,
17:   DialogContent,
18:   DialogDescription,
19:   DialogFooter,
20:   DialogHeader,
21:   DialogTitle,
22:   DialogTrigger,
23: } from "@/components/ui/dialog";
24: import { Save, Plus, Trash2, Edit3 } from "lucide-react";
25: import { useState, useEffect } from "react";
26: import { useToast } from "@/components/ui/use-toast";
27:
28: interface NutrientData {
29:   score: number;
30:   healthImpact: string;
31:   intakeLevel: string;

```

```

32:   source: string;
33: }
34:
35: interface FullNutritionData {
36:   quote?: string;
37:   description?: string;
38:   data: Record<string, Record<string, NutrientData>>;
39: }
40:
41: interface NutritionAdminProps {
42:   nutritionData?: FullNutritionData;
43:   updateNutritionData?: (
44:     section: string,
45:     field: string,
46:     data: Partial<NutrientData> | Record<string, NutrientData> | string
47:   ) => void;
48:   onSave?: () => void;
49:   onUpdateHeader?: (quote: string, description: string) => void;
50: }
51:
52: export default function NutritionAdmin({
53:   nutritionData: propNutritionData,
54:   updateNutritionData: propUpdateNutritionData,
55:   onSave: propOnSave,
56: }: NutritionAdminProps) {
57:   // Default data structure
58:   const defaultNutrientData: NutrientData = {
59:     score: 5,
60:     healthImpact: "",
61:     intakeLevel: "NORMAL INTAKE",
62:     source: "DIET",
63:   };
64:
65:   // State management
66:   const [isAddSectionDialogOpen, setIsAddSectionDialogOpen] = useState(false);
67:   const [newSectionName, setNewSectionName] = useState("");
68:   const [isAddDialogOpen, setIsAddDialogOpen] = useState(false);
69:   const [newCategoryName, setNewCategoryName] = useState("");
70:   const [isEditMode, setIsEditMode] = useState<string | null>(null);
71:   const [editingName, setEditingName] = useState("");
72:   const [confirmDelete, setConfirmDelete] = useState<null | string>(null);
73:   const { toast } = useToast();
74:   const [quote, setQuote] = useState(propNutritionData?.quote || "");
75:   const [description, setDescription] = useState(
76:     propNutritionData?.description || ""
77:   );
78:   const [localNutritionData, setLocalNutritionData] =
79:     useState<FullNutritionData>({
80:       data: propNutritionData?.data || {},
81:     });
82:
83:   const [activeSection, setActiveSection] = useState<string>("");
84:
85:   // New nutrient form state
86:   const [newNutrient, setNewNutrient] = useState({
87:     name: "",
88:     score: 5,
89:     healthImpact: "",
90:     intakeLevel: "NORMAL INTAKE",
91:     source: "DIET",
92:   });
93:
94:   // ? FIXED: Proper delete field function
95:   const handleDeleteField = async (sectionName: string, fieldName: string) => {
96:     try {
97:       // 1. Update local state first
98:       const updatedSection = { ...localNutritionData.data[sectionName] };
99:       delete updatedSection[fieldName];
100:
101:       setLocalNutritionData((prev) => ({
102:         ...prev,
103:         data: {
104:           ...prev.data,

```

```

105:         [sectionName]: updatedSection,
106:     },
107: });
108:
109: // 2. Update parent state
110: if (propUpdateNutritionData) {
111:     propUpdateNutritionData(sectionName, "", updatedSection);
112: }
113:
114: toast({
115:     title: "Field Deleted",
116:     description: `${fieldName} removed from ${sectionName}.`,
117:     variant: "destructive",
118:     duration: 3000,
119: });
120: } catch (error) {
121:     console.error("Delete field error:", error);
122:     toast({
123:         title: "Delete Failed",
124:         description: "Failed to delete field",
125:         variant: "destructive",
126:     });
127: }
128: };
129:
130: // ? FIXED: Proper delete section function
131: const handleDeleteSection = (sectionName: string) => {
132:     try {
133:         // 1. Update local state
134:         const updatedData = { ...localNutritionData.data };
135:         delete updatedData[sectionName];
136:
137:         setLocalNutritionData((prev) => ({
138:             ...prev,
139:             data: updatedData,
140:         }));
141:
142:         // 2. Update parent state with deletion signal
143:         if (propUpdateNutritionData) {
144:             propUpdateNutritionData(sectionName, "", "__delete__");
145:         }
146:
147:         // 3. Reset active section if needed
148:         if (activeSection === sectionName) {
149:             const remaining = Object.keys(updatedData);
150:             setActiveSection(remaining[0] || "");
151:         }
152:
153:         // 4. Call parent save
154:         if (propOnSave) {
155:             propOnSave();
156:         }
157:
158:         toast({
159:             title: "Deleted",
160:             description: `Deleted section "${sectionName}" successfully.`,
161:             variant: "destructive",
162:         });
163:     } catch (error) {
164:         console.error("Delete section error:", error);
165:         toast({
166:             title: "Delete Failed",
167:             description: "Failed to delete section",
168:             variant: "destructive",
169:         });
170:     }
171: };
172:
173: // ? FIXED: Proper rename function
174: const renameNutrient = (
175:     section: string,
176:     oldField: string,
177:     newName: string

```

```

178:   ) => {
179:     try {
180:       if (!localNutritionData.data || !newName.trim()) {
181:         return false;
182:       }
183:
184:       const newField = newName.toLowerCase().replace(/\s+/g, "");
185:
186:       // Check if name unchanged
187:       if (newField === oldField) {
188:         return true;
189:       }
190:
191:       // Check if new name already exists
192:       if (localNutritionData.data[section]?.[newField]) {
193:         return false;
194:       }
195:
196:       // Create updated section with renamed field
197:       const updatedSection = { ...localNutritionData.data[section] };
198:       updatedSection[newField] = updatedSection[oldField];
199:       delete updatedSection[oldField];
200:
201:       // Update local state
202:       setLocalNutritionData((prev) => ({
203:         ...prev,
204:         data: {
205:           ...prev.data,
206:           [section]: updatedSection,
207:         },
208:       }));
209:
210:       // Update parent state
211:       if (propUpdateNutritionData) {
212:         propUpdateNutritionData(section, "", updatedSection);
213:       }
214:
215:       toast({
216:         title: "Renamed",
217:         description: `Nutrient renamed from "${oldField}" to "${newField}"`,
218:         variant: "success",
219:       });
220:
221:       return true;
222:     } catch (error) {
223:       console.error("Rename error:", error);
224:       toast({
225:         title: "Rename Failed",
226:         description: "Failed to rename nutrient",
227:         variant: "destructive",
228:       });
229:       return false;
230:     }
231:   };
232:
233:   // ? FIXED: Proper save function
234:   const onSave = async () => {
235:     try {
236:       // Sync quote & description into parent
237:       if (propUpdateNutritionData) {
238:         propUpdateNutritionData("quote", "", quote);
239:         propUpdateNutritionData("description", "", description);
240:       }
241:
242:       if (propOnSave) {
243:         propOnSave();
244:       }
245:
246:       toast({
247:         title: "Updated",
248:         description: "Nutrition data has been saved successfully.",
249:         variant: "success",
250:       });

```

```

251:     } catch (err: any) {
252:       console.error("Save error:", err);
253:       toast({
254:         title: "Save Failed",
255:         description: err.message || "An unknown error occurred.",
256:         variant: "destructive",
257:       });
258:     }
259:   };
260:
261:   // ? FIXED: Proper add category function
262:   const addNewCategory = () => {
263:     const key = newSectionName.trim().toLowerCase().replace(/\s+/g, "");
264:
265:     if (!key) {
266:       toast({
267:         title: "Add Failed",
268:         description: "Category name cannot be empty.",
269:         variant: "destructive",
270:       });
271:       return;
272:     }
273:
274:     if (localNutritionData.data[key]) {
275:       toast({
276:         title: "Add Failed",
277:         description: "Category already exists.",
278:         variant: "destructive",
279:       });
280:       return;
281:     }
282:
283:     setLocalNutritionData((prev) => ({
284:       ...prev,
285:       data: {
286:         ...prev.data,
287:         [key]: {},
288:       },
289:     }));
290:
291:     if (propUpdateNutritionData) {
292:       propUpdateNutritionData(key, "", {});
293:     }
294:
295:     setActiveSection(key);
296:     setNewSectionName(""); // reset input
297:     setIsAddSectionDialogOpen(false);
298:
299:     toast({
300:       title: "Category Added",
301:       description: `${key} was created successfully.`,
302:       variant: "success",
303:     });
304:   };
305:
306:   // ? FIXED: Proper add nutrient function
307:   const handleAddNutrient = () => {
308:     if (!activeSection) {
309:       toast({
310:         title: "Add Failed",
311:         description: "Please select a section first.",
312:         variant: "destructive",
313:       });
314:       return;
315:     }
316:
317:     if (addNutrient(activeSection, newNutrient)) {
318:       setNewNutrient({
319:         name: "",
320:         score: 5,
321:         healthImpact: "",
322:         intakeLevel: "NORMAL INTAKE",
323:         source: "DIET",

```

```

324:     });
325:     setIsAddDialogOpen(false);
326:
327:     toast({
328:       title: "Added",
329:       description: "New nutrient added successfully",
330:       variant: "success",
331:       duration: 3000,
332:     });
333:   } else {
334:     toast({
335:       title: "Add Failed",
336:       description: "Invalid name or nutrient already exists!",
337:       variant: "destructive",
338:       duration: 3000,
339:     });
340:   }
341: };
342:
343: // ? FIXED: Proper rename handler
344: const handleRename = (section: string, field: string) => {
345:   if (renameNutrient(section, field, editingName)) {
346:     setIsEditMode(null);
347:     setEditingName("");
348:   } else {
349:     toast({
350:       title: "Rename Failed",
351:       description: "Invalid name or nutrient already exists!",
352:       variant: "destructive",
353:       duration: 3000,
354:     });
355:   }
356: };
357:
358: // ? FIXED: Proper add nutrient function
359: const addNutrient = (section: string, nutrientData: typeof newNutrient) => {
360:   if (!localNutritionData.data || !nutrientData.name.trim()) {
361:     return false;
362:   }
363:
364:   const fieldName = nutrientData.name.toLowerCase().replace(/\s+/g, "");
365:
366:   if (localNutritionData.data[section]?.[fieldName]) {
367:     return false;
368:   }
369:
370:   const newEntry: NutrientData = {
371:     score: nutrientData.score,
372:     healthImpact: nutrientData.healthImpact,
373:     intakeLevel: nutrientData.intakeLevel,
374:     source: nutrientData.source,
375:   };
376:
377:   // Update local state
378:   setLocalNutritionData((prev) => ({
379:     ...prev,
380:     data: {
381:       ...prev.data,
382:       [section]: {
383:         ...prev.data[section],
384:         [fieldName]: newEntry,
385:       },
386:     },
387:   }));
388:
389:   // Update parent state
390:   if (propUpdateNutritionData) {
391:     propUpdateNutritionData(section, fieldName, newEntry);
392:   }
393:
394:   return true;
395: };
396:

```

```

397: // ? FIXED: Proper local update function
398: const updateLocalNutritionData = (
399:   section: string,
400:   field: string,
401:   data: Partial<NutrientData> | Record<string, NutrientData>
402: ) => {
403:   setLocalNutritionData((prev) => {
404:     const updated = { ...prev.data };
405:
406:     if (!updated[section]) {
407:       updated[section] = {};
408:     }
409:
410:     if (field && typeof data === "object" && !Array.isArray(data)) {
411:       // Update specific field
412:       updated[section][field] = {
413:         ...(updated[section][field] || {}),
414:         ...(data as Partial<NutrientData>),
415:       };
416:     } else if (!field && typeof data === "object" && !Array.isArray(data)) {
417:       // Update entire section
418:       updated[section] = {
419:         ...(updated[section] || {}),
420:         ...(data as Record<string, NutrientData>),
421:       };
422:     }
423:
424:     return {
425:       ...prev,
426:       data: updated,
427:     };
428:   });
429:
430:   // Also update parent state
431:   if (propUpdateNutritionData) {
432:     propUpdateNutritionData(section, field, data);
433:   }
434: };
435:
436: const resetAddForm = () => {
437:   setNewNutrient({
438:     name: "",
439:     score: 5,
440:     healthImpact: "",
441:     intakeLevel: "NORMAL INTAKE",
442:     source: "DIET",
443:   });
444: };
445:
446: // ? FIXED: Proper useEffect
447: useEffect(() => {
448:   if (propNutritionData) {
449:     setQuote(propNutritionData.quote || "");
450:     setDescription(propNutritionData.description || "");
451:     setLocalNutritionData({
452:       data: propNutritionData.data || {},
453:     });
454:
455:     const keys = Object.keys(propNutritionData.data || {});
456:     if (keys.length > 0 && !activeSection) {
457:       setActiveSection(keys[0]);
458:     }
459:   }
460: }, [propNutritionData]); // ? Watch for prop changes
461:
462: if (!localNutritionData || typeof localNutritionData.data !== "object") {
463:   return <div>Loading nutrition data...</div>;
464: }
465:
466: const renderNutrientField = (
467:   section: string,
468:   field: string,
469:   data: NutrientData,

```

```

470:     title: string
471:   ) => (
472:     <Card key={field} className="border-2 border-green-100 bg-green-50/30">
473:       <CardHeader className="pb-3">
474:         <CardTitle className="text-lg sm:text-xl md:text-2xl text-green-800 flex items-center justify-between">
475:           {isEditMode === field ? (
476:             <div className="flex items-center gap-2 flex-1">
477:               <Input
478:                 value={editingName}
479:                 onChange={(e) => setEditingName(e.target.value)}
480:                 className="text-lg font-semibold border-2 focus:border-green-500"
481:                 onPress={(e) =>
482:                   e.key === "Enter" && handleRename(section, field)
483:                 }
484:               />
485:               <Button size="sm" onClick={() => handleRename(section, field)}>
486:                 ?
487:               </Button>
488:               <Button
489:                 size="sm"
490:                 variant="outline"
491:                 onClick={() => {
492:                   setIsEditMode(null);
493:                   setEditingName("");
494:                 }}
495:               >
496:                 ?
497:               </Button>
498:             </div>
499:           ) : (
500:             <div className="flex items-center gap-2 flex-1">
501:               <span>{title}</span>
502:               <Button
503:                 size="sm"
504:                 variant="ghost"
505:                 onClick={() => {
506:                   setIsEditMode(field);
507:                   setEditingName(title);
508:                 }}
509:               >
510:                 <Edit3 className="h-3 w-3" />
511:               </Button>
512:             </div>
513:           )}
514:         <div className="flex items-center gap-2">
515:           <Badge
516:             variant="outline"
517:             className={`{
518:               data.score >= 7
519:               ? "border-red-500 text-red-700"
520:               : data.score >= 4
521:               ? "border-yellow-500 text-yellow-700"
522:               : "border-green-500 text-green-700"
523:             }`}
524:           >
525:             Score: {data.score}/10
526:           </Badge>
527:           <Button
528:             size="sm"
529:             variant="ghost"
530:             onClick={() => handleDeleteField(section, field)}
531:             className="text-red-600 hover:bg-red-100"
532:           >
533:             <Trash2 className="h-4 w-4" />
534:           </Button>
535:         </div>
536:       </CardTitle>
537:     </CardHeader>
538:     <CardContent className="space-y-4">
539:       <div className="grid grid-cols-2 gap-4">
540:         <div className="space-y-2">
541:           <Label className="text-sm font-medium">Score (1-10)</Label>
542:           <Input

```



```

543:         type="number"
544:         min="1"
545:         max="10"
546:         value={data.score}
547:         onChange={(e) => {
548:             const score = Math.max(
549:                 1,
550:                 Math.min(10, Number.parseInt(e.target.value) || 1)
551:             );
552:             updateLocalNutritionData(section, field, { score });
553:         }}
554:         className="border-2 focus:border-green-500"
555:     />
556: </div>
557: <div className="space-y-2">
558:     <Label className="text-sm font-medium">Health Impact</Label>
559:     <Input
560:         value={data.healthImpact}
561:         onChange={(e) =>
562:             updateLocalNutritionData(section, field, {
563:                 healthImpact: e.target.value,
564:             })
565:         }
566:         placeholder="e.g., Skin & Vision"
567:         className="border-2 focus:border-green-500"
568:     />
569: </div>
570: </div>
571: <div className="space-y-2">
572:     <Label className="text-sm font-medium">Intake Level</Label>
573:     <Select
574:         value={data.intakeLevel}
575:         onChange={(value) =>
576:             updateLocalNutritionData(section, field, { intakeLevel: value })
577:         }
578:     >
579:         <SelectTrigger className="border-2 focus:border-green-500">
580:             <SelectValue />
581:         </SelectTrigger>
582:         <SelectContent>
583:             <SelectItem value="ENHANCED INTAKE">ENHANCED INTAKE</SelectItem>
584:             <SelectItem value="NORMAL INTAKE">NORMAL INTAKE</SelectItem>
585:             <SelectItem value="RESTRICTED INTAKE">
586:                 RESTRICTED INTAKE
587:             </SelectItem>
588:             <SelectItem value="ENHANCED INTAKE (METHYLCOBALAMIN)">
589:                 ENHANCED INTAKE (METHYLCOBALAMIN)
590:             </SelectItem>
591:             <SelectItem value="ENHANCED INTAKE (L METHYLFOLATE)">
592:                 ENHANCED INTAKE (L METHYLFOLATE)
593:             </SelectItem>
594:             <SelectItem value="ENHANCED INTAKE & SUN EXPOSURE">
595:                 ENHANCED INTAKE & SUN EXPOSURE
596:             </SelectItem>
597:             <SelectItem value="ENHANCED INTAKE (VEG SOURCES)">
598:                 ENHANCED INTAKE (VEG SOURCES)
599:             </SelectItem>
600:         </SelectContent>
601:     </Select>
602: </div>
603: <div className="space-y-2">
604:     <Label className="text-sm font-medium">Source</Label>
605:     <Select
606:         value={data.source}
607:         onChange={(value) =>
608:             updateLocalNutritionData(section, field, { source: value })
609:         }
610:     >
611:         <SelectTrigger className="border-2 focus:border-green-500">
612:             <SelectValue />
613:         </SelectTrigger>
614:         <SelectContent>
615:             <SelectItem value="DIET">DIET</SelectItem>

```

```

616:         <SelectItem value="SUPPLEMENTS">SUPPLEMENTS</SelectItem>
617:         <SelectItem value="DIET & SUPPLEMENTS">
618:             DIET & SUPPLEMENTS
619:         </SelectItem>
620:         <SelectItem value="DIET & SUN EXPOSURE">
621:             DIET & SUN EXPOSURE
622:         </SelectItem>
623:         <SelectItem value="ENHANCED INTAKE (VEG/FISH OIL) & DIET">
624:             ENHANCED INTAKE (VEG/FISH OIL) & DIET
625:         </SelectItem>
626:     </SelectContent>
627: </Select>
628: </div>
629: </CardContent>
630: </Card>
631: );
632:
633: const getSectionTitle = (section: string) => {
634:     switch (section) {
635:         case "vitamins":
636:             return "Vitamins";
637:         default:
638:             return section;
639:     }
640: };
641:
642: return (
643:     <Card className="shadow-lg border-0 bg-white">
644:         <CardHeader className="bg-gradient-to-r from-green-500 to-emerald-600 rounded-lg p-4 sm:p-6 text-white">
645:             <CardTitle className="text-base sm:text-xl md:text-2xl font-bold flex items-center justify-center">
646:                 Nutrition Analysis
647:             </CardTitle>
648:         </CardHeader>
649:
650:         <div className="text-end my-2 sm:my-4 px-4 sm:px-6 md:px-8">
651:             <Button onClick={onSave} size="sm">
652:                 <Save className="h-4 w-4 mr-2" />
653:                 Save Changes
654:             </Button>
655:         </div>
656:         <CardContent className="space-y-4 sm:space-y-6 md:space-y-8 p-4 sm:p-6 md:p-8">
657:             { /* Quote and Description Section */ }
658:             <div className="space-y-3 sm:space-y-4 bg-gray-50 p-3 sm:p-4 rounded-lg">
659:                 <div>
660:                     <Label className="text-sm font-medium">Quote</Label>
661:                     <Input
662:                         value={quote}
663:                         onChange={(e) => setQuote(e.target.value)}
664:                         placeholder="Enter a motivating quote..."
665:                         className="mt-1"
666:                     />
667:                 </div>
668:                 <div>
669:                     <Label className="text-sm font-medium">Description</Label>
670:                     <Input
671:                         value={description}
672:                         onChange={(e) => setDescription(e.target.value)}
673:                         placeholder="Short summary for this page..."
674:                         className="mt-1"
675:                     />
676:                 </div>
677:             </div>
678:
679:             { /* Section Navigation and Action Buttons */ }
680:             <div className="flex flex-col sm:flex-row justify-between items-start sm:items-center gap-4 bg-white">
681:                 <div className="flex gap-2 flex-wrap w-full sm:w-auto">
682:                     {Object.entries(localNutritionData.data).map(
683:                         ([section, fields]) => (
684:                             <div key={section} className="relative">
685:                                 <Button
686:                                     variant={activeSection === section ? "default" : "outline"}
687:                                     size="sm"
688:                                     onClick={() => setActiveSection(section)}

```

```

689:         className="flex items-center gap-2 pr-6 text-xs sm:text-sm"
690:         >
691:         <span className="capitalize">
692:             {section.replace(/([A-Z])/g, " $1")}
693:         </span>
694:     </Button>
695:     <button
696:         onClick={() => setConfirmDelete(section)}
697:         className="absolute -right-1 -top-1 bg-red-500 text-white rounded-full w-4 h-4 sm:w-5 sm:h-5"
698:         title="Delete Category"
699:     >
700:         ?
701:     </button>
702: </div>
703: )
704: }}
705:
706: <Dialog
707:     open={!confirmDelete}
708:     onOpenChange={(open) => !open && setConfirmDelete(null)}
709: >
710:     <DialogContent className="mx-4 sm:mx-0 max-w-sm sm:max-w-md">
711:         <DialogHeader>
712:             <DialogTitle>Confirm Delete</DialogTitle>
713:             <DialogDescription>
714:                 Are you sure you want to delete the category{" "}
715:                 <strong>{confirmDelete}</strong>?
716:             </DialogDescription>
717:         </DialogHeader>
718:         <DialogFooter className="flex-col sm:flex-row gap-2">
719:             <Button
720:                 variant="outline"
721:                 onClick={() => setConfirmDelete(null)}
722:                 className="w-full sm:w-auto"
723:             >
724:                 Cancel
725:             </Button>
726:             <Button
727:                 variant="destructive"
728:                 onClick={() => {
729:                     if (confirmDelete) {
730:                         handleDeleteSection(confirmDelete);
731:                         setConfirmDelete(null);
732:                     }
733:                 }}
734:                 className="w-full sm:w-auto"
735:             >
736:                 Yes, Delete
737:             </Button>
738:         </DialogFooter>
739:     </DialogContent>
740: </Dialog>
741: </div>
742:
743: <div className="flex gap-2 w-full sm:w-auto">
744:     {/* Add New Nutrient Button */}
745:     <Dialog open={isAddDialogOpen} onOpenChange={setIsAddDialogOpen}>
746:         <DialogTrigger asChild>
747:             <Button
748:                 variant="outline"
749:                 size="sm"
750:                 className="bg-green-50 border-green-200 text-green-700 hover:bg-green-100 flex-1 sm:flex-1"
751:                 disabled={!activeSection}
752:             >
753:                 <Plus className="h-3 w-3 sm:h-4 sm:w-4 mr-1 sm:mr-2" />
754:                 Add New
755:             </Button>
756:         </DialogTrigger>
757:         <DialogContent className="mx-4 sm:mx-0 max-w-[95vw] sm:max-w-2xl max-h-[90vh] overflow-y-auto">
758:             <DialogHeader>
759:                 <DialogTitle className="text-lg sm:text-xl">
760:                     Add New{" "}
761:                     {getSectionTitle(activeSection)}

```

```

762:         .replace(/[^\w\s]/gi, "")
763:         .replace(/\b\w/g, (c) => c.toUpperCase())}{" "}
764:         Item
765:     </DialogTitle>
766:
767:     <DialogDescription className="text-sm">
768:         Enter the details for the new nutrient in the{" "}
769:         {activeSection} section.
770:     </DialogDescription>
771: </DialogHeader>
772: <div className="space-y-4 sm:space-y-6 py-4">
773:     <div className="space-y-2">
774:         <Label htmlFor="nutrientName">Nutrient Name *</Label>
775:         <Input
776:             id="nutrientName"
777:             value={newNutrient.name}
778:             onChange={(e) =>
779:                 setNewNutrient({ ...newNutrient, name: e.target.value })
780:             }
781:             placeholder="e.g., Vitamin E, Selenium, etc."
782:             className="border-2 focus:border-green-500"
783:         />
784:     </div>
785:
786:     <div className="grid grid-cols-1 sm:grid-cols-2 gap-4">
787:         <div className="space-y-2">
788:             <Label htmlFor="nutrientScore">Score (1-10)</Label>
789:             <Input
790:                 id="nutrientScore"
791:                 type="number"
792:                 min="1"
793:                 max="10"
794:                 value={newNutrient.score}
795:                 onChange={(e) =>
796:                     setNewNutrient({
797:                         ...newNutrient,
798:                         score: Math.max(
799:                             1,
800:                             Math.min(10, Number.parseInt(e.target.value) || 1)
801:                         ),
802:                     })
803:                 }
804:                 className="border-2 focus:border-green-500"
805:             />
806:         </div>
807:         <div className="space-y-2">
808:             <Label htmlFor="nutrientImpact">Health Impact</Label>
809:             <Input
810:                 id="nutrientImpact"
811:                 value={newNutrient.healthImpact}
812:                 onChange={(e) =>
813:                     setNewNutrient({
814:                         ...newNutrient,
815:                         healthImpact: e.target.value,
816:                     })
817:                 }
818:                 placeholder="e.g., Skin & Vision"
819:                 className="border-2 focus:border-green-500"
820:             />
821:         </div>
822:     </div>
823:
824:     <div className="space-y-2">
825:         <Label htmlFor="nutrientIntake">Intake Level</Label>
826:         <Select
827:             value={newNutrient.intakeLevel}
828:             onChange={(value) =>
829:                 setNewNutrient({ ...newNutrient, intakeLevel: value })
830:             }
831:         >
832:             <SelectTrigger className="border-2 focus:border-green-500">
833:                 <SelectValue />
834:             </SelectTrigger>

```

```

835:         <SelectContent>
836:             <SelectItem value="ENHANCED INTAKE">
837:                 ENHANCED INTAKE
838:             </SelectItem>
839:             <SelectItem value="NORMAL INTAKE">
840:                 NORMAL INTAKE
841:             </SelectItem>
842:             <SelectItem value="RESTRICTED INTAKE">
843:                 RESTRICTED INTAKE
844:             </SelectItem>
845:             <SelectItem value="ENHANCED INTAKE (METHYLCOBALAMIN)">
846:                 ENHANCED INTAKE (METHYLCOBALAMIN)
847:             </SelectItem>
848:             <SelectItem value="ENHANCED INTAKE (L METHYLFOLATE)">
849:                 ENHANCED INTAKE (L METHYLFOLATE)
850:             </SelectItem>
851:             <SelectItem value="ENHANCED INTAKE & SUN EXPOSURE">
852:                 ENHANCED INTAKE & SUN EXPOSURE
853:             </SelectItem>
854:             <SelectItem value="ENHANCED INTAKE (VEG SOURCES)">
855:                 ENHANCED INTAKE (VEG SOURCES)
856:             </SelectItem>
857:         </SelectContent>
858:     </Select>
859: </div>
860:
861: <div className="space-y-2">
862:     <Label htmlFor="nutrientSource">Source</Label>
863:     <Select
864:         value={newNutrient.source}
865:         onChange={(value) =>
866:             setNewNutrient({ ...newNutrient, source: value })
867:         }
868:     >
869:         <SelectTrigger className="border-2 focus:border-green-500">
870:             <SelectValue />
871:         </SelectTrigger>
872:         <SelectContent>
873:             <SelectItem value="DIET">DIET</SelectItem>
874:             <SelectItem value="SUPPLEMENTS">SUPPLEMENTS</SelectItem>
875:             <SelectItem value="DIET & SUPPLEMENTS">
876:                 DIET & SUPPLEMENTS
877:             </SelectItem>
878:             <SelectItem value="DIET & SUN EXPOSURE">
879:                 DIET & SUN EXPOSURE
880:             </SelectItem>
881:             <SelectItem value="ENHANCED INTAKE (VEG/FISH OIL) & DIET">
882:                 ENHANCED INTAKE (VEG/FISH OIL) & DIET
883:             </SelectItem>
884:         </SelectContent>
885:     </Select>
886: </div>
887: </div>
888: <DialogFooter className="flex-col sm:flex-row gap-2">
889:     <Button
890:         variant="outline"
891:         onClick={() => {
892:             setIsAddDialogOpen(false);
893:             resetAddForm();
894:         }}
895:         className="w-full sm:w-auto"
896:     >
897:         Cancel
898:     </Button>
899:     <Button
900:         onClick={handleAddNutrient}
901:         disabled={!newNutrient.name.trim()}
902:         className="w-full sm:w-auto"
903:     >
904:         Add Nutrient
905:     </Button>
906: </DialogFooter>
907: </DialogContent>

```

```

908:         </Dialog>
909:
910:     <Dialog
911:       open={isAddSectionDialogOpen}
912:       onOpenChange={setIsAddSectionDialogOpen}
913:     >
914:       <DialogTrigger asChild>
915:         <Button
916:           variant="outline"
917:           size="sm"
918:           className="bg-blue-50 border-blue-200 text-blue-700 hover:bg-blue-100 flex-1 sm:flex-none"
919:         >
920:           <Plus className="h-3 w-3 sm:h-4 sm:w-4 mr-1 sm:mr-2" />
921:           Add Category
922:         </Button>
923:       </DialogTrigger>
924:       <DialogContent className="mx-4 sm:mx-0 max-w-sm sm:max-w-md">
925:         <DialogHeader>
926:           <DialogTitle>Add New Category</DialogTitle>
927:           <DialogDescription>
928:             Create a new category (e.g., "aminoAcids", "fiber", etc.)
929:           </DialogDescription>
930:         </DialogHeader>
931:         <div className="space-y-4 py-4">
932:           <Label htmlFor="sectionName">Category Name *</Label>
933:           <Input
934:             id="sectionName"
935:             placeholder="e.g., aminoAcids"
936:             value={newSectionName}
937:             onChange={(e) => setNewSectionName(e.target.value)}
938:           />
939:         </div>
940:         <DialogFooter className="flex-col sm:flex-row gap-2">
941:           <Button
942:             variant="outline"
943:             onClick={() => {
944:               setIsAddSectionDialogOpen(false);
945:               setNewSectionName("");
946:             }}
947:             className="w-full sm:w-auto"
948:           >
949:             Cancel
950:           </Button>
951:           <Button
952:             onClick={addNewCategory}
953:             disabled={!newSectionName.trim()}
954:             className="w-full sm:w-auto"
955:           >
956:             Add Category
957:           </Button>
958:         </DialogFooter>
959:       </DialogContent>
960:     </Dialog>
961:   </div>
962: </div>
963:
964:   { /* Section Content */ }
965:   <div className="space-y-4 sm:space-y-6">
966:     <div className="flex items-center justify-between">
967:       <h3 className="text-lg sm:text-xl font-semibold text-gray-800 border-b-2 border-green-200 pb-2">
968:         {getSectionTitle(activeSection)}
969:         .replace(/[^\w\s]/gi, "")
970:         .replace(/^(\\w)/, (c) => c.toUpperCase())
971:       </h3>
972:     </div>
973:     <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-4 sm:gap-6">
974:       {Object.entries(localNutritionData.data[activeSection] || {}).map(
975:         ([key, item]) =>
976:           renderNutrientField(
977:             activeSection,
978:             key,
979:             item as NutrientData,
980:             key

```

```

981:         .replace(/[A-Z])/g, " $1")
982:         .replace(/^./, (str) => str.toUpperCase())
983:     )
984:   })
985: </div>
986: {Object.keys(localNutritionData.data[activeSection] || {}).length ===
987:   0 && (
988:     <div className="text-center py-8 sm:py-12 bg-gray-50 rounded-lg border-2 border-dashed border-g
989:       <div className="text-gray-500 text-base sm:text-lg mb-2">
990:         No items in this section
991:       </div>
992:       <div className="text-gray-400 text-sm">
993:         Click "Add New" to add your first nutrient
994:       </div>
995:     </div>
996:   )}
997: </div>
998: </CardContent>
999: </Card>
1000: );
1001: }

```

## ■ File: components\admin\patient-info-admin.tsx

```

1: "use client";
2:
3: import {
4:   Card,
5:   CardContent,
6:   CardDescription,
7:   CardHeader,
8:   CardTitle,
9: } from "@components/ui/card";
10: import { Textarea } from "@components/ui/textarea";
11: import { Input } from "@components/ui/input";
12: import { Label } from "@components/ui/label";
13: import {
14:   Select,
15:   SelectContent,
16:   SelectItem,
17:   SelectTrigger,
18:   SelectValue,
19: } from "@components/ui/select";
20: import { Button } from "@components/ui/button";
21: import { Save, RotateCcw, Upload, X } from "lucide-react";
22: import { useRef } from "react";
23: import type { PatientInfo } from "@types/report-types";
24:
25: interface PatientInfoAdminProps {
26:   patientInfo: PatientInfo;
27:   updatePatientInfo: (field: keyof PatientInfo, value: string) => void;
28:   onSave: () => void;
29:   onReset: () => void;
30: }
31:
32: export default function PatientInfoAdmin({
33:   patientInfo,
34:   updatePatientInfo,
35:   onSave,
36:   onReset,
37: }: PatientInfoAdminProps) {
38:   const fileInput1Ref = useRef<HTMLInputElement>(null);
39:   const fileInput2Ref = useRef<HTMLInputElement>(null);
40:
41:   const handleImageUpload = async (
42:     e: React.ChangeEvent<HTMLInputElement>,
43:     key: "signature1" | "signature2"
44:   ) => {
45:     const file = e.target.files?.[0];
46:     if (!file) return;
47:

```

```

48:     const formData = new FormData();
49:     formData.append("file", file);
50:
51:     try {
52:       const res = await fetch("/api/upload-signature", {
53:         method: "POST",
54:         body: formData,
55:       });
56:
57:       if (!res.ok) throw new Error("Upload failed");
58:       const { url } = await res.json();
59:
60:       // Save backend URL instead of blob URL
61:       updatePatientInfo(key, url);
62:     } catch (err) {
63:       console.error(err);
64:       alert("Failed to upload signature");
65:     }
66:   };
67:
68:   const removeSignature = (
69:     key: "signature1" | "signature2",
70:     ref: React.RefObject<HTMLInputElement>
71:   ) => {
72:     updatePatientInfo(key, null);
73:     if (ref.current) ref.current.value = "";
74:   };
75:
76:   return (
77:     <Card className="shadow-lg border-0 bg-white">
78:       <CardHeader className="bg-gradient-to-r from-blue-500 to-purple-600 text-white rounded-lg px-3 py-4">
79:         <CardTitle className="text-lg sm:text-xl md:text-2xl lg:text-3xl font-semibold uppercase text-center">
80:           Patient Information
81:         </CardTitle>
82:       </CardHeader>
83:
84:       <CardContent className="space-y-6 p-8">
85:         { /* Action Buttons */ }
86:         <div className="flex justify-end gap-2 mb-6">
87:           <Button onClick={onSave} size="sm">
88:             <Save className="h-4 w-4 mr-2" />
89:             Save Changes
90:           </Button>
91:         </div>
92:
93:         { /* Basic Information */ }
94:         <div className="space-y-4">
95:           <h3 className="text-lg font-semibold text-gray-800 pb-2">
96:             Personal Information
97:           </h3>
98:           <div className="grid md:grid-cols-3 gap-6">
99:             <div className="space-y-2">
100:               <Label
101:                 htmlFor="name"
102:                 className="text-sm font-semibold text-gray-700 required"
103:               >
104:                 Patient Name *
105:               </Label>
106:               <Input
107:                 id="name"
108:                 value={patientInfo.name}
109:                 onChange={(e) => updatePatientInfo("name", e.target.value)}
110:                 placeholder="Enter full patient name"
111:                 className="border-2"
112:                 required
113:               />
114:             </div>
115:             <div className="space-y-2">
116:               <Label
117:                 htmlFor="gender"
118:                 className="text-sm font-semibold text-gray-700 required"
119:               >
120:                 Gender *

```



```

121:         </Label>
122:         <Select
123:             value={patientInfo.gender}
124:             onChange={(value) => updatePatientInfo("gender", value)}
125:         >
126:             <SelectTrigger className="border-2">
127:                 <SelectValue placeholder="Select gender" />
128:             </SelectTrigger>
129:             <SelectContent>
130:                 <SelectItem value="MALE">Male</SelectItem>
131:                 <SelectItem value="FEMALE">Female</SelectItem>
132:                 <SelectItem value="OTHER">Other</SelectItem>
133:             </SelectContent>
134:         </Select>
135:     </div>
136:     <div className="space-y-2">
137:         <Label
138:             htmlFor="birthDate"
139:             className="text-sm font-semibold text-gray-700 required"
140:         >
141:             Birth Date *
142:         </Label>
143:         <Input
144:             id="birthDate"
145:             type="date"
146:             value={patientInfo.birthDate}
147:             onChange={(e) => updatePatientInfo("birthDate", e.target.value)}
148:             className="border-2"
149:             required
150:         />
151:     </div>
152: </div>
153: </div>
154:
155: { /* Sample Information */ }
156: <div className="space-y-4">
157:     <h3 className="text-lg font-semibold text-gray-800 border-b pb-2">
158:         Sample Information
159:     </h3>
160:     <div className="grid md:grid-cols-3 gap-6">
161:         <div className="space-y-2">
162:             <Label
163:                 htmlFor="sampleCode"
164:                 className="text-sm font-semibold text-gray-700 required"
165:             >
166:                 Sample Code *
167:             </Label>
168:             <Input
169:                 id="sampleCode"
170:                 value={patientInfo.sampleCode}
171:                 onChange={(e) =>
172:                     updatePatientInfo("sampleCode", e.target.value)
173:                 }
174:                 placeholder="e.g., DNL1000000110"
175:                 className="border-2 animate-pulse border-red-400 shadow-red-500/50 focus:shadow-red-500/80"
176:                 required
177:             />
178:             <p className="text-xs text-gray-500">
179:                 Unique identifier for the DNA sample
180:             </p>
181:         </div>
182:         <div className="space-y-2">
183:             <Label
184:                 htmlFor="sampleDate"
185:                 className="text-sm font-semibold text-gray-700 required"
186:             >
187:                 Sample Collection Date *
188:             </Label>
189:             <Input
190:                 id="sampleDate"
191:                 type="date"
192:                 value={patientInfo.sampleDate}
193:                 onChange={(e) =>

```

```

194:         updatePatientInfo("sampleDate", e.target.value)
195:     }
196:     className="border-2"
197:     required
198: />
199: </div>
200: <div className="space-y-2">
201:     <Label
202:         htmlFor="reportDate"
203:         className="text-sm font-semibold text-gray-700 required"
204:     >
205:         Report Date *
206:     </Label>
207:     <Input
208:         id="reportDate"
209:         type="date"
210:         value={patientInfo.reportDate}
211:         onChange={(e) =>
212:             updatePatientInfo("reportDate", e.target.value)
213:         }
214:         className="border-2"
215:         required
216:     />
217: </div>
218: </div>
219: </div>
220:
221: { /* Authorization & Verification */ }
222: <div className="space-y-6 border-t border-black py-6">
223:     { /* Section Title */ }
224:     <h2 className="text-center text-lg md:text-xl font-bold text-gray-800 pb-2">
225:         REPORT AUTHENTICATION & ANALYTICS
226:     </h2>
227:
228:     { /* Roles */ }
229:     <div className="grid md:grid-cols-2 gap-6 text-center">
230:         <div>
231:             <p className="text-base font-semibold text-gray-700 uppercase">
232:                 Genomic Data Analytics
233:             <br />
234:             Checked & Verified By
235:         </p>
236:     </div>
237:     <div>
238:         <p className="text-base font-semibold text-gray-700 uppercase">
239:             Scientific Content
240:         <br />
241:         Checked & Verified By
242:     </p>
243:     </div>
244: </div>
245:
246: { /* Signature Previews */ }
247: <div className="grid md:grid-cols-2 gap-6 text-center mt-4">
248:     { /* Signature 1 */ }
249:     <div className="space-y-2">
250:         {patientInfo.signature1 ? (
251:             <div className="relative w-full border rounded bg-white p-3">
252:                 <img
253:                     src={patientInfo.signature1}
254:                     alt="Primary Signature"
255:                     className="max-h-24 mx-auto object-contain"
256:                 />
257:                 <Button
258:                     onClick={() => removeSignature("signature1", fileInput1Ref)}
259:                     variant="outline"
260:                     size="sm"
261:                     className="absolute top-1 right-1 h-6 w-6 p-0"
262:                 >
263:                     <X className="h-3 w-3" />
264:                 </Button>
265:             </div>
266:         ) : (

```

```

267:         <div className="border-2 border-dashed border-gray-300 rounded-lg p-4 bg-gray-50">
268:             <Upload className="h-8 w-8 text-gray-400 mx-auto mb-2" />
269:             <p className="text-sm text-gray-500 mb-2">
270:                 Upload primary signature
271:             </p>
272:             <Button
273:                 onClick={() => fileInput1Ref.current?.click()}
274:                 variant="outline"
275:                 size="sm"
276:             >
277:                 <Upload className="h-4 w-4 mr-2" />
278:                 Choose File
279:             </Button>
280:         </div>
281:     )}
282:     <input
283:         ref={fileInput1Ref}
284:         type="file"
285:         accept="image/*"
286:         onChange={(e) => handleImageUpload(e, "signature1")}
287:         className="hidden"
288:     />
289:     <p className="text-sm font-medium text-gray-800 mt-1">
290:         {patientInfo.checkedBy || "Dr. Amol D Raut (PhD)}
291:     </p>
292: </div>
293:
294: { /* Signature 2 */ }
295: <div className="space-y-2">
296:     {patientInfo.signature2 ? (
297:         <div className="relative w-full border rounded bg-white p-3">
298:             <img
299:                 src={patientInfo.signature2}
300:                 alt="Secondary Signature"
301:                 className="max-h-24 mx-auto object-contain"
302:             />
303:             <Button
304:                 onClick={() => removeSignature("signature2", fileInput2Ref)}
305:                 variant="outline"
306:                 size="sm"
307:                 className="absolute top-1 right-1 h-6 w-6 p-0"
308:             >
309:                 <X className="h-3 w-3" />
310:             </Button>
311:         </div>
312:     ) : (
313:         <div className="border-2 border-dashed border-gray-300 rounded-lg p-4 bg-gray-50">
314:             <Upload className="h-8 w-8 text-gray-400 mx-auto mb-2" />
315:             <p className="text-sm text-gray-500 mb-2">
316:                 Upload secondary signature
317:             </p>
318:             <Button
319:                 onClick={() => fileInput2Ref.current?.click()}
320:                 variant="outline"
321:                 size="sm"
322:             >
323:                 <Upload className="h-4 w-4 mr-2" />
324:                 Choose File
325:             </Button>
326:         </div>
327:     )}
328:     <input
329:         ref={fileInput2Ref}
330:         type="file"
331:         accept="image/*"
332:         onChange={(e) => handleImageUpload(e, "signature2")}
333:         className="hidden"
334:     />
335:     <p className="text-sm font-medium text-gray-800 mt-1">
336:         {patientInfo.scientificContent || "Dr. Gouri A Raut (PhD)}
337:     </p>
338: </div>
339: </div>

```

```

340:
341:     {/* Optional Text Fields (if editing verifier names manually) */}
342:     <div className="grid md:grid-cols-2 gap-6 mt-6">
343:         <div className="space-y-2">
344:             <Label
345:                 htmlFor="checkedBy"
346:                 className="text-sm font-semibold text-gray-700"
347:             >
348:                 Checked & Verified By
349:             </Label>
350:             <Input
351:                 id="checkedBy"
352:                 value={patientInfo.checkedBy}
353:                 onChange={(e) => updatePatientInfo("checkedBy", e.target.value)}
354:                 placeholder="Enter verifier name"
355:                 className="border-2"
356:             />
357:         </div>
358:         <div className="space-y-2">
359:             <Label
360:                 htmlFor="scientificContent"
361:                 className="text-sm font-semibold text-gray-700"
362:             >
363:                 Scientific Content Verified By
364:             </Label>
365:             <Input
366:                 id="scientificContent"
367:                 value={patientInfo.scientificContent}
368:                 onChange={(e) =>
369:                     updatePatientInfo("scientificContent", e.target.value)
370:                 }
371:                 placeholder="Enter scientific content verifier"
372:                 className="border-2"
373:             />
374:         </div>
375:     </div>
376: </div>
377:
378: <div className="space-y-3">
379:     <Label
380:         htmlFor="disclaimer"
381:         className="text-lg font-semibold text-gray-800"
382:     >
383:         Disclaimer
384:     </Label>
385:     <Textarea
386:         id="disclaimer"
387:         value={patientInfo.disclaimer}
388:         onChange={(e) => updatePatientInfo("disclaimer", e.target.value)}
389:         placeholder="Enter disclaimer text"
390:         rows={5}
391:         className="border-2 focus:border-green-500 text-sm leading-relaxed"
392:     />
393: </div>
394:
395: <style jsx>{`
396:     .required:after {
397:         content: " *";
398:         color: #ef4444;
399:     }
400: `}</style>
401: </CardContent>
402: </Card>
403: );
404: }

```

## ■ File: components\admin\pdf-generator.tsx

```

=====
1: "use client";
2:
3: import { Button } from "@/components/ui/button";

```

```

4: import { Card, CardContent, CardHeader, CardTitle } from "@/components/ui/card";
5: import { Download, Printer, Eye } from "lucide-react";
6: import { useState, useRef } from "react";
7: import type { ComprehensiveReportData } from "@/types/report-types";
8: import { useToast } from "@/components/ui/use-toast";
9: import html2canvas from "html2canvas";
10: import jsPDF from "jspdf";
11: import ComprehensiveReportViewer from "@/components/comprehensive-report-viewer";
12:
13: interface PDFGeneratorProps {
14:   reportData: ComprehensiveReportData;
15:   reportIndex: number;
16:   patientCode: string;
17: }
18:
19: export default function PDFGenerator({
20:   reportData,
21:   patientCode,
22:   reportIndex,
23: }: PDFGeneratorProps) {
24:   const [isGenerating, setIsGenerating] = useState(false);
25:   const { toast } = useToast();
26:   const [showForCapture, setShowForCapture] = useState(false);
27:   const reportRef = useRef<HTMLDivElement>(null);
28:
29:   /** Print logic (unchanged) */
30:   const handlePrint = () => {
31:     const printWindow = window.open("/report/pdf", "_blank");
32:     if (!printWindow) {
33:       toast({
34:         title: "Error",
35:         description: "Please allow pop-ups to print the report.",
36:         variant: "destructive",
37:       });
38:     }
39:   };
40:
41:   /** Enhanced multi-page PDF capture with smart page breaks */
42:   const handleMultiPagePDF = async () => {
43:     setIsGenerating(true);
44:     setShowForCapture(true);
45:
46:     try {
47:       // Wait for render and resources
48:       await new Promise((r) => setTimeout(r, 1000));
49:
50:       if (!reportRef.current) {
51:         throw new Error("Report not found");
52:       }
53:
54:       // Wait for fonts and images to load
55:       if (document.fonts?.ready) await document.fonts.ready;
56:       const images = reportRef.current.querySelectorAll("img");
57:       await Promise.all(
58:         Array.from(images).map((img) =>
59:           img.complete
60:             ? Promise.resolve()
61:             : new Promise<void>((resolve) => {
62:                 img.onload = () => resolve();
63:                 img.onerror = () => resolve();
64:               })
65:         )
66:       );
67:
68:       // PDF configuration
69:       const pdf = new jsPDF("p", "mm", "a4");
70:       const pageWidth = 210; // A4 width in mm
71:       const pageHeight = 297; // A4 height in mm
72:       const margin = 10; // 10mm margin
73:       const contentWidth = pageWidth - 2 * margin;
74:       const contentHeight = pageHeight - 2 * margin;
75:
76:       // High quality scale for better resolution

```

```

77:     const scale = 3;
78:     const element = reportRef.current;
79:
80:     // Calculate dimensions
81:     const elementWidth = element.offsetWidth;
82:     const elementHeight = element.scrollHeight;
83:
84:     // Calculate how much content fits per page in pixels
85:     const pixelsPerMM = (elementWidth * scale) / contentWidth;
86:     const pageHeightInPixels = contentHeight * pixelsPerMM;
87:
88:     // Find all sections to avoid cutting them
89:     const sections = element.querySelectorAll('.report-section, .card, .mb-6, .space-y-4 > div, .grid > d
90:     const sectionPositions: Array<{ top: number; height: number; element: Element }> = [];
91:
92:     sections.forEach(section => {
93:         const rect = section.getBoundingClientRect();
94:         const elementRect = element.getBoundingClientRect();
95:         sectionPositions.push({
96:             top: (rect.top - elementRect.top) * scale,
97:             height: rect.height * scale,
98:             element: section
99:         });
100:     });
101:
102:     let currentY = 0;
103:     let pageNumber = 0;
104:
105:     while (currentY < elementHeight * scale) {
106:         const remainingHeight = (elementHeight * scale) - currentY;
107:         let sliceHeight = Math.min(pageHeightInPixels, remainingHeight);
108:
109:         // Find the best break point to avoid cutting sections
110:         if (remainingHeight > pageHeightInPixels) {
111:             const cutPoint = currentY + sliceHeight;
112:
113:             // Find sections that would be cut
114:             const problematicSections = sectionPositions.filter(section =>
115:                 section.top < cutPoint &&
116:                 section.top + section.height > cutPoint &&
117:                 section.height < pageHeightInPixels * 0.8 // Only adjust if section isn't too large
118:             );
119:
120:             if (problematicSections.length > 0) {
121:                 // Adjust slice height to end before the problematic section
122:                 const earliestSection = problematicSections.reduce((min, section) =>
123:                     section.top < min.top ? section : min
124:                 );
125:
126:                 const newSliceHeight = earliestSection.top - currentY;
127:                 if (newSliceHeight > pageHeightInPixels * 0.3) { // Ensure we don't create tiny pages
128:                     sliceHeight = newSliceHeight;
129:                 }
130:             }
131:         }
132:
133:         // Capture the page slice
134:         const canvas = await html2canvas(element, {
135:             scale: scale,
136:             useCORS: true,
137:             allowTaint: false,
138:             backgroundColor: "#FFFFFF",
139:             logging: false,
140:             width: elementWidth,
141:             height: sliceHeight / scale,
142:             x: 0,
143:             y: currentY / scale,
144:             windowWidth: elementWidth,
145:             windowHeight: sliceHeight / scale,
146:             scrollX: 0,
147:             scrollY: currentY / scale
148:         });
149:

```

```

150:         // Add page to PDF
151:         if (pageNumber > 0) {
152:             pdf.addPage();
153:         }
154:
155:         // Calculate the actual content dimensions to maintain aspect ratio
156:         const canvasAspectRatio = canvas.width / canvas.height;
157:         let imgWidth = contentWidth;
158:         let imgHeight = contentWidth / canvasAspectRatio;
159:
160:         // If image height exceeds page, scale it down
161:         if (imgHeight > contentHeight) {
162:             imgHeight = contentHeight;
163:             imgWidth = contentHeight * canvasAspectRatio;
164:         }
165:
166:         // Center the image on the page
167:         const xOffset = margin + (contentWidth - imgWidth) / 2;
168:         const yOffset = margin + (contentHeight - imgHeight) / 2;
169:
170:         const imgData = canvas.toDataURL("image/jpeg", 0.95);
171:         pdf.addImage(imgData, "JPEG", xOffset, yOffset, imgWidth, imgHeight);
172:
173:         currentY += sliceHeight;
174:         pageNumber++;
175:
176:         // Add a small progress indicator
177:         toast({
178:             title: "Generating PDF",
179:             description: `Processing page ${pageNumber}...`,
180:         });
181:     }
182:
183:     // Save the PDF
184:     const fileName = `${patientCode} || "medical-report"}-${new Date().toISOString().split('T')[0]}.pdf`;
185:     pdf.save(fileName);
186:
187:     toast({
188:         title: "Success",
189:         description: `PDF generated successfully with ${pageNumber} pages`,
190:         variant: "default",
191:     });
192:
193: } catch (error) {
194:     console.error("PDF generation error:", error);
195:     toast({
196:         title: "Error",
197:         description: "Failed to generate PDF. Please try again.",
198:         variant: "destructive",
199:     });
200: } finally {
201:     setShowForCapture(false);
202:     setIsGenerating(false);
203: }
204: };
205:
206: return (
207:     <Card className="shadow-lg border-0 bg-white">
208:         <CardHeader className="bg-gradient-to-r from-red-500 to-pink-600 text-white rounded-lg p-4 sm:p-5">
209:             <CardTitle className="text-lg sm:text-xl md:text-2xl font-bold uppercase flex items-center justify-
210:                 PDF Generation
211:             </CardTitle>
212:         </CardHeader>
213:         <CardContent className="space-y-6 p-8">
214:             { /* Enhanced Offscreen Report for PDF Export */ }
215:             {showForCapture && (
216:                 <div
217:                     style={{
218:                         position: "fixed",
219:                         top: "-10000px",
220:                         left: "0",
221:                         width: "210mm",
222:                         maxWidth: "210mm",

```

```

223:         background: "white",
224:         zIndex: 9999,
225:         padding: "20px",
226:         fontFamily: "Arial, sans-serif",
227:         fontSize: "12px",
228:         lineHeight: "1.4",
229:         color: "#000000"
230:     }}
231: >
232:     <div
233:         ref={reportRef}
234:         className="report-container"
235:         style={{
236:             pageBreakInside: "avoid",
237:             orphans: 3,
238:             widows: 3
239:         }}
240:     >
241:         <ComprehensiveReportViewer reportData={reportData} />
242:     </div>
243: </div>
244: )}
245:
246: {/* Quick Actions */}
247: <div className="grid md:grid-cols-3 gap-4">
248:     <Button
249:         onClick={() =>
250:             window.open(
251:                 `/report/preview?patient=${encodeURIComponent(
252:                     patientCode
253:                 )}&report=${reportIndex}`,
254:                 "_blank"
255:             )
256:         }
257:         variant="outline"
258:         className="h-20 flex flex-col items-center justify-center gap-2 bg-blue-200 hover:bg-blue-300 t
259:         disabled={isGenerating}
260:     >
261:         <Eye className="h-6 w-6" />
262:         <span>Preview Report</span>
263:     </Button>
264:
265:     <Button
266:         onClick={handlePrint}
267:         disabled={isGenerating}
268:         className="h-20 flex flex-col items-center justify-center gap-2 bg-blue-600 hover:bg-blue-700 t
269:     >
270:         <Printer className="h-6 w-6" />
271:         <span>Print Report</span>
272:     </Button>
273:
274:     <Button
275:         onClick={handleMultiPagePDF}
276:         disabled={isGenerating}
277:         className="h-20 flex flex-col items-center justify-center gap-2 bg-green-600 hover:bg-green-700 t
278:     >
279:         <Download className={`h-6 w-6 ${isGenerating ? 'animate-spin' : ''}`} />
280:         <span>
281:             {isGenerating ? "Generating..." : "Download PDF"}
282:         </span>
283:     </Button>
284: </div>
285:
286: {/* Report Information */}
287: <div className="bg-gray-50 border border-gray-200 rounded-lg p-4">
288:     <h4 className="font-semibold text-red-800 mb-3">
289:         Report Information
290:     </h4>
291:     <div className="grid md:grid-cols-1 gap-4 text-sm">
292:         <div>
293:             <span className="font-medium">Patient Name:</span>{" "}
294:             {reportData.patientInfo.name || "Not specified"}
295:         </div>

```



```

296:         <div>
297:             <span className="font-medium">Sample Code:</span>{ " "}
298:             {reportData.patientInfo.sampleCode || "Not specified"}
299:         </div>
300:         <div>
301:             <span className="font-medium">Report Date:</span>{ " "}
302:             {reportData.patientInfo.reportDate || "Not specified"}
303:         </div>
304:     </div>
305: </div>
306:
307:     { /* PDF Generation Tips */ }
308:     <div className="bg-blue-50 border border-blue-200 rounded-lg p-4">
309:         <h4 className="font-semibold text-blue-800 mb-2">
310:             PDF Quality Features
311:         </h4>
312:         <ul className="text-sm text-blue-700 space-y-1">
313:             <li>? High-resolution output (3x scaling)</li>
314:             <li>? Smart page breaks to avoid cutting content</li>
315:             <li>? Proper margins and formatting</li>
316:             <li>? Multi-page support with progress tracking</li>
317:         </ul>
318:     </div>
319: </CardContent>
320: </Card>
321: );
322: }

```

## ■ File: components\admin\preventive-health-admin.tsx

```

1: "use client";
2:
3: import {
4:   Card,
5:   CardContent,
6:   CardDescription,
7:   CardHeader,
8:   CardTitle,
9: } from "@/components/ui/card";
10: import { Input } from "@/components/ui/input";
11: import { Label } from "@/components/ui/label";
12: import { Button } from "@/components/ui/button";
13: import { Textarea } from "@/components/ui/textarea";
14: import {
15:   Save,
16:   RotateCcw,
17:   Plus,
18:   Trash2,
19:   Stethoscope,
20:   TestTube,
21:   Pill,
22:   Calendar,
23:   Clock,
24:   FileText,
25:   Loader2,
26: } from "lucide-react";
27: import { Checkbox } from "@/components/ui/checkbox";
28: import { useState } from "react";
29: import type { PreventiveHealth } from "@/types/report-types";
30:
31: interface PreventiveHealthAdminProps {
32:   preventiveHealth: PreventiveHealth;
33:   updatePreventiveHealth: (section: keyof PreventiveHealth, data: any) => void;
34:   onSave: () => void;
35:   onReset: () => void;
36:   loading?: boolean;
37: }
38:
39: export default function PreventiveHealthAdmin({
40:   preventiveHealth,
41:   updatePreventiveHealth,

```

```

42:   onSave,
43:   onReset,
44:   loading = false,
45: }: PreventiveHealthAdminProps) {
46:   const [showTestModal, setShowTestModal] = useState(false);
47:   const [showSupplementModal, setShowSupplementModal] = useState(false);
48:   const [newTestName, setNewTestName] = useState("");
49:   const [newTestType, setNewTestType] = useState<"halfYearly" | "yearly">(
50:     "halfYearly"
51:   );
52:   const [newSupplementName, setNewSupplementName] = useState("");
53:
54:   // Ensure preventiveHealth has default values
55:   const safePreventiveHealth: PreventiveHealth = {
56:     description: preventiveHealth?.description || "",
57:     diagnosticTests: {
58:       halfYearly: preventiveHealth?.diagnosticTests?.halfYearly || [],
59:       yearly: preventiveHealth?.diagnosticTests?.yearly || [],
60:     },
61:     nutritionalSupplements: preventiveHealth?.nutritionalSupplements || [],
62:   };
63:
64:   const addDiagnosticTest = (type: "halfYearly" | "yearly", name?: string) => {
65:     updatePreventiveHealth("diagnosticTests", {
66:       ...safePreventiveHealth.diagnosticTests,
67:       [type]: [...safePreventiveHealth.diagnosticTests[type], name || ""],
68:     });
69:   };
70:
71:   const handleAddTest = () => {
72:     if (newTestName.trim()) {
73:       addDiagnosticTest(newTestType, newTestName.trim());
74:       setNewTestName("");
75:       setShowTestModal(false);
76:     }
77:   };
78:
79:   const handleAddSupplement = () => {
80:     if (newSupplementName.trim()) {
81:       updatePreventiveHealth("nutritionalSupplements", [
82:         ...safePreventiveHealth.nutritionalSupplements,
83:         {
84:           id: crypto.randomUUID(),
85:           supplement: newSupplementName.trim(),
86:           needed: true,
87:         },
88:       ]);
89:       setNewSupplementName("");
90:       setShowSupplementModal(false);
91:     }
92:   };
93:
94:   const updateDiagnosticTest = (
95:     type: "halfYearly" | "yearly",
96:     index: number,
97:     value: string
98:   ) => {
99:     const newTests = [...safePreventiveHealth.diagnosticTests[type]];
100:     newTests[index] = value;
101:     updatePreventiveHealth("diagnosticTests", {
102:       ...safePreventiveHealth.diagnosticTests,
103:       [type]: newTests,
104:     });
105:   };
106:
107:   const removeDiagnosticTest = (
108:     type: "halfYearly" | "yearly",
109:     index: number
110:   ) => {
111:     updatePreventiveHealth("diagnosticTests", {
112:       ...safePreventiveHealth.diagnosticTests,
113:       [type]: safePreventiveHealth.diagnosticTests[type].filter(
114:         (_, i) => i !== index

```

```

115:     ),
116:   });
117: };
118:
119: const updateNutritionalSupplement = (
120:   index: number,
121:   field: "supplement" | "needed",
122:   value: string | boolean
123: ) => {
124:   const newSupplements = [...safePreventiveHealth.nutritionalSupplements];
125:   newSupplements[index] = { ...newSupplements[index], [field]: value };
126:   updatePreventiveHealth("nutritionalSupplements", newSupplements);
127: };
128:
129: const removeNutritionalSupplement = (index: number) => {
130:   updatePreventiveHealth(
131:     "nutritionalSupplements",
132:     safePreventiveHealth.nutritionalSupplements.filter((_, i) => i !== index)
133:   );
134: };
135:
136: const handleKeyPress = (
137:   e: React.KeyboardEvent,
138:   action: "test" | "supplement"
139: ) => {
140:   if (e.key === "Enter") {
141:     if (action === "test") handleAddTest();
142:     else handleAddSupplement();
143:   } else if (e.key === "Escape") {
144:     if (action === "test") {
145:       setShowTestModal(false);
146:       setNewTestName("");
147:     } else {
148:       setShowSupplementModal(false);
149:       setNewSupplementName("");
150:     }
151:   }
152: };
153:
154: return (
155:   <div className="w-full max-w-8xl">
156:     <Card className="shadow-xl border-0 bg-gradient-to-br from-slate-50 to-teal-50/20 overflow-hidden">
157:       <CardHeader className="bg-gradient-to-r from-teal-600 via-cyan-600 to-blue-600 text-white rounded-l">
158:         <CardTitle className="text-lg sm:text-xl md:text-2xl font-bold uppercase flex items-center justify">
159:           Preventive Health Management
160:         </CardTitle>
161:       </CardHeader>
162:
163:       <div className="my-4 text-end">
164:         <Button onClick={onSave}>
165:           <Save className="h-4 w-4 mr-2" />
166:           Save Changes
167:         </Button>
168:       </div>
169:
170:       <CardContent className="mt-4">
171:         { /* Description Section */ }
172:         <div className="space-y-2 mb-10">
173:           <div className="flex items-center gap-3 mb-6">
174:             <h3 className="text-xl sm:text-2xl font-bold text-gray-800">
175:               Description
176:             </h3>
177:           </div>
178:
179:           <Card className="border-0 bg-white shadow-lg hover:shadow-xl transition-all duration-300">
180:             <CardContent className="p-6">
181:               <div>
182:                 <Input
183:                   value={preventiveHealth.description || ""}
184:                   onChange={e =>
185:                     updatePreventiveHealth("description", e.target.value)
186:                   }
187:                   placeholder="Enter a general description for preventive health recommendations..."

```

```

188:         className="border-2 border-indigo-200 focus:border-indigo-500 focus:ring-2 focus:ring-i
189:     />
190: </div>
191: </CardContent>
192: </Card>
193: </div>
194:
195: { /* Diagnostic Tests Section */}
196: <div className="space-y-6 mb-10">
197:     <div className="flex items-center gap-3 mb-6">
198:         <h3 className="text-xl sm:text-2xl font-bold text-gray-800">
199:             Diagnostic Tests
200:         </h3>
201:         <div className="flex-1 h-px bg-gradient-to-r from-blue-200 to-transparent"></div>
202:     </div>
203:
204:     <div className="grid grid-cols-1 lg:grid-cols-2 gap-6">
205:         { /* Half-Yearly Tests */}
206:         <Card className="border-0 bg-white shadow-lg hover:shadow-xl transition-all duration-300">
207:             <CardHeader className="bg-gradient-to-r from-blue-50 to-indigo-50 border-b border-blue-100">
208:                 <div className="flex justify-between items-start">
209:                     <div>
210:                         <CardTitle className="text-lg font-bold text-blue-800 flex items-center gap-2">
211:                             Half-Yearly Tests
212:                         </CardTitle>
213:                         <p className="text-sm text-blue-600 mt-1">
214:                             Every 6 months
215:                         </p>
216:                     </div>
217:                     <Button
218:                         onClick={() => {
219:                             setNewTestType("halfYearly");
220:                             setShowTestModal(true);
221:                         }}
222:                         size="sm"
223:                         className="bg-blue-500 hover:bg-blue-600 text-white shadow-md"
224:                     >
225:                         Add Test
226:                     </Button>
227:                 </div>
228:             </CardHeader>
229:             <CardContent className="p-4 space-y-3">
230:                 {preventiveHealth.diagnosticTests.halfYearly.map(
231:                     (test, index) => (
232:                         <div
233:                             key={index}
234:                             className="flex items-center gap-2 p-3 bg-blue-50/50 rounded-lg border border-blue-
235:                         >
236:                             <Input
237:                                 value={test}
238:                                 onChange={(e) =>
239:                                     updateDiagnosticTest(
240:                                         "halfYearly",
241:                                         index,
242:                                         e.target.value
243:                                     )
244:                                 }
245:                                 placeholder="Enter test name..."
246:                                 className="border-2 border-blue-200 focus:border-blue-500 focus:ring-2 focus:ring-
247:                             />
248:                             <Button
249:                                 onClick={() =>
250:                                     removeDiagnosticTest("halfYearly", index)
251:                                 }
252:                                 size="sm"
253:                                 variant="destructive"
254:                                 className="px-2 shadow-md hover:shadow-lg transition-all duration-200"
255:                             >
256:                                 <Trash2 className="h-3 w-3" />
257:                             </Button>
258:                         </div>
259:                     )
260:                 )}

```

```

261:         {preventiveHealth.diagnosticTests.halfYearly.length === 0 && (
262:             <div className="text-center py-8">
263:                 <p className="text-sm text-gray-500 mb-4">
264:                     No half-yearly tests configured
265:                 </p>
266:                 <Button
267:                     onClick={() => {
268:                         setNewTestType("halfYearly");
269:                         setShowTestModal(true);
270:                     }}
271:                     size="sm"
272:                     variant="outline"
273:                     className="border-blue-300 text-blue-600 hover:bg-blue-50"
274:                 >
275:                     Add First Test
276:                 </Button>
277:             </div>
278:         )}
279:     </CardContent>
280: </Card>
281:
282:     { /* Yearly Tests */ }
283:     <Card className="border-0 bg-white shadow-lg hover:shadow-xl transition-all duration-300">
284:         <CardHeader className="bg-gradient-to-r from-green-50 to-emerald-50 border-b border-green-100">
285:             <div className="flex justify-between items-start">
286:                 <div>
287:                     <CardTitle className="text-lg font-bold text-green-800 flex items-center gap-2">
288:                         Yearly Tests
289:                     </CardTitle>
290:                     <p className="text-sm text-green-600 mt-1">
291:                         Annual screening
292:                     </p>
293:                 </div>
294:                 <Button
295:                     onClick={() => {
296:                         setNewTestType("yearly");
297:                         setShowTestModal(true);
298:                     }}
299:                     size="sm"
300:                     className="bg-green-500 hover:bg-green-600 text-white shadow-md"
301:                 >
302:                     Add Test
303:                 </Button>
304:             </div>
305:         </CardHeader>
306:         <CardContent className="p-4 space-y-3">
307:             {preventiveHealth.diagnosticTests.yearly.map(
308:                 (test, index) => (
309:                     <div
310:                         key={index}
311:                         className="flex items-center gap-2 p-3 bg-green-50/50 rounded-lg border border-green-100"
312:                     >
313:                         <Input
314:                             value={test}
315:                             onChange={(e) =>
316:                                 updateDiagnosticTest(
317:                                     "yearly",
318:                                     index,
319:                                     e.target.value
320:                                 )
321:                             }
322:                             placeholder="Enter test name..."
323:                             className="border-2 border-green-200 focus:border-green-500 focus:ring-2 focus:ring-green-500"
324:                         />
325:                         <Button
326:                             onClick={() => removeDiagnosticTest("yearly", index)}
327:                             size="sm"
328:                             variant="destructive"
329:                             className="px-2 shadow-md hover:shadow-lg transition-all duration-200"
330:                         >
331:                             <Trash2 className="h-3 w-3" />
332:                         </Button>
333:                     </div>

```

```

334:     )
335:   }}
336:   {preventiveHealth.diagnosticTests.yearly.length === 0 && (
337:     <div className="text-center py-8">
338:       <p className="text-sm text-gray-500 mb-4">
339:         No yearly tests configured
340:       </p>
341:       <Button
342:         onClick={() => {
343:           setNewTestType("yearly");
344:           setShowTestModal(true);
345:         }}
346:         size="sm"
347:         variant="outline"
348:         className="border-green-300 text-green-600 hover:bg-green-50"
349:       >
350:         Add First Test
351:       </Button>
352:     </div>
353:   )}
354: </CardContent>
355: </Card>
356: </div>
357: </div>
358:
359: { /* Nutritional Supplements Section */ }
360: <div className="space-y-6">
361:   <div className="flex items-center gap-3 mb-6">
362:     <h3 className="text-xl sm:text-2xl font-bold text-gray-800">
363:       Nutritional Supplements
364:     </h3>
365:     <div className="flex-1 h-px bg-gradient-to-r from-purple-200 to-transparent"></div>
366:   </div>
367:
368:   <Card className="border-0 bg-white shadow-lg hover:shadow-xl transition-all duration-300">
369:     <CardHeader className="bg-gradient-to-r from-purple-50 to-pink-50 border-b border-purple-100">
370:       <div className="flex justify-between items-start">
371:         <div>
372:           <CardTitle className="text-lg font-bold text-purple-800 flex items-center gap-2">
373:             Recommended Supplements
374:           </CardTitle>
375:         </div>
376:         <Button
377:           onClick={() => setShowSupplementModal(true)}
378:           size="sm"
379:           className="bg-purple-500 hover:bg-purple-600 text-white shadow-md"
380:         >
381:           Add Supplement
382:         </Button>
383:       </div>
384:     </CardHeader>
385:     <CardContent className="p-4 space-y-3">
386:       {preventiveHealth.nutritionalSupplements.map(
387:         (supplement, index) => (
388:           <div
389:             key={index}
390:             className="flex items-center gap-3 p-4 bg-purple-50/50 rounded-lg border border-purple-100"
391:           >
392:             <Input
393:               value={supplement.supplement}
394:               onChange={(e) =>
395:                 updateNutritionalSupplement(
396:                   index,
397:                   "supplement",
398:                   e.target.value
399:                 )
400:             }
401:             placeholder="Enter supplement name..."
402:             className="flex-1 border-2 border-purple-200 focus:border-purple-500 focus:ring-2 focus:ring-purple-500"
403:           />
404:             <div className="flex items-center space-x-2 bg-white px-3 py-2 rounded-lg border border-purple-100">
405:               <Checkbox
406:                 id={`needed-${index}`}

```

```

407:                 checked={supplement.needed}
408:                 onCheckedChange={({checked: boolean}) =>
409:                     updateNutritionalSupplement(
410:                         index,
411:                         "needed",
412:                         checked
413:                     )
414:                 }
415:                 className="border-purple-300"
416:             />
417:             <Label
418:                 htmlFor={`needed-${index}`}
419:                 className="text-sm font-medium text-gray-700"
420:             >
421:                 Required
422:             </Label>
423:         </div>
424:         <Button
425:             onClick={() => removeNutritionalSupplement(index)}
426:             size="sm"
427:             variant="destructive"
428:             className="px-2 shadow-md hover:shadow-lg transition-all duration-200"
429:         >
430:             <Trash2 className="h-3 w-3" />
431:         </Button>
432:     </div>
433: )
434: }}
435: {preventiveHealth.nutritionalSupplements.length === 0 && (
436:     <div className="text-center py-12">
437:         <Pill className="h-16 w-16 text-purple-300 mx-auto mb-4" />
438:         <p className="text-lg text-gray-600 mb-2">
439:             No supplements configured
440:         </p>
441:         <p className="text-sm text-gray-500 mb-6">
442:             Add nutritional supplements based on genetic
443:             recommendations
444:         </p>
445:         <Button
446:             onClick={() => setShowSupplementModal(true)}
447:             className="bg-gradient-to-r from-purple-600 to-pink-600 hover:from-purple-700 hover:t
448:         >
449:             <Plus className="h-4 w-4 mr-2" />
450:             Add First Supplement
451:         </Button>
452:     </div>
453: )}
454: </CardContent>
455: </Card>
456: </div>
457: </CardContent>
458: </Card>
459:
460: {/* Add Diagnostic Test Modal */}
461: {showTestModal && (
462:     <div className="fixed inset-0 bg-black/50 backdrop-blur-sm flex items-center justify-center z-50 p-
463:         <div className="bg-white rounded-xl shadow-2xl w-full max-w-md mx-auto">
464:             <div className="p-6">
465:                 <div className="flex items-center justify-between mb-6">
466:                     <h3 className="text-xl font-bold text-gray-800 flex items-center gap-2">
467:                         <div className="p-2 bg-blue-100 rounded-lg">
468:                             <TestTube className="h-5 w-5 text-blue-600" />
469:                         </div>
470:                         Add Diagnostic Test
471:                     </h3>
472:                     <Button
473:                         onClick={() => {
474:                             setShowTestModal(false);
475:                             setNewTestName("");
476:                         }}
477:                         variant="ghost"
478:                         size="sm"
479:                         className="text-gray-500 hover:text-gray-700"

```

```

480:         >
481:         <Trash2 className="h-5 w-5" />
482:     </Button>
483: </div>
484:
485: <div className="space-y-4">
486:     <div>
487:         <Label className="text-sm font-semibold text-gray-700 mb-2 block">
488:             Test Type
489:         </Label>
490:         <div className="grid grid-cols-2 gap-2">
491:             <Button
492:                 onClick={() => setNewTestType("halfYearly")}
493:                 variant={
494:                     newTestType === "halfYearly" ? "default" : "outline"
495:                 }
496:                 className="text-sm"
497:             >
498:                 <Clock className="h-4 w-4 mr-1" />
499:                 Half-Yearly
500:             </Button>
501:             <Button
502:                 onClick={() => setNewTestType("yearly")}
503:                 variant={newTestType === "yearly" ? "default" : "outline"}
504:                 className="text-sm"
505:             >
506:                 <Calendar className="h-4 w-4 mr-1" />
507:                 Yearly
508:             </Button>
509:         </div>
510:     </div>
511:
512:     <div>
513:         <Label className="text-sm font-semibold text-gray-700 mb-2 block">
514:             Test Name
515:         </Label>
516:         <Input
517:             value={newTestName}
518:             onChange={(e) => setNewTestName(e.target.value)}
519:             onKeyDown={(e) => handleKeyPress(e, "test")}
520:             placeholder="Enter diagnostic test name..."
521:             className="border-2 border-gray-200 focus:border-blue-500 focus:ring-2 focus:ring-blue-500"
522:             autoFocus
523:         />
524:     </div>
525: </div>
526:
527: <div className="flex gap-3 mt-8">
528:     <Button
529:         onClick={() => {
530:             setShowTestModal(false);
531:             setNewTestName("");
532:         }}
533:         variant="outline"
534:         className="flex-1"
535:     >
536:         Cancel
537:     </Button>
538:     <Button
539:         onClick={handleAddTest}
540:         disabled={!newTestName.trim()}
541:         className="flex-1 bg-gradient-to-r from-blue-600 to-indigo-600 hover:from-blue-700 hover:to-indigo-700"
542:     >
543:         <Plus className="h-4 w-4 mr-2" />
544:         Add Test
545:     </Button>
546: </div>
547: </div>
548: </div>
549: </div>
550: )}
551:
552: { /* Add Supplement Modal */}

```



```

553:     {showSupplementModal && (
554:       <div className="fixed inset-0 bg-black/50 backdrop-blur-sm flex items-center justify-center z-50 p-
555:         <div className="bg-white rounded-xl shadow-2xl w-full max-w-md mx-auto">
556:           <div className="p-6">
557:             <div className="flex items-center justify-between mb-6">
558:               <h3 className="text-xl font-bold text-gray-800 flex items-center gap-2">
559:                 <div className="p-2 bg-purple-100 rounded-lg">
560:                   <Pill className="h-5 w-5 text-purple-600" />
561:                 </div>
562:                 Add Supplement
563:               </h3>
564:               <Button
565:                 onClick={() => {
566:                   setShowSupplementModal(false);
567:                   setNewSupplementName("");
568:                 }}
569:                 variant="ghost"
570:                 size="sm"
571:                 className="text-gray-500 hover:text-gray-700"
572:               >
573:                 <Trash2 className="h-5 w-5" />
574:               </Button>
575:             </div>
576:
577:             <div className="space-y-4">
578:               <div>
579:                 <Label className="text-sm font-semibold text-gray-700 mb-2 block">
580:                   Supplement Name
581:                 </Label>
582:                 <Input
583:                   value={newSupplementName}
584:                   onChange={(e) => setNewSupplementName(e.target.value)}
585:                   onKeyDown={(e) => handleKeyPress(e, "supplement")}
586:                   placeholder="Enter supplement name..."
587:                   className="border-2 border-gray-200 focus:border-purple-500 focus:ring-2 focus:ring-purp
588:                   autoFocus
589:                 />
590:                 <p className="text-xs text-gray-500 mt-1">
591:                   The supplement will be marked as required by default
592:                 </p>
593:               </div>
594:             </div>
595:
596:             <div className="flex gap-3 mt-8">
597:               <Button
598:                 onClick={() => {
599:                   setShowSupplementModal(false);
600:                   setNewSupplementName("");
601:                 }}
602:                 variant="outline"
603:                 className="flex-1"
604:               >
605:                 Cancel
606:               </Button>
607:               <Button
608:                 onClick={handleAddSupplement}
609:                 disabled={!newSupplementName.trim()}
610:                 className="flex-1 bg-gradient-to-r from-purple-600 to-pink-600 hover:from-purple-700 hove
611:               >
612:                 <Plus className="h-4 w-4 mr-2" />
613:                 Add Supplement
614:               </Button>
615:             </div>
616:           </div>
617:         </div>
618:       </div>
619:     )}
620:   </div>
621: );
622: }

```

## ■ File: components\admin\report-settings-admin.tsx

```
=====
1: "use client";
2:
3: import {
4:   Card,
5:   CardContent,
6:   CardDescription,
7:   CardHeader,
8:   CardTitle,
9: } from "@/components/ui/card";
10: import { Input } from "@/components/ui/input";
11: import { Label } from "@/components/ui/label";
12: import { Button } from "@/components/ui/button";
13: import { Save, RotateCcw } from "lucide-react";
14: import type { ReportSettings } from "@/types/report-types";
15:
16: interface ReportSettingsAdminProps {
17:   settings: ReportSettings;
18:   updateSettings: (field: keyof ReportSettings, value: string) => void;
19:   onSave: () => void;
20:   onReset: () => void;
21: }
22:
23: export default function ReportSettingsAdmin({
24:   settings,
25:   updateSettings,
26:   onSave,
27:   onReset,
28: }: ReportSettingsAdminProps) {
29:   return (
30:     <Card className="shadow-lg border-0 bg-white">
31:       <CardHeader className="bg-gradient-to-r from-purple-500 to-pink-600 text-white rounded-lg">
32:         <CardTitle className="text-2xl">Report Settings</CardTitle>
33:         <CardDescription className="text-purple-100">
34:           Customize the appearance and branding of your report
35:         </CardDescription>
36:       </CardHeader>
37:       <CardContent className="space-y-6 p-8">
38:         </* Action Buttons */>
39:         <div className="flex justify-end gap-2 mb-6">
40:           <Button onClick={onSave} size="sm">
41:             <Save className="h-4 w-4 mr-2" />
42:             Save Changes
43:           </Button>
44:         </div>
45:
46:         <div className="grid md:grid-cols-2 gap-6">
47:           <div className="space-y-2">
48:             <Label
49:               htmlFor="title"
50:               className="text-sm font-semibold text-gray-700"
51:             >
52:               Report Title
53:             </Label>
54:             <Input
55:               id="title"
56:               value={settings.title}
57:               onChange={(e) => updateSettings("title", e.target.value)}
58:               className="border-2 focus:border-purple-500 font-bold"
59:               placeholder="e.g., Comprehensive Health Report"
60:             />
61:           </div>
62:           <div className="space-y-2">
63:             <Label
64:               htmlFor="subtitle"
65:               className="text-sm font-semibold text-gray-700"
66:             >
67:               Report Subti
68:             </Label>
69:             <Input
70:               id="subtitle"
71:               value={settings.subtitle}
```

```

72:         onChange={(e) => updateSettings("subtitle", e.target.value)}
73:         className="border-2 focus:border-purple-500"
74:         placeholder="e.g., A Detailed Analysis of Your Genetics"
75:     />
76: </div>
77: <div className="space-y-2">
78:     <Label
79:         htmlFor="companyName"
80:         className="text-sm font-semibold text-gray-700"
81:     >
82:         Company Name
83:     </Label>
84:     <Input
85:         id="companyName"
86:         value={settings.companyName}
87:         onChange={(e) => updateSettings("companyName", e.target.value)}
88:         className="border-2 focus:border-purple-500"
89:         placeholder="e.g., Health Solutions Inc."
90:     />
91: </div>
92: <div className="space-y-2">
93:     <Label
94:         htmlFor="headerColor"
95:         className="text-sm font-semibold text-gray-700"
96:     >
97:         Header Color
98:     </Label>
99:     <div className="flex gap-2">
100:         <Input
101:             id="headerColor"
102:             type="color"
103:             value={settings.headerColor}
104:             onChange={(e) => updateSettings("headerColor", e.target.value)}
105:             className="w-16 h-10 border-2 focus:border-purple-500"
106:         />
107:         <Input
108:             value={settings.headerColor}
109:             onChange={(e) => updateSettings("headerColor", e.target.value)}
110:             className="flex-1 border-2 focus:border-purple-500 font-mono"
111:             placeholder="#2C3E50"
112:         />
113:     </div>
114: </div>
115: <div className="space-y-2">
116:     <Label
117:         htmlFor="accentColor"
118:         className="text-sm font-semibold text-gray-700"
119:     >
120:         Accent Color
121:     </Label>
122:     <div className="flex gap-2">
123:         <Input
124:             id="accentColor"
125:             type="color"
126:             value={settings.accentColor}
127:             onChange={(e) => updateSettings("accentColor", e.target.value)}
128:             className="w-16 h-10 border-2 focus:border-purple-500"
129:         />
130:         <Input
131:             value={settings.accentColor}
132:             onChange={(e) => updateSettings("accentColor", e.target.value)}
133:             className="flex-1 border-2 focus:border-purple-500 font-mono"
134:             placeholder="#E74C3C"
135:         />
136:     </div>
137: </div>
138: </div>
139:
140: <div className="space-y-4">
141:     <h3 className="text-lg font-semibold text-gray-800 border-b pb-2">
142:         Font Settings
143:     </h3>
144:     <div className="grid md:grid-cols-3 gap-6">

```

```

145:     <div className="space-y-2">
146:       <Label
147:         htmlFor="primaryFont"
148:         className="text-sm font-semibold text-gray-700"
149:       >
150:         Primary Font
151:       </Label>
152:       <Input
153:         id="primaryFont"
154:         value={settings.fonts.primary}
155:         onChange={(e) =>
156:           updateSettings(
157:             "fonts",
158:             JSON.stringify({
159:               ...settings.fonts,
160:               primary: e.target.value,
161:             })
162:           )
163:         }
164:         className="border-2 focus:border-purple-500"
165:         placeholder="e.g., Helvetica, Arial, sans-serif"
166:       />
167:     </div>
168:     <div className="space-y-2">
169:       <Label
170:         htmlFor="secondaryFont"
171:         className="text-sm font-semibold text-gray-700"
172:       >
173:         Secondary Font
174:       </Label>
175:       <Input
176:         id="secondaryFont"
177:         value={settings.fonts.secondary}
178:         onChange={(e) =>
179:           updateSettings(
180:             "fonts",
181:             JSON.stringify({
182:               ...settings.fonts,
183:               secondary: e.target.value,
184:             })
185:           )
186:         }
187:         className="border-2 focus:border-purple-500"
188:         placeholder="e.g., Georgia, serif"
189:       />
190:     </div>
191:     <div className="space-y-2">
192:       <Label
193:         htmlFor="monoFont"
194:         className="text-sm font-semibold text-gray-700"
195:       >
196:         Monospace Font
197:       </Label>
198:       <Input
199:         id="monoFont"
200:         value={settings.fonts.mono}
201:         onChange={(e) =>
202:           updateSettings(
203:             "fonts",
204:             JSON.stringify({ ...settings.fonts, mono: e.target.value })
205:           )
206:         }
207:         className="border-2 focus:border-purple-500"
208:         placeholder="e.g., Courier New, monospace"
209:       />
210:     </div>
211:   </div>
212: </div>
213:
214: <div className="bg-purple-50 border-2 border-purple-200 rounded-lg p-4">
215:   <h4 className="font-semibold text-purple-800 mb-2">
216:     ? Customization Tips:
217:   </h4>

```

```

218:         <ul className="text-sm text-purple-700 space-y-1">
219:             <li>? Use hex codes for precise color matching (e.g., #RRGGBB).</li>
220:             <li>? Choose fonts that align with your brand's aesthetic.</li>
221:             <li>
222:                 ? Ensure sufficient contrast for readability, especially for text.
223:             </li>
224:         </ul>
225:     </div>
226: </CardContent>
227: </Card>
228: );
229: }

```

## ■ File: components\admin\report-summaries-admin.tsx

```

=====
1: "use client"
2:
3: import { Card, CardContent, CardDescription, CardHeader, CardTitle } from "@/components/ui/card"
4: import { Label } from "@/components/ui/label"
5: import { Textarea } from "@/components/ui/textarea"
6: import { Button } from "@/components/ui/button"
7: import { Save, RotateCcw } from "lucide-react"
8: import type { ReportSummaries } from "@/types/report-types"
9:
10: interface ReportSummariesAdminProps {
11:   summaries: ReportSummaries
12:   updateSummaries: (field: keyof ReportSummaries, value: string) => void
13:   onSave: () => void
14:   onReset: () => void
15: }
16:
17: export default function ReportSummariesAdmin({
18:   summaries,
19:   updateSummaries,
20:   onSave,
21:   onReset,
22: }: ReportSummariesAdminProps) {
23:   return (
24:     <Card className="shadow-lg border-0 bg-white">
25:       <CardHeader className="bg-gradient-to-r from-violet-500 to-purple-600 text-white rounded-t-lg">
26:         <CardTitle className="text-2xl">? Report Summaries</CardTitle>
27:         <CardDescription className="text-violet-100">Edit the summary sections for experts</CardDescription>
28:       </CardHeader>
29:       <CardContent className="space-y-8 p-8">
30:         <div>
31:           <div className="flex justify-end gap-2 mb-6">
32:             <Button onClick={onReset} variant="outline" size="sm">
33:               <RotateCcw className="h-4 w-4 mr-2" />
34:               Reset
35:             </Button>
36:             <Button onClick={onSave} size="sm">
37:               <Save className="h-4 w-4 mr-2" />
38:               Save Changes
39:             </Button>
40:           </div>
41:
42:           <div className="grid md:grid-cols-2 gap-8">
43:             <div className="space-y-3">
44:               <Label htmlFor="nutrigenomicsSummary" className="text-lg font-semibold text-gray-800">
45:                 ? Nutrigenomics Summary
46:               </Label>
47:               <p className="text-sm text-gray-600">For the nutrigenomics expert</p>
48:               <Textarea
49:                 id="nutrigenomicsSummary"
50:                 value={summaries.nutrigenomicsSummary}
51:                 onChange={(e) => updateSummaries("nutrigenomicsSummary", e.target.value)}
52:                 rows={8}
53:                 className="border-2 focus:border-violet-500"
54:                 placeholder="Based on the overall nutrigenomics analysis, provide expert summary for personal
55:               />
56:             </div>

```

```

57:         <div className="space-y-3">
58:             <Label htmlFor="exerciseGenomicsSummary" className="text-lg font-semibold text-gray-800">
59:                 ??? Exercise Genomics Summary
60:             </Label>
61:             <p className="text-sm text-gray-600">For the fitness genomics expert</p>
62:             <Textarea
63:                 id="exerciseGenomicsSummary"
64:                 value={summaries.exerciseGenomicsSummary}
65:                 onChange={(e) => updateSummaries("exerciseGenomicsSummary", e.target.value)}
66:                 rows={8}
67:                 className="border-2 focus:border-violet-500"
68:                 placeholder="Based on the overall genomics analysis, provide expert summary for personalized
69:             />
70:         </div>
71:     </div>
72:
73:     <div className="bg-blue-50 border-2 border-blue-200 rounded-lg p-6">
74:         <h4 className="font-semibold text-blue-800 mb-2">? Expert Summary Guidelines:</h4>
75:         <ul className="text-sm text-blue-700 space-y-1">
76:             <li>? Provide actionable insights based on genetic analysis</li>
77:             <li>? Include specific recommendations for healthcare professionals</li>
78:             <li>? Reference key genetic markers and their implications</li>
79:             <li>? Suggest monitoring parameters and intervention strategies</li>
80:             <li>? Keep language professional yet accessible</li>
81:         </ul>
82:     </div>
83: </CardContent>
84: </Card>
85: )
86: }

```

## File: components\admin\settings-admin.tsx

```

1: "use client";
2:
3: import React from "react";
4: import {
5:   Card,
6:   CardContent,
7:   CardDescription,
8:   CardHeader,
9:   CardTitle,
10: } from "@/components/ui/card";
11: import { Input } from "@/components/ui/input";
12: import { Label } from "@/components/ui/label";
13: import { Button } from "@/components/ui/button";
14: import { Save, RotateCcw } from "lucide-react";
15:
16: // Define types locally since we don't have access to the external types
17: interface FontSettings {
18:   primary: string;
19:   secondary: string;
20:   mono: string;
21: }
22:
23: interface ReportSettings {
24:   title: string;
25:   subtitle: string;
26:   companyName: string;
27:   headerColor: string;
28:   accentColor: string;
29:   fonts: FontSettings;
30: }
31:
32: interface SettingsAdminProps {
33:   settings: ReportSettings;
34:   updateSettings: (
35:     field: keyof ReportSettings,
36:     value: string | FontSettings
37:   ) => void;
38:   onSave: () => void;

```

```

39:   onReset: () => void;
40: }
41:
42: export default function SettingsAdmin({
43:   settings,
44:   updateSettings,
45:   onSave,
46:   onReset,
47: }: SettingsAdminProps) {
48:   // Helper function to update font settings
49:   const updateFontSetting = (fontType: keyof FontSettings, value: string) => {
50:     const updatedFonts = {
51:       ...settings.fonts,
52:       [fontType]: value,
53:     };
54:     updateSettings("fonts", updatedFonts);
55:   };
56:
57:   return (
58:     <Card className="shadow-lg border-0 bg-white">
59:       <CardHeader className="bg-gradient-to-r from-purple-500 to-pink-600 text-white rounded-lg px-3 py-4">
60:         <CardTitle className="text-lg sm:text-xl md:text-2xl lg:text-3xl font-semibold uppercase text-center">
61:           Report Settings
62:         </CardTitle>
63:       </CardHeader>
64:
65:       <CardContent className="space-y-6 p-8">
66:         { /* Action Buttons */ }
67:         <div className="flex justify-end gap-2 mb-6">
68:           <Button onClick={onSave} size="sm">
69:             <Save className="h-4 w-4 mr-2" />
70:             Save Changes
71:           </Button>
72:         </div>
73:
74:         <div className="grid md:grid-cols-2 gap-6">
75:           <div className="space-y-2">
76:             <Label
77:               htmlFor="title"
78:               className="text-sm font-semibold text-gray-700"
79:             >
80:               Report Title
81:             </Label>
82:             <Input
83:               id="title"
84:               value={settings.title}
85:               onChange={(e) => updateSettings("title", e.target.value)}
86:               className="border-2 focus:border-purple-500 font-bold"
87:               placeholder="Enter report title"
88:             />
89:           </div>
90:           <div className="space-y-2">
91:             <Label
92:               htmlFor="subtitle"
93:               className="text-sm font-semibold text-gray-700"
94:             >
95:               Report Subtitle
96:             </Label>
97:             <Input
98:               id="subtitle"
99:               value={settings.subtitle}
100:              onChange={(e) => updateSettings("subtitle", e.target.value)}
101:              className="border-2 focus:border-purple-500"
102:              placeholder="Enter report subtitle"
103:            />
104:          </div>
105:          <div className="space-y-2">
106:            <Label
107:              htmlFor="companyName"
108:              className="text-sm font-semibold text-gray-700"
109:            >
110:              Company Name
111:            </Label>

```

```

112:         <Input
113:             id="companyName"
114:             value={settings.companyName}
115:             onChange={(e) => updateSettings("companyName", e.target.value)}
116:             className="border-2 focus:border-purple-500"
117:             placeholder="Enter company name"
118:         />
119:     </div>
120:     <div className="space-y-2">
121:         <Label
122:             htmlFor="headerColor"
123:             className="text-sm font-semibold text-gray-700"
124:         >
125:             Header Color
126:         </Label>
127:         <div className="flex gap-2">
128:             <Input
129:                 id="headerColor"
130:                 type="color"
131:                 value={settings.headerColor}
132:                 onChange={(e) => updateSettings("headerColor", e.target.value)}
133:                 className="w-16 h-10 border-2 focus:border-purple-500"
134:             />
135:             <Input
136:                 value={settings.headerColor}
137:                 onChange={(e) => updateSettings("headerColor", e.target.value)}
138:                 className="flex-1 border-2 focus:border-purple-500 font-mono"
139:                 placeholder="#000000"
140:             />
141:         </div>
142:     </div>
143:     <div className="space-y-2">
144:         <Label
145:             htmlFor="accentColor"
146:             className="text-sm font-semibold text-gray-700"
147:         >
148:             Accent Color
149:         </Label>
150:         <div className="flex gap-2">
151:             <Input
152:                 id="accentColor"
153:                 type="color"
154:                 value={settings.accentColor}
155:                 onChange={(e) => updateSettings("accentColor", e.target.value)}
156:                 className="w-16 h-10 border-2 focus:border-purple-500"
157:             />
158:             <Input
159:                 value={settings.accentColor}
160:                 onChange={(e) => updateSettings("accentColor", e.target.value)}
161:                 className="flex-1 border-2 focus:border-purple-500 font-mono"
162:                 placeholder="#000000"
163:             />
164:         </div>
165:     </div>
166:     <div className="space-y-2">
167:         <Label
168:             htmlFor="primaryFont"
169:             className="text-sm font-semibold text-gray-700"
170:         >
171:             Primary Font
172:         </Label>
173:         <Input
174:             id="primaryFont"
175:             value={settings.fonts.primary}
176:             onChange={(e) => updateFontSetting("primary", e.target.value)}
177:             placeholder="e.g., Helvetica, Arial, sans-serif"
178:             className="border-2 focus:border-purple-500"
179:         />
180:     </div>
181:     <div className="space-y-2">
182:         <Label
183:             htmlFor="secondaryFont"
184:             className="text-sm font-semibold text-gray-700"

```



```

185:         >
186:         Secondary Font
187:     </Label>
188:     <Input
189:         id="secondaryFont"
190:         value={settings.fonts.secondary}
191:         onChange={(e) => updateFontSetting("secondary", e.target.value)}
192:         placeholder="e.g., Georgia, serif"
193:         className="border-2 focus:border-purple-500"
194:     />
195: </div>
196: <div className="space-y-2">
197:     <Label
198:         htmlFor="monoFont"
199:         className="text-sm font-semibold text-gray-700"
200:     >
201:         Monospace Font
202:     </Label>
203:     <Input
204:         id="monoFont"
205:         value={settings.fonts.mono}
206:         onChange={(e) => updateFontSetting("mono", e.target.value)}
207:         placeholder="e.g., Courier New, monospace"
208:         className="border-2 focus:border-purple-500"
209:     />
210: </div>
211: </div>
212: </CardContent>
213: </Card>
214: );
215: }

```

## ■ File: components\admin\share-management.tsx

```

=====
1: "use client";
2:
3: import { useState, useEffect } from "react";
4: import { Button } from "@/components/ui/button";
5: import { Input } from "@/components/ui/input";
6: import { Label } from "@/components/ui/label";
7: import { Card, CardContent, CardHeader, CardTitle } from "@/components/ui/card";
8: import { useToast } from "@/components/ui/use-toast";
9: import {
10:   Dialog,
11:   DialogContent,
12:   DialogHeader,
13:   DialogTitle,
14:   DialogTrigger,
15: } from "@/components/ui/dialog";
16: import {
17:   Select,
18:   SelectContent,
19:   SelectItem,
20:   SelectTrigger,
21:   SelectValue,
22: } from "@/components/ui/select";
23: import {
24:   Share2,
25:   Copy,
26:   Eye,
27:   Calendar,
28:   Lock,
29:   Trash2,
30:   ExternalLink,
31:   CheckCircle2,
32:   XCircle,
33: } from "lucide-react";
34:
35: interface ShareManagementProps {
36:   reportId: string;
37:   patientId: string;

```

```

38:   reportName?: string;
39: }
40:
41: interface ShareToken {
42:   id: string;
43:   token: string;
44:   url: string;
45:   expiresAt: Date | null;
46:   maxViews: number | null;
47:   viewCount: number;
48:   hasPassword: boolean;
49:   createdAt: Date;
50:   lastAccessedAt: Date | null;
51:   isActive: boolean;
52:   reportName: string;
53:   isExpired: boolean;
54:   isMaxViewsReached: boolean;
55: }
56:
57: export default function ShareManagement({
58:   reportId,
59:   patientId,
60:   reportName,
61: }: ShareManagementProps) {
62:   const { toast } = useToast();
63:   const [isDialogOpen, setIsDialogOpen] = useState(false);
64:   const [isLoading, setIsLoading] = useState(false);
65:   const [shareTokens, setShareTokens] = useState<ShareToken[]>([]);
66:   const [copiedToken, setCopiedToken] = useState<string | null>(null);
67:
68:   // Form state
69:   const [expirationOption, setExpirationOption] = useState<string>("7");
70:   const [customDays, setCustomDays] = useState("");
71:   const [maxViews, setMaxViews] = useState("");
72:   const [password, setPassword] = useState("");
73:   const [usePassword, setUsePassword] = useState(false);
74:
75:   // Load existing share tokens
76:   const loadShareTokens = async () => {
77:     try {
78:       const response = await fetch(`/api/share-report?reportId=${reportId}`);
79:       if (response.ok) {
80:         const data = await response.json();
81:         setShareTokens(data.shareTokens || []);
82:       }
83:     } catch (error) {
84:       console.error("Failed to load share tokens:", error);
85:     }
86:   };
87:
88:   useEffect(() => {
89:     loadShareTokens();
90:   }, [reportId]);
91:
92:   const handleCreateShare = async () => {
93:     setIsLoading(true);
94:
95:     try {
96:       // Calculate expiration days
97:       let expiresInDays: number | null = null;
98:       if (expirationOption === "custom" && customDays) {
99:         expiresInDays = parseInt(customDays);
100:       } else if (expirationOption !== "never") {
101:         expiresInDays = parseInt(expirationOption);
102:       }
103:
104:       const response = await fetch("/api/share-report", {
105:         method: "POST",
106:         headers: { "Content-Type": "application/json" },
107:         body: JSON.stringify({
108:           reportId,
109:           patientId,
110:           password: usePassword && password ? password : null,

```

```

111:         expiresInDays,
112:         maxViews: maxViews ? parseInt(maxViews) : null,
113:         createdBy: "admin", // You can replace with actual user info
114:     }},
115: });
116:
117: if (!response.ok) {
118:     throw new Error("Failed to create share link");
119: }
120:
121: const data = await response.json();
122:
123: toast({
124:     title: "Share Link Created!",
125:     description: "The secure share link has been generated.",
126:     variant: "success",
127:     duration: 3000,
128: });
129:
130: // Copy to clipboard
131: await navigator.clipboard.writeText(data.shareToken.url);
132: setCopiedToken(data.shareToken.token);
133:
134: // Reset form
135: setExpirationOption("7");
136: setCustomDays("");
137: setMaxViews("");
138: setPassword("");
139: setUsePassword(false);
140: setIsDialogOpen(false);
141:
142: // Reload tokens
143: loadShareTokens();
144: } catch (error: any) {
145:     toast({
146:         title: "Error",
147:         description: error.message || "Failed to create share link",
148:         variant: "destructive",
149:         duration: 3000,
150:     });
151: } finally {
152:     setIsLoading(false);
153: }
154: };
155:
156: const handleCopyLink = async (url: string, token: string) => {
157:     try {
158:         await navigator.clipboard.writeText(url);
159:         setCopiedToken(token);
160:         toast({
161:             description: "Link copied to clipboard!",
162:             variant: "success",
163:             duration: 2000,
164:         });
165:
166:         setTimeout(() => setCopiedToken(null), 2000);
167:     } catch (error) {
168:         toast({
169:             description: "Failed to copy link",
170:             variant: "destructive",
171:             duration: 2000,
172:         });
173:     }
174: };
175:
176: const handleRevokeToken = async (tokenId: string) => {
177:     if (
178:         !confirm(
179:             "Are you sure you want to revoke this share link? It will no longer be accessible."
180:         )
181:     ) {
182:         return;
183:     }

```

```

184:
185:     try {
186:         const response = await fetch(`/api/share-report?tokenId=${tokenId}`, {
187:             method: "DELETE",
188:         });
189:
190:         if (!response.ok) {
191:             throw new Error("Failed to revoke token");
192:         }
193:
194:         toast({
195:             title: "Token Revoked",
196:             description: "The share link has been revoked.",
197:             variant: "success",
198:             duration: 2000,
199:         });
200:
201:         loadShareTokens();
202:     } catch (error: any) {
203:         toast({
204:             title: "Error",
205:             description: error.message || "Failed to revoke token",
206:             variant: "destructive",
207:             duration: 2000,
208:         });
209:     }
210: };
211:
212: const getStatusBadge = (token: ShareToken) => {
213:     if (!token.isActive) {
214:         return (
215:             <span className="inline-flex items-center gap-1 px-2 py-1 rounded-full text-xs font-medium bg-gray-100">
216:                 <XCircle className="w-3 h-3" />
217:                 Revoked
218:             </span>
219:         );
220:     }
221:     if (token.isExpired) {
222:         return (
223:             <span className="inline-flex items-center gap-1 px-2 py-1 rounded-full text-xs font-medium bg-red-100">
224:                 <XCircle className="w-3 h-3" />
225:                 Expired
226:             </span>
227:         );
228:     }
229:     if (token.isMaxViewsReached) {
230:         return (
231:             <span className="inline-flex items-center gap-1 px-2 py-1 rounded-full text-xs font-medium bg-orange-100">
232:                 <XCircle className="w-3 h-3" />
233:                 Max Views
234:             </span>
235:         );
236:     }
237:     return (
238:         <span className="inline-flex items-center gap-1 px-2 py-1 rounded-full text-xs font-medium bg-green-100">
239:             <CheckCircle2 className="w-3 h-3" />
240:             Active
241:         </span>
242:     );
243: };
244:
245: return (
246:     <div className="space-y-4">
247:         <div className="flex justify-between items-center">
248:             <div className="mb-8">
249:                 <Card className="shadow-lg border-0 w-full">
250:                     <CardHeader className="bg-gradient-to-r from-blue-600 via-blue-500 to-cyan-500 text-white rounded">
251:                         <CardTitle className="text-lg sm:text-xl md:text-2xl font-bold uppercase flex items-center justify-between">
252:                             Sleep & Rest Management
253:                         </CardTitle>
254:                     </CardHeader>
255:                     </Card>
256:                 </div>

```

```

257:     <Dialog open={isDialogOpen} onOpenChange={setIsDialogOpen}>
258:       <DialogTrigger asChild>
259:         <Button className="gap-2">
260:           <Share2 className="w-4 h-4" />
261:           Create Share Link
262:         </Button>
263:       </DialogTrigger>
264:     <DialogContent className="max-w-md">
265:       <DialogHeader>
266:         <DialogTitle>Create Secure Share Link</DialogTitle>
267:       </DialogHeader>
268:       <div className="space-y-4 pt-4">
269:         { /* Expiration */ }
270:         <div className="space-y-2">
271:           <Label>Link Expiration</Label>
272:           <Select
273:             value={expirationOption}
274:             onValueChange={setExpirationOption}
275:           >
276:             <SelectTrigger>
277:               <SelectValue />
278:             </SelectTrigger>
279:             <SelectContent>
280:               <SelectItem value="1">1 Day</SelectItem>
281:               <SelectItem value="7">7 Days</SelectItem>
282:               <SelectItem value="30">30 Days</SelectItem>
283:               <SelectItem value="90">90 Days</SelectItem>
284:               <SelectItem value="custom">Custom</SelectItem>
285:               <SelectItem value="never">Never Expire</SelectItem>
286:             </SelectContent>
287:           </Select>
288:           {expirationOption === "custom" && (
289:             <Input
290:               type="number"
291:               placeholder="Number of days"
292:               value={customDays}
293:               onChange={(e) => setCustomDays(e.target.value)}
294:               min="1"
295:             />
296:           )}
297:         </div>
298:
299:         { /* Max Views - Optional, typically not used with password */ }
300:         <div className="space-y-2">
301:           <Label>Maximum Total Views (Optional)</Label>
302:           <Input
303:             type="number"
304:             placeholder="Leave empty for unlimited"
305:             value={maxViews}
306:             onChange={(e) => setMaxViews(e.target.value)}
307:             min="1"
308:           />
309:           <p className="text-xs text-gray-500">
310:             Rarely used. Anyone with the password can access unlimited
311:             times unless this is set.
312:           </p>
313:         </div>
314:
315:         { /* Password Protection - Primary Security */ }
316:         <div className="space-y-2">
317:           <div className="flex items-center gap-2">
318:             <input
319:               type="checkbox"
320:               id="usePassword"
321:               checked={usePassword}
322:               onChange={(e) => setUsePassword(e.target.checked)}
323:               className="rounded"
324:             />
325:             <Label
326:               htmlFor="usePassword"
327:               className="cursor-pointer font-semibold"
328:             >
329:               Password Protection (Recommended)

```

```

330:         </Label>
331:     </div>
332:     <p className="text-xs text-gray-500 ml-6">
333:         Anyone with the password can access this report unlimited
334:         times
335:     </p>
336:     {usePassword && (
337:         <Input
338:             type="password"
339:             placeholder="Enter password"
340:             value={password}
341:             onChange={(e) => setPassword(e.target.value)}
342:         />
343:     )}
344: </div>
345:
346: <Button
347:     onClick={handleCreateShare}
348:     disabled={isLoading || (usePassword && !password)}
349:     className="w-full"
350: >
351:     {isLoading ? "Creating..." : "Generate Share Link"}
352: </Button>
353: </div>
354: </DialogContent>
355: </Dialog>
356: </div>
357:
358: {/* Existing Share Links */}
359: <div className="space-y-3">
360:     {shareTokens.length === 0 ? (
361:         <Card>
362:             <CardContent className="pt-6">
363:                 <p className="text-center text-gray-500 text-sm">
364:                     No share links created yet. Create one to share this report
365:                     securely.
366:                 </p>
367:             </CardContent>
368:         </Card>
369:     ) : (
370:         shareTokens.map((token) => (
371:             <Card key={token.id} className="relative">
372:                 <CardContent className="pt-4">
373:                     <div className="space-y-3">
374:                         <div className="flex items-start justify-between">
375:                             <div className="flex-1 min-w-0">
376:                                 <div className="flex items-center gap-2 mb-2">
377:                                     {getStatusBadge(token)}
378:                                     {token.hasPassword && (
379:                                         <span className="inline-flex items-center gap-1 px-2 py-1 rounded-full text-xs font-medium">
380:                                             <Lock className="w-3 h-3" />
381:                                             Protected
382:                                         </span>
383:                                     )}
384:                                 </div>
385:                                 <div className="text-sm font-mono text-gray-600 truncate">
386:                                     {token.url}
387:                                 </div>
388:                             </div>
389:                             <div className="flex gap-2 ml-4">
390:                                 <Button
391:                                     size="sm"
392:                                     variant="outline"
393:                                     onClick={() => handleCopyLink(token.url, token.token)}
394:                                     className="gap-1"
395:                                 >
396:                                     {copiedToken === token.token ? (
397:                                         <>
398:                                             <CheckCircle2 className="w-4 h-4" />
399:                                             Copied
400:                                         </>
401:                                     ) : (
402:                                         <>

```

```

403:             <Copy className="w-4 h-4" />
404:             Copy
405:         </>
406:     )}
407: </Button>
408: {token.isActive && (
409:     <Button
410:         size="sm"
411:         variant="destructive"
412:         onClick={() => handleRevokeToken(token.id)}
413:     >
414:         <Trash2 className="w-4 h-4" />
415:     </Button>
416: )}
417: </div>
418: </div>
419:
420: <div className="grid grid-cols-2 gap-4 text-xs text-gray-600">
421:     <div className="flex items-center gap-1">
422:         <Eye className="w-3 h-3" />
423:         <span>
424:             Total Views: {token.viewCount}
425:             {token.maxViews ? `/${token.maxViews}` : " (Unlimited)"}
426:         </span>
427:     </div>
428:     <div className="flex items-center gap-1">
429:         <Calendar className="w-3 h-3" />
430:         <span>
431:             {token.expiresAt
432:                 ? `Expires: ${new Date(
433:                     token.expiresAt
434:                 ).toLocaleDateString()}`
435:                 : "Never expires"}
436:         </span>
437:     </div>
438: </div>
439:
440: {token.lastAccessedAt && (
441:     <div className="text-xs text-gray-500">
442:         Last accessed:{" "}
443:         {new Date(token.lastAccessedAt).toLocaleString()}
444:     </div>
445: )}
446: </div>
447: </CardContent>
448: </Card>
449: ))
450: )}
451: </div>
452: </div>
453: );
454: }

```

## ■ File: components\admin\sleep-rest-admin.tsx

```

=====
1: "use client";
2:
3: import { useEffect, useState } from "react";
4: import { Card, CardContent, CardHeader, CardTitle } from "@/components/ui/card";
5: import { Input } from "@/components/ui/input";
6: import { Textarea } from "@/components/ui/textarea";
7: import { Label } from "@/components/ui/label";
8: import { Button } from "@/components/ui/button";
9: import { Save, Plus, Upload, Trash2, X } from "lucide-react";
10: import Image from "next/image";
11: import { SleepAndRest } from "@/types/report-types";
12: import {
13:     Select,
14:     SelectContent,
15:     SelectItem,
16:     SelectTrigger,

```

```

17:   SelectValue,
18: } from "@components/ui/select";
19:
20: interface SleepImage {
21:   label: string;
22:   url: string;
23: }
24:
25: interface Props {
26:   data: SleepAndRest;
27:   updateData: (data: Partial<SleepAndRest>) => void;
28:   onSave: () => void;
29:   onReset: () => void;
30: }
31:
32: interface SleepEntry {
33:   title: string;
34:   intervention: string;
35:   image: string;
36: }
37:
38: export default function SleepRestAdmin({
39:   data,
40:   updateData,
41:   onSave,
42:   onReset,
43: }: Props) {
44:   const [quote, setQuote] = useState(data.quote || "");
45:   const [description, setDescription] = useState(data.description || "");
46:   const [entries, setEntries] = useState<Record<string, SleepEntry>>({
47:     data.data || {}
48:   });
49:   const [newEntryTitle, setNewEntryTitle] = useState("");
50:   const [showAddModal, setShowAddModal] = useState(false);
51:   const [images, setImages] = useState<Record<string, string>>({});
52:   const [showImageModal, setShowImageModal] = useState(false);
53:   const [selectedImageField, setSelectedImageField] = useState<string | null>(
54:     null
55:   );
56:   const [imageLabel, setImageLabel] = useState("");
57:   const [imageFile, setImageFile] = useState<File | null>(null);
58:
59:   useEffect(() => {
60:     updateData({ quote, description });
61:   }, [quote, description]);
62:
63:   useEffect(() => {
64:     const fetchImages = async () => {
65:       const res = await fetch("/api/sleep-image");
66:       const json = await res.json();
67:       if (json.success) {
68:         const map: Record<string, string> = {};
69:         for (const img of json.images) {
70:           map[img.label] = img.url;
71:         }
72:         setImages(map);
73:       }
74:     };
75:     fetchImages();
76:   }, []);
77:
78:   const handleUploadImage = async () => {
79:     if (!imageLabel.trim() || !imageFile) return;
80:
81:     const customLabel = imageLabel.trim().toLowerCase();
82:     const formData = new FormData();
83:     formData.append("file", imageFile);
84:     formData.append("label", customLabel);
85:     formData.append("folder", "sleep");
86:
87:     const res = await fetch("/api/sleep-image", {
88:       method: "POST",
89:       body: formData,

```



```

90:     });
91:
92:     const json = await res.json();
93:     if (json.success) {
94:         // Use the custom label you provided, not what the API returns
95:         setImages((prev) => ({ ...prev, [customLabel]: json.url }));
96:         setImageLabel("");
97:         setImageFile(null);
98:     }
99: };
100:
101: const handleFieldChange = (
102:     key: string,
103:     field: keyof SleepEntry,
104:     value: string
105: ) => {
106:     const updated = {
107:         ...entries,
108:         [key]: {
109:             ...entries[key],
110:             [field]: value,
111:         },
112:     };
113:     setEntries(updated);
114:     updateData({ data: updated });
115: };
116:
117: const handleDeleteField = (key: string) => {
118:     const updated = { ...entries };
119:     delete updated[key];
120:     setEntries(updated);
121:     updateData({ data: updated });
122: };
123:
124: const handleAddField = () => {
125:     const newKey = `entry_${Date.now()}`;
126:     const updated = {
127:         ...entries,
128:         [newKey]: {
129:             title: newEntryTitle.trim(),
130:             intervention: "",
131:             image: "",
132:         },
133:     };
134:     setEntries(updated);
135:     updateData({ data: updated });
136:     setNewEntryTitle("");
137:     setShowAddModal(false);
138: };
139:
140: const handleImageUpload = async (file: File, key: string) => {
141:     const label = `entry_image_${Date.now()}`;
142:     const formData = new FormData();
143:     formData.append("file", file);
144:     formData.append("label", label);
145:     formData.append("folder", "sleep");
146:
147:     const res = await fetch("/api/sleep-image", {
148:         method: "POST",
149:         body: formData,
150:     });
151:
152:     const json = await res.json();
153:     if (json.success) {
154:         handleFieldChange(key, "image", json.label || label);
155:         setImages((prev) => ({ ...prev, [json.label || label]: json.url }));
156:     }
157: };
158:
159: return (
160:     <div className="min-h-screen bg-gradient-to-br from-gray-50 to-blue-50/30 ">
161:         <div className="max-w-8xl mx-auto">
162:             {/* Header Section */}

```

```

163:     <div className="mb-8">
164:         <Card className="shadow-lg border-0">
165:             <CardHeader className="bg-gradient-to-r from-blue-600 via-blue-500 to-cyan-500 text-white rounded">
166:                 <CardTitle className="text-lg sm:text-xl md:text-2xl font-bold uppercase flex items-center justify-center">
167:                     Sleep & Rest Management
168:                 </CardTitle>
169:             </CardHeader>
170:         </Card>
171:     </div>
172:
173:     <div className="my-4 text-end">
174:         <Button onClick={onSave}>
175:             <Save className="w-4 h-4 mr-2" />
176:             Save Changes
177:         </Button>
178:     </div>
179:
180:     { /* Content Management Section */ }
181:     <div className="mb-8">
182:         <h2 className="text-xl font-semibold text-gray-800 mb-4">
183:             Content Management
184:         </h2>
185:         <div className="grid grid-cols-1 lg:grid-cols-2 gap-6">
186:             { /* Quote Section */ }
187:             <Card className="shadow-md hover:shadow-lg transition-shadow">
188:                 <CardHeader className="bg-gradient-to-r from-purple-50 to-blue-50 border-b">
189:                     <CardTitle className="text-lg text-gray-700 flex items-center">
190:                         <div className="w-2 h-2 bg-purple-500 rounded-full mr-3"></div>
191:                         Inspirational Quote
192:                     </CardTitle>
193:                 </CardHeader>
194:                 <CardContent className="p-6">
195:                     <Textarea
196:                         value={quote}
197:                         onChange={(e) => setQuote(e.target.value)}
198:                         placeholder="Write a sleep-related inspirational quote..."
199:                         className="min-h-[120px] resize-none"
200:                     />
201:                 </CardContent>
202:             </Card>
203:
204:             { /* Description Section */ }
205:             <Card className="shadow-md hover:shadow-lg transition-shadow">
206:                 <CardHeader className="bg-gradient-to-r from-green-50 to-blue-50 border-b">
207:                     <CardTitle className="text-lg text-gray-700 flex items-center">
208:                         <div className="w-2 h-2 bg-green-500 rounded-full mr-3"></div>
209:                         Section Description
210:                     </CardTitle>
211:                 </CardHeader>
212:                 <CardContent className="p-6">
213:                     <Textarea
214:                         value={description}
215:                         onChange={(e) => setDescription(e.target.value)}
216:                         placeholder="Provide a comprehensive description for the Sleep & Rest section..."
217:                         className="min-h-[120px] resize-none"
218:                     />
219:                 </CardContent>
220:             </Card>
221:         </div>
222:     </div>
223:
224:     { /* Image Management Section */ }
225:     { /* Image Management Section */ }
226:     <div className="mb-8">
227:         <h2 className="text-xl font-semibold text-gray-800 mb-4">
228:             Image Management
229:         </h2>
230:         <Card className="shadow-md">
231:             <CardHeader className="bg-gradient-to-r from-orange-50 to-yellow-50 border-b">
232:                 <CardTitle className="text-lg text-gray-700 flex items-center">
233:                     <Upload className="w-5 h-5 mr-3 text-orange-500" />
234:                     Upload New Image
235:                 </CardTitle>

```

```

236:         </CardHeader>
237:
238:         <CardContent className="p-6">
239:             <div className="grid grid-cols-1 md:grid-cols-4 gap-4 items-end">
240:                 {/* Image Label Input */}
241:                 <div className="space-y-2">
242:                     <Label htmlFor="imageLabel" className="text-sm font-medium">
243:                         Image Label
244:                     </Label>
245:                     <Input
246:                         id="imageLabel"
247:                         placeholder="e.g., sleep_cycle"
248:                         value={imageLabel}
249:                         onChange={(e) => setImageLabel(e.target.value)}
250:                     />
251:                 </div>
252:
253:                 {/* Image File Picker */}
254:                 <div className="space-y-2">
255:                     <Label htmlFor="imageFile" className="text-sm font-medium">
256:                         Select File
257:                     </Label>
258:                     <Input
259:                         id="imageFile"
260:                         type="file"
261:                         accept="image/*"
262:                         onChange={(e) => setImageFile(e.target.files?.[0] || null)}
263:                     />
264:                 </div>
265:
266:                 <Button
267:                     onClick={handleUploadImage}
268:                     disabled={!imageLabel || !imageFile}
269:                     className="bg-orange-500 hover:bg-orange-600 text-white h-10"
270:                 >
271:                     <Upload className="w-4 h-4 mr-2" />
272:                     Upload
273:                 </Button>
274:
275:                 <Button
276:                     variant="outline"
277:                     onClick={() => setShowImageModal(true)}
278:                     className="text-blue-600 border-blue-300 hover:bg-blue-50"
279:                 >
280:                     View Uploaded Images
281:                 </Button>
282:             </div>
283:         </CardContent>
284:     </Card>
285: </div>
286:
287: {/* Sleep Entries Section */}
288: <div className="mb-8">
289:     <div className="flex justify-between items-center mb-6">
290:         <h2 className="text-xl font-semibold text-gray-800">
291:             Sleep Entries
292:         </h2>
293:         <span className="text-sm text-gray-500 bg-gray-100 px-3 py-1 rounded-full">
294:             {Object.keys(entries).length} entries
295:         </span>
296:     </div>
297:
298:     {/* Add Entry Button */}
299:     <div className="mb-6 justify-end">
300:         <Button
301:             variant="outline"
302:             className="py-6 border-red-600"
303:             onClick={() => setShowAddModal(true)}
304:         >
305:             <Plus className="mr-2" />
306:             Add Sleep Entry
307:         </Button>
308:     </div>

```

[illegible]

```

382:             {label.replace(/_/g, " ")}
383:         </span>
384:     </div>
385: </SelectedItem>
386:     )}}
387: </SelectContent>
388: </Select>
389: </div>
390:
391: <div className="space-y-2">
392:   <Label
393:     htmlFor={`intervention-${key}`}
394:     className="text-sm font-medium"
395:   >
396:     Intervention Advice
397:   </Label>
398:   <Input
399:     id={`intervention-${key}`}
400:     value={entry.intervention}
401:     onChange={(e) =>
402:       handleFieldChange(key, "intervention", e.target.value)
403:     }
404:     placeholder="Provide intervention advice..."
405:     className="resize-none"
406:   />
407: </div>
408: </CardContent>
409: </Card>
410:   )}}
411: </div>
412: )}
413: </div>
414:
415: { /* Add Modal */ }
416: {showAddModal && (
417:   <div className="fixed inset-0 bg-black/60 z-50 flex items-center justify-center p-4 backdrop-blur"
418:     <div className="bg-white rounded-xl shadow-2xl w-full max-w-md transform transition-all">
419:       <div className="flex justify-between items-center p-6 border-b bg-gradient-to-r from-blue-50"
420:         <h3 className="text-xl font-semibold text-gray-800">
421:           Add New Sleep Entry
422:         </h3>
423:         <button
424:           onClick={() => setShowAddModal(false)}
425:           className="text-gray-500 hover:text-gray-700 hover:bg-gray-100 p-1 rounded-full transition"
426:         >
427:           <X className="w-5 h-5" />
428:         </button>
429:       </div>
430:       <div className="p-6">
431:         <div className="space-y-2">
432:           <Label
433:             htmlFor="newEntryTitle"
434:             className="text-sm font-medium"
435:           >
436:             Entry Title
437:           </Label>
438:           <Input
439:             id="newEntryTitle"
440:             placeholder="Enter title for the new entry"
441:             value={newEntryTitle}
442:             onChange={(e) => setNewEntryTitle(e.target.value)}
443:             onKeyDown={(e) => e.key === "Enter" && handleAddField()}
444:             className="w-full"
445:           />
446:         </div>
447:       </div>
448:       <div className="flex justify-end gap-3 p-6 border-t bg-gray-50">
449:         <Button
450:           variant="outline"
451:           onClick={() => setShowAddModal(false)}
452:           className="px-4 py-2"
453:         >
454:           Cancel

```

```

455:         </Button>
456:     <Button
457:       onClick={handleAddField}
458:       disabled={!newEntryTitle.trim()}
459:       className="px-4 py-2 bg-blue-500 hover:bg-blue-600 text-white"
460:     >
461:       <Plus className="w-4 h-4 mr-2" />
462:       Create Entry
463:     </Button>
464:   </div>
465: </div>
466: </div>
467: )}
468:
469: { /* Image Modal */ }
470: { /* Image Modal */ }
471: {showImageModal && (
472:   <div className="fixed inset-0 bg-black/60 z-50 flex items-center justify-center p-4 backdrop-blur"
473:     <div className="bg-white rounded-xl shadow-2xl w-full max-w-3xl max-h-[90vh] overflow-hidden">
474:       <div className="flex justify-between items-center p-6 border-b bg-gradient-to-r from-blue-50
475:         <h3 className="text-xl font-semibold text-gray-800">
476:           Select Sleep Image
477:         </h3>
478:         <button
479:           onClick={() => setShowImageModal(false)}
480:           className="text-gray-500 hover:text-gray-700 hover:bg-gray-100 p-1 rounded-full transition"
481:         >
482:           <X className="w-5 h-5" />
483:         </button>
484:       </div>
485:       <div className="p-6 overflow-y-auto max-h-[calc(90vh-140px)] space-y-3">
486:         {Object.entries(images).map(([label, url]) => {
487:           const fileName = url.split("/").pop();
488:
489:           return (
490:             <div
491:               key={label}
492:               className="flex flex-col sm:flex-row sm:items-center sm:justify-between gap-3 border
493:             >
494:               <div
495:                 className="flex items-center gap-4 cursor-pointer flex-1"
496:                 onClick={() => {
497:                   if (selectedImageField) {
498:                     handleFieldChange(
499:                       selectedImageField,
500:                       "image",
501:                       label
502:                     );
503:                     setShowImageModal(false);
504:                   }
505:                 }}
506:               >
507:                 <Image
508:                   src={url}
509:                   alt={label}
510:                   width={50}
511:                   height={50}
512:                   className="rounded-md object-cover shrink-0"
513:                 />
514:                 <span className="capitalize text-sm text-gray-700 truncate">
515:                   {label.replace(/_/g, " ")}
516:                 </span>
517:               </div>
518:
519:               <Button
520:                 size="icon"
521:                 variant="destructive"
522:                 className="shrink-0"
523:                 onClick={async (e) => {
524:                   e.stopPropagation();
525:                   if (!fileName) return;
526:
527:                   const confirmed = confirm(

```

```

528:         `Delete image "${fileName}"?`
529:     );
530:     if (!confirmed) return;
531:
532:     try {
533:         const res = await fetch(
534:             `/api/sleep-image?file=${fileName}`,
535:             {
536:                 method: "DELETE",
537:             }
538:         );
539:
540:         const json = await res.json();
541:         if (json.success) {
542:             const updated = { ...images };
543:             delete updated[label];
544:             setImages(updated);
545:         } else {
546:             alert("Delete failed");
547:         }
548:     } catch (err) {
549:         console.error("Error deleting image:", err);
550:         alert("Delete error");
551:     }
552: }}
553: >
554:     <Trash2 className="w-4 h-4" />
555:   </Button>
556: </div>
557:   );
558:   }}}
559: </div>
560: </div>
561: </div>
562:   )}
563: </div>
564: </div>
565: );
566: }

```

## **File: components\admin\sports-fitness-admin.tsx**

```

1: "use client";
2:
3: import {
4:   Card,
5:   CardContent,
6:   CardDescription,
7:   CardHeader,
8:   CardTitle,
9: } from "@/components/ui/card";
10: import {
11:   Dialog,
12:   DialogTrigger,
13:   DialogContent,
14:   DialogHeader,
15:   DialogTitle,
16:   DialogDescription,
17: } from "@/components/ui/dialog";
18:
19: import { Input } from "@/components/ui/input";
20: import { Label } from "@/components/ui/label";
21: import { Textarea } from "@/components/ui/textarea";
22: import { Button } from "@/components/ui/button";
23: import { Save, RotateCcw, Plus, Trash2, Settings } from "lucide-react";
24: import { useState } from "react";
25: import type { SportsAndFitness } from "@/types/report-types";
26: import {
27:   Select,
28:   SelectTrigger,
29:   SelectValue,

```

```

30:   SelectContent,
31:   SelectItem,
32: } from "@components/ui/select";
33: import {
34:   AlertDialog,
35:   AlertDialogAction,
36:   AlertDialogCancel,
37:   AlertDialogContent,
38:   AlertDialogDescription,
39:   AlertDialogFooter,
40:   AlertDialogHeader,
41:   AlertDialogTitle,
42:   AlertDialogTrigger,
43: } from "@components/ui/alert-dialog";
44:
45: import { useEffect } from "react";
46: import { useToast } from "@components/ui/use-toast";
47:
48: // Define the field data structure to match the interface
49: interface FieldData {
50:   level: string;
51:   description: string;
52: }
53:
54: interface SportsFitnessAdminProps {
55:   sportsAndFitness: SportsAndFitness;
56:   updateSportsAndFitness: (
57:     section: keyof SportsAndFitness | "quote" | "description" | "customImages",
58:     field: string,
59:     data: FieldData | string | { label: string; url: string } | null
60:   ) => void;
61:   onSave: () => void;
62:   onReset: () => void;
63: }
64:
65: export default function SportsFitnessAdmin({
66:   sportsAndFitness,
67:   updateSportsAndFitness,
68:   onSave,
69:   onReset,
70: }: SportsFitnessAdminProps) {
71:   const { toast } = useToast();
72:   const [activeSection, setActiveSection] = useState<string>("");
73:   const [newFieldName, setNewFieldName] = useState("");
74:   const [imageLabel, setImageLabel] = useState("");
75:   const [newCategoryName, setNewCategoryName] = useState("");
76:   const [customImages, setCustomImages] = useState<Record<string, string>>({});
77:   const [showImageViewer, setShowImageViewer] = useState(false);
78:   const [showImageForm, setShowImageForm] = useState(false);
79:   const [deleteConfirm, setDeleteConfirm] = useState<{
80:     type: "category" | "field" | null;
81:     target: string;
82:     section?: string;
83:   }>({ type: null, target: "" });
84:
85:   // Get categories (exclude quote, description, customImages)
86:   const getCategories = (): string[] => {
87:     return Object.keys(sportsAndFitness).filter(
88:       (key) =>
89:         ![ "quote", "description", "customImages" ].includes(key) &&
90:         typeof sportsAndFitness[key] === "object" &&
91:         !Array.isArray(sportsAndFitness[key])
92:     );
93:   };
94:
95:   // Initialize active section
96:   useEffect(() => {
97:     const categories = getCategories();
98:     if (!activeSection && categories.length > 0) {
99:       setActiveSection(categories[0]);
100:     }
101:   }, [sportsAndFitness, activeSection]);
102:

```



```

103: const addNewCategory = () => {
104:     const key = newCategoryName.trim().replace(/\s+/g, "_").toLowerCase();
105:
106:     if (!key || key in sportsAndFitness) {
107:         toast({
108:             title: "Error",
109:             description: "Invalid or existing category name",
110:             variant: "destructive",
111:             duration: 1500,
112:         });
113:         return;
114:     }
115:
116:     // Initialize new category with empty object
117:     updateSportsAndFitness(key as keyof SportsAndFitness, "__init__", {
118:         level: "",
119:         description: "",
120:     });
121:
122:     setNewCategoryName("");
123:     setActiveSection(key);
124:
125:     toast({
126:         title: "Category Added",
127:         description: `Successfully created category "${key}"`,
128:         duration: 1500,
129:         variant: "success",
130:     });
131: };
132:
133: const addNewField = () => {
134:     if (!activeSection) return;
135:
136:     const fieldKey = newFieldName.toLowerCase().replace(/\s+/g, "_");
137:     const currentSection = sportsAndFitness[activeSection];
138:
139:     if (
140:         typeof currentSection === "object" &&
141:         !Array.isArray(currentSection) &&
142:         currentSection[fieldKey]
143:     ) {
144:         toast({
145:             title: "Error",
146:             description: "Field already exists",
147:             variant: "destructive",
148:             duration: 1500,
149:         });
150:         return;
151:     }
152:
153:     updateSportsAndFitness(activeSection as keyof SportsAndFitness, fieldKey, {
154:         level: "",
155:         description: "",
156:     });
157:
158:     setNewFieldName("");
159:
160:     toast({
161:         title: "Field Added",
162:         description: `Successfully added field "${newFieldName}"`,
163:         duration: 1500,
164:         variant: "success",
165:     });
166: };
167:
168: const handleImageUpload = async (e: React.FormEvent<HTMLFormElement>) => {
169:     e.preventDefault();
170:
171:     const form = e.currentTarget;
172:     const formData = new FormData(form);
173:
174:     const label = (formData.get("imageLabel") as string)?.trim().toLowerCase();
175:     const file = formData.get("imageFile") as File;

```

```

176:
177:     if (!file || !label) {
178:         toast({
179:             title: "Error",
180:             description: "Both label and image file are required",
181:             variant: "destructive",
182:             duration: 1500,
183:         });
184:         return;
185:     }
186:
187:     const uploadData = new FormData();
188:     uploadData.append("file", file);
189:     uploadData.append("label", label);
190:     uploadData.append("folder", "sports");
191:
192:     try {
193:         const response = await fetch("/api/sports-images", {
194:             method: "POST",
195:             body: uploadData,
196:         });
197:
198:         const text = await response.text();
199:         let result;
200:
201:         try {
202:             result = JSON.parse(text);
203:         } catch {
204:             console.error("Invalid JSON from server:", text);
205:             toast({
206:                 title: "Upload Failed",
207:                 description: "Server returned invalid response",
208:                 variant: "destructive",
209:                 duration: 1500,
210:             });
211:             return;
212:         }
213:
214:         if (result.success) {
215:             setCustomImages((prev) => ({
216:                 ...prev,
217:                 [label]: result.url,
218:             }));
219:
220:             updateSportsAndFitness("customImages" as any, label, {
221:                 label,
222:                 url: result.url,
223:             });
224:
225:             form.reset();
226:             setImageLabel("");
227:
228:             toast({
229:                 title: "Image Uploaded",
230:                 description: `Successfully uploaded image "${label}"`,
231:                 duration: 1500,
232:                 variant: "success",
233:             });
234:         } else {
235:             toast({
236:                 title: "Upload Failed",
237:                 description: result.error || "Unknown error occurred",
238:                 variant: "destructive",
239:                 duration: 1500,
240:             });
241:         }
242:     } catch (err) {
243:         console.error("Upload error", err);
244:         toast({
245:             title: "Upload Error",
246:             description: "An error occurred while uploading",
247:             variant: "destructive",
248:             duration: 1500,

```

```

249:     });
250:   }
251: };
252:
253: const deleteField = (field: string) => {
254:   setDeleteConfirm({
255:     type: "field",
256:     target: field,
257:     section: activeSection,
258:   });
259: };
260:
261: const deleteCategory = (section: string) => {
262:   setDeleteConfirm({
263:     type: "category",
264:     target: section,
265:   });
266: };
267:
268: const confirmDelete = () => {
269:   if (deleteConfirm.type === "field" && deleteConfirm.section) {
270:     // Delete field by passing null as the data
271:     updateSportsAndFitness(
272:       deleteConfirm.section as keyof SportsAndFitness,
273:       deleteConfirm.target,
274:       null
275:     );
276:
277:     toast({
278:       title: "Field Deleted",
279:       description: `Successfully deleted field "${deleteConfirm.target}"`,
280:       duration: 1500,
281:       variant: "success",
282:     });
283:   } else if (deleteConfirm.type === "category") {
284:     updateSportsAndFitness(
285:       deleteConfirm.target as keyof SportsAndFitness,
286:       "__delete_category__",
287:       null
288:     );
289:
290:     const categories = getCategories().filter(
291:       (cat) => cat !== deleteConfirm.target
292:     );
293:     setActiveSection(categories[0] || "");
294:
295:     toast({
296:       title: "Category Deleted",
297:       description: `Successfully deleted category "${deleteConfirm.target}"`,
298:       duration: 1500,
299:       variant: "success",
300:     });
301:   }
302:
303:   setDeleteConfirm({ type: null, target: "" });
304: };
305:
306: useEffect(() => {
307:   const fetchImages = async () => {
308:     try {
309:       const res = await fetch("/api/sports-images");
310:       const data = await res.json();
311:
312:       if (data.success) {
313:         const imageMap: Record<string, string> = {};
314:         data.images.forEach((img: { label: string; url: string }) => {
315:           imageMap[img.label.toLowerCase()] = img.url;
316:         });
317:         setCustomImages(imageMap);
318:       }
319:     } catch (err) {
320:       console.error("Failed to load images", err);
321:       toast({

```

```

322:         title: "Load Error",
323:         description: "Failed to load custom images",
324:         variant: "destructive",
325:         duration: 1500,
326:     });
327: }
328: };
329:
330:     fetchImages();
331: }, [toast]);
332:
333: const handleFieldUpdate = (
334:     section: string,
335:     field: string,
336:     data: Partial<FieldData>
337: ) => {
338:     const currentSection = sportsAndFitness[section];
339:     if (typeof currentSection === "object" && !Array.isArray(currentSection)) {
340:         const currentField = currentSection[field];
341:         if (typeof currentField === "object") {
342:             updateSportsAndFitness(section as keyof SportsAndFitness, field, {
343:                 ...currentField,
344:                 ...data,
345:             });
346:             toast({
347:                 title: "Field Updated",
348:                 description: `Successfully updated ${field}`,
349:                 duration: 1500,
350:                 variant: "success",
351:             });
352:         }
353:     }
354: };
355:
356: const renderExerciseField = (
357:     section: string,
358:     field: string,
359:     data: FieldData
360: ) => {
361:     const selectedImageKey = data.level?.toLowerCase();
362:     const imageUrl =
363:         customImages?.[selectedImageKey] || `/sports/${selectedImageKey}.png`;
364:
365:     return (
366:         <Card
367:             key={field}
368:             className="border border-green-200 hover:border-green-300 transition-colors"
369:         >
370:             <CardHeader className="pb-3">
371:                 <div className="flex items-center justify-between">
372:                     <CardTitle className="text-base font-medium text-green-800 truncate">
373:                         {field.replace(/_/g, " ").toUpperCase()}
374:                     </CardTitle>
375:                     <Button
376:                         size="sm"
377:                         variant="ghost"
378:                         onClick={() => deleteField(field)}
379:                         className="text-red-600 hover:bg-red-100 hover:text-red-700 flex-shrink-0 h-8 w-8 p-0"
380:                     >
381:                         <Trash2 className="h-4 w-4" />
382:                     </Button>
383:                 </div>
384:             </CardHeader>
385:             <CardContent className="space-y-3 pt-0">
386:                 <div className="space-y-2">
387:                     <Label className="text-xs font-medium text-gray-600">
388:                         Image Type
389:                     </Label>
390:                     <Select
391:                         value={data.level}
392:                         onChange={(value) =>
393:                             handleFieldUpdate(section, field, { level: value })
394:                     >

```

```

395:         >
396:         <SelectTrigger className="h-9 text-sm">
397:           <SelectValue placeholder="Select level" />
398:         </SelectTrigger>
399:
400:         <SelectContent>
401:           {Object.keys(customImages).map((option) => {
402:             const cleanLabel = option
403:               .replace(/^\d+$/, "")
404:               .replace(/\.png$/i, "");
405:             const imagePath =
406:               customImages[option.toLowerCase()] ||
407:               `/sports/${cleanLabel.toLowerCase()}.png`;
408:
409:             return (
410:               <SelectItem key={`image-${option}`} value={cleanLabel}>
411:                 <div className="flex items-center space-x-2 gap-9">
412:                   <img
413:                     src={imagePath}
414:                     alt={cleanLabel}
415:                     className="w-6 h-6 rounded object-cover"
416:                   />
417:                   <span>{cleanLabel.toUpperCase()}</span>
418:                 </div>
419:               </SelectItem>
420:             );
421:           })}
422:         </SelectContent>
423:       </Select>
424:     </div>
425:     <div className="space-y-2">
426:       <Label className="text-xs font-medium text-gray-600">
427:         Description
428:       </Label>
429:       <Input
430:         value={data.description}
431:         onChange={(e) =>
432:           handleFieldUpdate(section, field, {
433:             description: e.target.value,
434:           })
435:         }
436:         className="text-xs resize-none"
437:         placeholder="Enter description..."
438:       />
439:     </div>
440:   </CardContent>
441: </Card>
442: );
443: };
444:
445: const getSectionTitle = (section: string) =>
446:   section.replace(/([A-Z])/g, " $1").replace(/^./, (s) => s.toUpperCase());
447:
448: const getSectionIcon = (section: string) => {
449:   if (section.toLowerCase().includes("performance")) return "";
450:   if (section.toLowerCase().includes("exercise")) return "";
451:   if (section.toLowerCase().includes("cardio")) return "";
452:   if (section.toLowerCase().includes("strength")) return "";
453:   return "";
454: };
455:
456: const categories = getCategories();
457:
458: return (
459:   <div className="w-full rounded-lg max-w-8xl mx-auto">
460:     <Card className="shadow-lg border-0 bg-white">
461:       { /* Header */ }
462:       <CardHeader className="bg-gradient-to-r from-green-500 to-emerald-600 text-white rounded-lg p-4 sm:p-6">
463:         <CardTitle className="text-base sm:text-xl md:text-2xl font-bold uppercase flex items-center justify-between">
464:           Sports & Fitness
465:         </CardTitle>
466:       </CardHeader>
467:

```

```

468:     <div className="text-end mt-4">
469:         <Button onClick={onSave} size="sm">
470:             <Save className="h-4 w-4 mr-2" />
471:             Save Changes
472:         </Button>
473:     </div>
474:
475: <CardContent className="p-4 sm:p-6 space-y-6">
476:     <div className="grid gap-4 mb-8">
477:         <div>
478:             <label className="block font-semibold text-sm mb-1">Quote</label>
479:             <Input
480:                 value={sportsAndFitness.quote || ""}
481:                 onChange={(e) =>
482:                     updateSportsAndFitness("quote", "", e.target.value)
483:                 }
484:                 placeholder="Enter a motivational quote related to fitness"
485:             />
486:         </div>
487:
488:         <div>
489:             <label className="block font-semibold text-sm mb-1">
490:                 Description
491:             </label>
492:             <Input
493:                 value={sportsAndFitness.description || ""}
494:                 onChange={(e) =>
495:                     updateSportsAndFitness("description", "", e.target.value)
496:                 }
497:                 placeholder="Provide a short description of the Sports & Fitness section"
498:             />
499:         </div>
500:     </div>
501:
502:     { /* Category Navigation */ }
503:     <div className="space-y-4">
504:         <div className="flex flex-col lg:flex-row lg:items-center lg:justify-between gap-4">
505:             { /* Category Buttons */ }
506:             <div className="grid grid-cols-2 sm:grid-cols-3 md:grid-cols-4 gap-3">
507:                 {categories.map((section) => (
508:                     <div key={section} className="relative">
509:                         <Button
510:                             variant={
511:                                 activeSection === section ? "default" : "outline"
512:                             }
513:                             size="sm"
514:                             onClick={() => setActiveSection(section)}
515:                             className="flex items-center gap-2 pr-8 text-xs sm:text-sm w-full"
516:                         >
517:                             <span>{getSectionIcon(section)}</span>
518:                             <span className="capitalize hidden sm:inline">
519:                                 {section.replace(/[A-Z_]/g, " $1").trim()}
520:                             </span>
521:                             <span className="capitalize sm:hidden">
522:                                 {section.slice(0, 8)}...
523:                             </span>
524:                         </Button>
525:
526:                         <button
527:                             onClick={() => deleteCategory(section)}
528:                             className="absolute -right-1 -top-1 bg-red-500 text-white rounded-full w-5 h-5 flex i
529:                             title="Delete Category"
530:                         >
531:                             ?
532:                         </button>
533:                     </div>
534:                 )))
535:             </div>
536:
537:             { /* Add Category */ }
538:             <div className="flex flex-col sm:flex-row gap-2">
539:                 <Input
540:                     value={newCategoryName}

```

```

541:         onChange={(e) => setNewCategoryName(e.target.value)}
542:         placeholder="New category name"
543:         className="w-full sm:w-52 h-9 text-sm"
544:     />
545:     <Button size="sm" onClick={addNewCategory} className="h-9">
546:         <Plus className="h-4 w-4 mr-1" /> Add Category
547:     </Button>
548: </div>
549: </div>
550:
551: { /* Field Management */ }
552: <div className="flex flex-col sm:flex-row gap-2 items-start sm:items-center">
553:     <Input
554:         value={newFieldName}
555:         onChange={(e) => setNewFieldName(e.target.value.toUpperCase())}
556:         placeholder="New field name"
557:         className="w-full sm:w-64 h-9 text-sm"
558:         disabled={!activeSection}
559:     />
560:     <Button
561:         size="sm"
562:         onClick={addNewField}
563:         className="h-9 w-full sm:w-auto"
564:         disabled={!activeSection}
565:     >
566:         <Plus className="h-4 w-4 mr-1" /> Add Field
567:     </Button>
568: </div>
569:
570: { /* Custom Image Management */ }
571: <div className="rounded-lg justify-start">
572:     <div className="flex items-center mb-3">
573:         <h4 className="text-sm font-medium mr-3 text-gray-700">
574:             Custom Images
575:         </h4>
576:         <Button
577:             size="sm"
578:             variant="outline"
579:             onClick={() => setShowImageForm(!showImageForm)}
580:             className="h-8"
581:         >
582:             <Plus className="h-4 w-4" />
583:             {showImageForm ? "Hide" : "Add Image"}
584:         </Button>
585:         <Button
586:             size="sm"
587:             variant="outline"
588:             onClick={() => setShowImageViewer(true)}
589:             className="h-8 m-3"
590:         >
591:             View Images
592:         </Button>
593:     </div>
594:
595:     {showImageForm && (
596:         <form
597:             onSubmit={handleImageUpload}
598:             encType="multipart/form-data"
599:             className="flex flex-col sm:flex-row gap-2 items-center"
600:         >
601:             <Input
602:                 type="text"
603:                 name="imageLabel"
604:                 value={imageLabel}
605:                 onChange={(e) =>
606:                     setImageLabel(e.target.value.toUpperCase())
607:                 }
608:                 placeholder="Image Name (e.g. dumbbell)"
609:                 className="w-full sm:w-40 h-9 text-sm"
610:                 required
611:             />
612:
613:             <Input

```

```

614:         type="file"
615:         name="imageFile"
616:         accept="image/*"
617:         className="w-full sm:flex-1 h-9 text-sm"
618:         required
619:     />
620:     <Button
621:         type="submit"
622:         size="sm"
623:         className="h-9 w-full sm:w-auto"
624:     >
625:         <Save className="h-4 w-4 mr-2" />
626:         Upload
627:     </Button>
628: </form>
629: })
630: </div>
631: </div>
632:
633: { /* Content Area */ }
634: <div className="space-y-6">
635:     {activeSection && (
636:         <div className="space-y-4">
637:             <div className="flex items-center justify-between">
638:                 <h4 className="text-base sm:text-lg font-medium text-gray-700 border-l-4 border-green-400">
639:                     {getSectionTitle(activeSection)}
640:                 </h4>
641:                 <span className="text-xs text-gray-500 bg-gray-100 px-2 py-1 rounded">
642:                     {sportsAndFitness[activeSection] &&
643:                     typeof sportsAndFitness[activeSection] === "object" &&
644:                     !Array.isArray(sportsAndFitness[activeSection])
645:                     ? Object.keys(sportsAndFitness[activeSection]).length
646:                     : 0}{ " "}
647:                 items
648:             </span>
649:         </div>
650:
651:         <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-4">
652:             {sportsAndFitness[activeSection] &&
653:             typeof sportsAndFitness[activeSection] === "object" &&
654:             !Array.isArray(sportsAndFitness[activeSection]) &&
655:             Object.entries(sportsAndFitness[activeSection]).map(
656:                 ([fieldKey, fieldData]) => {
657:                     if (
658:                         typeof fieldData === "object" &&
659:                         "level" in fieldData &&
660:                         "description" in fieldData
661:                     ) {
662:                         return renderExerciseField(
663:                             activeSection,
664:                             fieldKey,
665:                             fieldData
666:                         );
667:                     }
668:                     return null;
669:                 }
670:             )}
671:         </div>
672:     </div>
673: )}
674:
675: {(!activeSection ||
676: !sportsAndFitness[activeSection] ||
677: typeof sportsAndFitness[activeSection] !== "object" ||
678: Array.isArray(sportsAndFitness[activeSection]) ||
679: Object.keys(sportsAndFitness[activeSection]).length === 0) && (
680: <div className="text-center py-12 text-gray-500">
681:     <div className="text-4xl mb-4">?</div>
682:     <p className="text-lg font-medium mb-2">No fields available</p>
683:     <p className="text-sm">
684:         {!activeSection
685:         ? "Select a category or add one to get started"
686:         : "Add some fields to get started"}

```



```

687:         </p>
688:     </div>
689:   )}
690: </div>
691: </CardContent>
692: </Card>
693:
694: <Dialog open={showImageViewer} onOpenChange={setShowImageViewer}>
695:   <DialogContent className="max-w-xl max-h-[90vh] overflow-y-auto">
696:     <DialogHeader>
697:       <DialogTitle>Uploaded Custom Images</DialogTitle>
698:     </DialogHeader>
699:
700:     {Object.entries(customImages).length === 0 ? (
701:       <div className="text-center text-gray-500 py-6">
702:         No images uploaded.
703:       </div>
704:     ) : (
705:       <ul className="space-y-3 mt-4">
706:         {Object.entries(customImages).map(([label, url]) => (
707:           <li
708:             key={label}
709:             className="flex items-center justify-between bg-gray-50 border rounded p-2"
710:           >
711:             <div className="flex items-center gap-3">
712:               <img
713:                 src={url}
714:                 alt={label}
715:                 className="w-12 h-12 rounded object-cover border"
716:               />
717:               <span className="text-sm font-medium truncate max-w-[150px] sm:max-w-none">
718:                 {label}
719:               </span>
720:             </div>
721:
722:             <Button
723:               variant="destructive"
724:               size="sm"
725:               onClick={async () => {
726:                 const confirm = window.confirm(
727:                   `Delete image "${label}"?`
728:                 );
729:                 if (!confirm) return;
730:
731:                 const fileName = url.split("/").pop();
732:                 try {
733:                   const res = await fetch(
734:                     `/api/sports-images?file=${fileName}`,
735:                     {
736:                       method: "DELETE",
737:                     }
738:                   );
739:                   const result = await res.json();
740:
741:                   if (result.success) {
742:                     const updated = { ...customImages };
743:                     delete updated[label];
744:                     setCustomImages(updated);
745:
746:                     updateSportsAndFitness("customImages" as any, label, {
747:                       label,
748:                       url: "",
749:                     });
750:
751:                     toast({
752:                       title: "Image Deleted",
753:                       description: `Successfully deleted image "${label}"`,
754:                       duration: 1500,
755:                       variant: "success",
756:                     });
757:                   } else {
758:                     toast({
759:                       title: "Delete Failed",

```

```

760:                 description:
761:                     result.error || "Failed to delete image",
762:                     variant: "destructive",
763:                     duration: 1500,
764:                 });
765:             }
766:         } catch (err) {
767:             console.error("Deletion error", err);
768:             toast({
769:                 title: "Delete Error",
770:                 description: "Error occurred while deleting image",
771:                 variant: "destructive",
772:                 duration: 1500,
773:             });
774:         }
775:     }}
776: >
777:     <Trash2 className="h-4 w-4" />
778: </Button>
779: </li>
780: )}}
781: </ul>
782: )}
783: </DialogContent>
784: </Dialog>
785:
786: { /* Delete Confirmation Dialog */}
787: <AlertDialog
788:   open={deleteConfirm.type !== null}
789:   onOpenChange={() => setDeleteConfirm({ type: null, target: "" })}
790: >
791:   <AlertDialogContent className="w-[90%] rounded-md max-w-md sm:max-w-lg md:max-w-xl lg:max-w-2xl p-4"
792:     <AlertDialogHeader>
793:       <AlertDialogTitle className="text-base sm:text-lg md:text-xl">
794:         Confirm Deletion
795:       </AlertDialogTitle>
796:       <AlertDialogDescription className="text-sm sm:text-base leading-relaxed">
797:         {deleteConfirm.type === "category" ? (
798:           <>
799:             Are you sure you want to delete the category{" "}
800:             <span className="text-red-600 font-medium">
801:               {deleteConfirm.target}
802:             </span>
803:             ? This action cannot be undone and will remove all fields
804:             within this category.
805:           </>
806:         ) : (
807:           <>
808:             Are you sure you want to delete the field{" "}
809:             <span className="text-red-600 font-medium">
810:               {deleteConfirm.target}
811:             </span>
812:             ? This action cannot be undone.
813:           </>
814:         )}
815:       </AlertDialogDescription>
816:     </AlertDialogHeader>
817:
818:     <AlertDialogFooter className="flex flex-col-reverse sm:flex-row sm:justify-end gap-2 sm:gap-3 mt-2"
819:       <AlertDialogCancel className="w-full sm:w-auto">
820:         Cancel
821:       </AlertDialogCancel>
822:       <AlertDialogAction
823:         onClick={confirmDelete}
824:         className="bg-red-600 hover:bg-red-700 w-full sm:w-auto"
825:       >
826:         Delete
827:       </AlertDialogAction>
828:     </AlertDialogFooter>
829:   </AlertDialogContent>
830: </AlertDialog>
831: </div>
832: );

```

833: }

## ■ File: components\comprehensive-report-viewer.tsx

```
=====
1: "use client";
2: import type { ComprehensiveReportData } from "@/types/report-types";
3: import React from "react";
4: import { Card, CardContent, CardHeader, CardTitle } from "@/components/ui/card";
5: import { Separator } from "@/components/ui/separator";
6: import { Check, X } from "lucide-react";
7: import {
8:   BarChart,
9:   Bar,
10:  Cell,
11:  LineChart,
12:  Line,
13:  XAxis,
14:  YAxis,
15:  Tooltip,
16:  ResponsiveContainer,
17: } from "recharts";
18:
19: import {
20:   RadarChart,
21:   Radar,
22:   PolarGrid,
23:   PolarAngleAxis,
24:   PolarRadiusAxis,
25:   PieChart,
26:   Pie,
27: } from "recharts";
28:
29: interface HealthConditionStatus {
30:   status: "strength" | "improvement";
31:   description?: string;
32:   sensitivity?: string;
33:   avoid?: string[];
34:   follow?: string[];
35:   consume?: string[];
36:   monitor?: string[];
37:   avoidLabel?: string;
38:   followLabel?: string;
39:   consumeLabel?: string;
40:   monitorLabel?: string;
41: }
42:
43: const defaultImageMap: Record<string, string> = {
44:   high: "/sports/high.png",
45:   low: "/sports/low.png",
46:   average: "/sports/average.png",
47:   normal: "/sports/normal.png",
48:   enhanced: "/sports/enhanced.png",
49: };
50:
51: interface ComprehensiveReportViewerProps {
52:   reportData: ComprehensiveReportData;
53: }
54:
55: const SectionTitle = ({ title, icon }: { title: string; icon: string }) => (
56:   <div className="w-full mb-3 mt-4 first:mt-0">
57:     <h2 className="text-base font-bold text-gray-800 mb-2 flex items-center justify-center gap-2">
58:       {icon && <span className="text-lg">{icon}</span>}
59:       <span className="break-words text-center">{title}</span>
60:     </h2>
61:   </div>
62: );
63:
64: const SubSectionTitle = ({ title, icon }: { title: string; icon?: string }) => (
65:   <div className="w-full mb-2 mt-3">
66:     <h3 className="text-sm font-semibold text-gray-700 mb-1 flex items-center justify-center gap-2">
67:       {icon && <span className="text-base">{icon}</span>}
```

```

68:     <span className="break-words text-center">{title}</span>
69:   </h3>
70: </div>
71: );
72:
73: const DataCard = ({
74:   title,
75:   children,
76:   className = "",
77: }): {
78:   title: string;
79:   children: React.ReactNode;
80:   className?: string;
81: }) => (
82:   <Card
83:     className={`border border-gray-200 bg-gray-50/50 shadow-sm h-fit min-w-0 ${className}`}
84:   >
85:     <CardHeader className="pb-1">
86:       <CardTitle className="text-xs font-semibold text-gray-800 break-words leading-tight">
87:         {title}
88:       </CardTitle>
89:     </CardHeader>
90:     <CardContent className="text-xs text-gray-700 space-y-1">
91:       {children}
92:     </CardContent>
93:   </Card>
94: );
95:
96: export default function ComprehensiveReportViewer({
97:   reportData,
98: }: ComprehensiveReportViewerProps) {
99:   if (!reportData) {
100:     return (
101:       <div className="w-full max-w-4xl mx-auto py-4 px-3 bg-white shadow-lg min-h-screen flex items-center
102:         <div className="text-center">
103:           <h2 className="text-lg font-bold text-gray-800 mb-3">
104:             No Report Data Available
105:           </h2>
106:           <p className="text-gray-600 text-sm">
107:             Please provide valid report data to display the comprehensive
108:             report.
109:           </p>
110:         </div>
111:       </div>
112:     );
113:   }
114:
115:   const {
116:     patientInfo = {
117:       name: "",
118:       gender: "",
119:       birthDate: "",
120:       sampleCode: "",
121:       sampleDate: "",
122:       reportDate: "",
123:       signature1: "",
124:       signature2: "",
125:       checkedBy: "",
126:       scientificContent: "",
127:       disclaimer: "",
128:       reportAuth: "",
129:     },
130:     content = {
131:       introduction: "",
132:       genomicsExplanation: "",
133:       genesHealthImpact: "",
134:       fundamentalsPRS: "",
135:       utilityDoctors: "",
136:       microarrayExplanation: "",
137:       microarrayData: "",
138:     },
139:     settings = {
140:       title: "Comprehensive Genetic Report",

```

```

141:     subtitle: "Personalized Health Analysis",
142:     headerColor: "#3B82F6",
143:     companyName: "Genetic Health Solutions",
144:     fonts: { primary: "Arial, sans-serif", secondary: "Arial, sans-serif" },
145: },
146: dynamicDietFieldDefinitions = [
147:   {
148:     meta: { quote: "", description: "" },
149:     fields: [],
150:   },
151: ],
152: patientDietAnalysisResults = [],
153: nutritionData = { data: {} },
154: sportsAndFitness = {
155:   customImages: {},
156:   quote: undefined,
157:   description: undefined,
158: },
159: lifestyleConditions = {},
160: metabolicCore = {},
161: digestiveHealth = { data: {} },
162: genesAndAddiction = { data: {} },
163: sleepAndRest = { data: {} },
164: allergiesAndSensitivity = { data: {} },
165: geneTestResults = [],
166: preventiveHealth = {
167:   diagnosticTests: { halfYearly: [], yearly: [] },
168:   nutritionalSupplements: [],
169: },
170: familyGeneticImpactSection = {
171:   description: "",
172:   impacts: [],
173: },
174: summaries = { nutrigenomicsSummary: "", exerciseGenomicsSummary: "" },
175: categories = [],
176: healthSummary = { description: "", data: [] },
177: } = reportData;
178:
179: type DietFieldResult = (typeof patientDietAnalysisResults)[0] & {
180:   label: string;
181:   score: number;
182: };
183:
184: const groupedDietFields = patientDietAnalysisResults.reduce((acc, result) => {
185:   const allFields = dynamicDietFieldDefinitions.flatMap(
186:     (def) => def.fields || []
187:   );
188:   const fieldDef = allFields.find((field) => field.id === result.fieldId);
189:
190:   if (fieldDef) {
191:     if (!acc[fieldDef.category]) acc[fieldDef.category] = [];
192:     acc[fieldDef.category].push({
193:       ...result,
194:       label: fieldDef.label,
195:       score: result.score ?? 0,
196:
197:       recommendations: {
198:         LOW: fieldDef.lowRecommendation,
199:         NORMAL: fieldDef.normalRecommendation,
200:         HIGH: fieldDef.highRecommendation,
201:       },
202:
203:       lowRecommendation: fieldDef.lowRecommendation,
204:       normalRecommendation: fieldDef.normalRecommendation,
205:       highRecommendation: fieldDef.highRecommendation,
206:     });
207:   }
208:   return acc;
209: }, {} as Record<string, any[]>);
210:
211: return (
212:   <>
213:     <style jsx global>{`

```

```

214:     @media print {
215:         /* Make PDF one tall page instead of multiple A4 pages */
216:         @page {
217:             size: auto; /* Automatically fit content height */
218:             margin: 0; /* No print margins */
219:         }
220:         html,
221:         body {
222:             height: auto !important;
223:             font-size: 9px;
224:             line-height: 1.2;
225:         }
226:         .print\\:\\:hidden {
227:             display: none !important;
228:         }
229:     }
230:     @media screen {
231:         .report-container {
232:             max-width: 210mm; /* Keep same on-screen width */
233:             margin: 0 auto;
234:             background: white;
235:             // border: 1px solid gray;
236:             border-radius: 10px;
237:             box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
238:         }
239:     }
240: `}</style>
241:
242: <div className="report-container w-full mx-auto py-3 px-3 bg-white shadow-lg min-h-screen">
243:     {/* Report Header */}
244:     <header
245:         className="w-full max-w-[210mm] mx-auto min-h-[297mm] flex flex-col justify-between rounded-lg px-3"
246:         style={{ backgroundColor: settings.headerColor, color: "#fff" }}
247:     >
248:         {/* Top Section - Company/Title */}
249:         <div className="text-center space-y-4 sm:space-y-6 mt-4 sm:mt-6 md:mt-8 print:mt-16">
250:             {/* Main Title */}
251:             <div className="space-y-2 sm:space-y-3 md:space-y-4">
252:                 <h1
253:                     className="text-2xl sm:text-3xl md:text-4xl lg:text-5xl font-extrabold mb-2 sm:mb-3 md:mb-4"
254:                     style={{ fontFamily: settings.fonts.primary }}
255:                 >
256:                     {settings.title}
257:                 </h1>
258:                 <div className="w-20 sm:w-24 md:w-32 h-0.5 sm:h-0.5 md:h-1 bg-white mx-auto opacity-80 rounded"
259:                 >
260:                     <p
261:                         className="text-base sm:text-lg md:text-xl lg:text-2xl font-light mb-4 sm:mb-6 md:mb-8 lead"
262:                         style={{ fontFamily: settings.fonts.secondary }}
263:                     >
264:                         {settings.subtitle}
265:                     </p>
266:                 </div>
267:             </div>
268:             {/* Middle Section - Patient Information Card */}
269:             <div className="flex-1 flex items-center justify-center px-2 sm:px-4 md:px-0">
270:                 <div className="bg-white bg-opacity-95 text-gray-800 rounded-xl sm:rounded-2xl p-4 sm:p-6 md:p-8"
271:                 >
272:                     <h2
273:                         className="text-lg sm:text-xl md:text-2xl font-bold text-center mb-4 sm:mb-6 md:mb-8 print:mt-16"
274:                         style={{
275:                             fontFamily: settings.fonts.primary,
276:                             color: settings.headerColor,
277:                         }}
278:                     >
279:                         Patient Information
280:                     </h2>
281:                     <div className="space-y-4 sm:space-y-5 md:space-y-6 print:space-y-4">
282:                         {/* Patient Name */}
283:                         <div className="text-center border-b border-gray-200 pb-3 sm:pb-4 print:pb-4">
284:                             <p
285:                                 className="text-xs sm:text-sm font-medium text-gray-600 uppercase tracking-wide mb-1 sm:mb-2"
286:                                 style={{ fontFamily: settings.fonts.secondary }}

```

```

287:         >
288:         Patient Name
289:     </p>
290:     <p
291:         className="text-lg sm:text-xl md:text-2xl font-bold break-words leading-tight px-2 print:px-4"
292:         style={{
293:             fontFamily: settings.fonts.primary,
294:             color: settings.headerColor,
295:         }}
296:     >
297:         {patientInfo.name}
298:     </p>
299: </div>
300:
301:     { /* Sample Code */ }
302:     <div className="text-center border-b border-gray-200 pb-3 sm:pb-4 md:pb-4 print:pb-6">
303:         <p
304:             className="text-xs sm:text-sm font-medium text-gray-600 uppercase tracking-wide mb-1 sm:mb-2"
305:             style={{ fontFamily: settings.fonts.secondary }}
306:         >
307:             Sample Code
308:         </p>
309:         <p
310:             className="text-base sm:text-lg md:text-xl font-mono font-bold break-all tracking-wider"
311:             style={{ color: settings.headerColor }}
312:         >
313:             {patientInfo.sampleCode}
314:         </p>
315:     </div>
316:
317:     { /* Report Details */ }
318:     <div className="text-center space-y-2 sm:space-y-3 print:space-y-4">
319:         <div>
320:             <p
321:                 className="text-xs sm:text-sm font-medium text-gray-600 uppercase tracking-wide mb-1 sm:mb-2"
322:                 style={{ fontFamily: settings.fonts.secondary }}
323:             >
324:                 Report Date
325:             </p>
326:             <p
327:                 className="text-sm sm:text-base md:text-lg font-semibold print:text-xl"
328:                 style={{
329:                     fontFamily: settings.fonts.secondary,
330:                     color: settings.headerColor,
331:                 }}
332:             >
333:                 {patientInfo.reportDate}
334:             </p>
335:         </div>
336:     </div>
337: </div>
338: </div>
339: </div>
340:
341: { /* Bottom Section - Company Info */ }
342: <div className="relative flex flex-col items-center space-y-2 sm:space-y-3 md:space-y-4 mb-4 sm:mb-6">
343:     <div className="w-16 sm:w-20 md:w-24 h-0.5 bg-white mx-auto opacity-60 rounded print:w-32"></div>
344:
345:     <p
346:         className="text-sm sm:text-base md:text-lg font-medium tracking-wide print:text-xl"
347:         style={{ fontFamily: settings.fonts.secondary }}
348:     >
349:         Generated by
350:     </p>
351:     <p
352:         className="text-xl sm:text-xl md:text-2xl font-bold tracking-wide print:text-3xl"
353:         style={{ fontFamily: settings.fonts.primary }}
354:     >
355:         {settings.companyName}
356:     </p>
357:     <p
358:         className="text-xs sm:text-sm opacity-90 px-4 print:text-base"
359:         style={{ fontFamily: settings.fonts.secondary }}

```

```

360:         >
361:         Professional Health Report ? Confidential
362:     </p>
363:
364:     <div className="flex justify-between items-end w-full px-4 mt-8">
365:         
370:         
375:     </div>
376: </div>
377: </header>
378:
379: <Separator className="my-6" />
380:
381: { /* Patient Information */}
382: <div className="bg-white border border-gray-300 rounded-lg shadow-sm p-6 mb-6 print:shadow-none print:outline-1">
383:     { /* Section Header */}
384:     <div className="flex items-center mb-6 pb-3 border-b border-gray-200 print:mb-4 print:pb-2">
385:         <div className="w-1 h-6 bg-orange-500 rounded-full mr-3 print:bg-black"></div>
386:         <h3 className="text-lg font-bold text-gray-800 uppercase tracking-wide print:text-base print:font-normal">
387:             Personal Details
388:         </h3>
389:     </div>
390:
391:     { /* Modern Table-Style Grid */}
392:     <div className="bg-gradient-to-br rounded-lg from-gray-50 to-gray-100 overflow-hidden border border-gray-200">
393:         <div className="divide-y divide-gray-200 print:divide-gray-600">
394:             { /* Patient Name Row */}
395:             <div className="flex items-center py-4 px-6 hover:bg-white/50 transition-all duration-200 print:transition-none">
396:                 <div className="w-2/5">
397:                     <label className="text-sm font-bold text-gray-700 uppercase tracking-wide print:text-xs print:font-normal">
398:                         Name
399:                     </label>
400:                 </div>
401:                 <div className="w-px h-8 bg-gray-300 print:bg-gray-600 mx-4"></div>
402:                 <div className="flex-1">
403:                     <p className="text-base font-semibold text-gray-900 break-words print:text-sm print:font-normal">
404:                         {patientInfo.name}
405:                     </p>
406:                 </div>
407:             </div>
408:
409:             { /* Gender Row */}
410:             <div className="flex items-center py-4 px-6 hover:bg-white/50 transition-all duration-200 print:transition-none">
411:                 <div className="w-2/5">
412:                     <label className="text-sm font-bold text-gray-700 uppercase tracking-wide print:text-xs print:font-normal">
413:                         Gender
414:                     </label>
415:                 </div>
416:                 <div className="w-px h-8 bg-gray-300 print:bg-gray-600 mx-4"></div>
417:                 <div className="flex-1">
418:                     <p className="text-base font-semibold text-gray-900 capitalize print:text-sm print:font-normal">
419:                         {patientInfo.gender}
420:                     </p>
421:                 </div>
422:             </div>
423:
424:             { /* Birth Date Row */}
425:             <div className="flex items-center py-4 px-6 hover:bg-white/50 transition-all duration-200 print:transition-none">
426:                 <div className="w-2/5">
427:                     <label className="text-sm font-bold text-gray-700 uppercase tracking-wide print:text-xs print:font-normal">
428:                         Birth Date
429:                     </label>
430:                 </div>
431:                 <div className="w-px h-8 bg-gray-300 print:bg-gray-600 mx-4"></div>
432:                 <div className="flex-1">

```



```

443:         <p className="text-base font-semibold text-gray-900 print:text-sm print:font-bold">
444:             {patientInfo.birthDate}
445:         </p>
446:     </div>
447: </div>
448:
449: { /* Sample Code Row */}
450: <div className="flex items-center py-4 px-6 hover:bg-white/50 transition-all duration-200 pri
451:     <div className="w-2/5">
452:         <label className="text-sm font-bold text-gray-700 uppercase tracking-wide print:text-xs p
453:             Sample Code
454:         </label>
455:     </div>
456:     <div className="w-px h-8 bg-gray-300 print:bg-gray-600 mx-4"></div>
457:     <div className="flex-1">
458:         <p className="text-base font-mono font-bold text-orange-600 break-all tracking-wide print
459:             {patientInfo.sampleCode}
460:         </p>
461:     </div>
462: </div>
463:
464: { /* Sample/Data Date Row */}
465: <div className="flex items-center py-4 px-6 hover:bg-white/50 transition-all duration-200 pri
466:     <div className="w-2/5">
467:         <label className="text-sm font-bold text-gray-700 uppercase tracking-wide print:text-xs p
468:             Sample Collection Date
469:         </label>
470:     </div>
471:     <div className="w-px h-8 bg-gray-300 print:bg-gray-600 mx-4"></div>
472:     <div className="flex-1">
473:         <p className="text-base font-semibold text-gray-900 print:text-sm print:font-bold">
474:             {patientInfo.sampleDate}
475:         </p>
476:     </div>
477: </div>
478:
479: { /* Report Date Row */}
480: <div className="flex items-center py-4 px-6 hover:bg-white/50 transition-all duration-200 pri
481:     <div className="w-2/5">
482:         <label className="text-sm font-bold text-gray-700 uppercase tracking-wide print:text-xs p
483:             Report Date
484:         </label>
485:     </div>
486:     <div className="w-px h-8 bg-gray-300 print:bg-gray-600 mx-4"></div>
487:     <div className="flex-1">
488:         <p className="text-base font-semibold text-gray-900 print:text-sm print:font-bold">
489:             {patientInfo.reportDate}
490:         </p>
491:     </div>
492: </div>
493: </div>
494: </div>
495:
496: { /* Authentication Section */}
497: <div className="mt-6 mb-6">
498:     <Separator className="mb-4" />
499:     <h3 className="text-center font-bold mb-4 text-gray-600 text-sm">
500:         Report Authentication & Analytics
501:     </h3>
502:     <div className="grid grid-cols-2 gap-6 mb-5">
503:         <div className="text-center space-y-3">
504:             <p className="text-xs font-semibold text-gray-700">
505:                 Genomic Data Analytics - Checked & Verified By
506:             </p>
507:             <div className="flex justify-center py-2">
508:                 {patientInfo.signature1 ? (
509:                     <img
510:                         src={patientInfo.signature1 || "/placeholder.svg"}
511:                         alt="Primary Signature"
512:                         className="max-h-12 w-auto object-contain"
513:                     />
514:                 ) : (

```

```

506:         <p className="italic text-gray-500 text-xs">
507:             No signature available
508:         </p>
509:     )}
510: </div>
511: <p className="text-xs break-words text-gray-600">
512:     {patientInfo.checkedBy}
513: </p>
514: </div>
515: <div className="text-center space-y-3">
516:     <p className="text-xs font-semibold text-gray-700">
517:         Scientific Content - Checked & Verified By
518:     </p>
519:     <div className="flex justify-center py-2">
520:         {patientInfo.signature2 ? (
521:             <img
522:                 src={patientInfo.signature2 || "/placeholder.svg"}
523:                 alt="Secondary Signature"
524:                 className="max-h-12 w-auto object-contain"
525:             />
526:         ) : (
527:             <p className="italic text-gray-500 text-xs">
528:                 No signature available
529:             </p>
530:         )}
531:     </div>
532:     <p className="text-xs break-words text-gray-600">
533:         {patientInfo.scientificContent}
534:     </p>
535: </div>
536: </div>
537: <DataCard title="Disclaimer" className="text-center">
538:     <p className="text-justify whitespace-pre-wrap break-words text-xs leading-relaxed">
539:         {patientInfo.disclaimer}
540:     </p>
541: </DataCard>
542: </div>
543:
544: <Separator className="my-4" />
545: { /* Preface Section */ }
546: <div className="text-center mb-3">
547:     <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
548:         PREFACE
549:     </h2>
550: </div>
551:
552: <div className="m-3 w-1/2">
553:     {content.introduction.split(/\n\s*\n/).map((para, idx) => (
554:         <p
555:             key={idx}
556:             className="text-xs mb-4"
557:             style={{ textAlign: "justify", textJustify: "inter-word" }}
558:         >
559:             {para}
560:         </p>
561:     ) )}
562: </div>
563:
564: <div className="space-y-3 mb-4">
565:     <div className="text-center mb-3">
566:         <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
567:             WELCOME TO THE WORLD OF GENOMICS
568:         </h2>
569:     </div>
570:
571:     <div className="grid grid-cols-2 gap-3 mb-4">
572:         <DataCard title="What is Genomics?">
573:             <p className="text-justify whitespace-pre-line text-xs">
574:                 {content.genomicsExplanation}
575:             </p>
576:         </DataCard>
577:         <div className="flex justify-center items-center">
578:             
583: </div>
584: </div>
585:
586: <div className="grid grid-cols-2 gap-3 mb-4">
587:     <DataCard title="Genes and Health Impact">
588:         <p className="text-justify whitespace-pre-line text-xs">
589:             {content.genesHealthImpact}
590:         </p>
591:     </DataCard>
592:     <div className="flex justify-center items-center">
593:         
598:     </div>
599: </div>
600:
601: <Separator className="my-4" />
602:
603: <div className="text-center mb-3">
604:     <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
605:         WHAT IS NUTRIGENOMICS
606:     </h2>
607: </div>
608:
609: <div className="grid grid-cols-2 gap-3 mb-4">
610:     <DataCard title="Fundamentals and PRS">
611:         <p className="text-justify whitespace-pre-line text-xs ">
612:             {content.fundamentalsPRS}
613:         </p>
614:     </DataCard>
615:     <div className="flex justify-center items-center ">
616:         
621:     </div>
622: </div>
623:
624: <div className="grid grid-cols-2 gap-3 mb-4">
625:     <DataCard title="Utility for Doctors and Dietitians">
626:         <p className="text-justify whitespace-pre-line text-xs">
627:             {content.utilityDoctors}
628:         </p>
629:     </DataCard>
630:     <div className="flex justify-center items-center ">
631:         
636:     </div>
637: </div>
638:
639: <Separator className="my-4" />
640:
641: <div className="text-center mb-3">
642:     <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
643:         ABOUT MICROARRAY
644:     </h2>
645: </div>
646:
647: <div className="grid grid-cols-2 gap-3 mb-4">
648:     <div>
649:         <DataCard title="Microarray Explanation">
650:             <p className="text-justify whitespace-pre-line text-xs">
651:                 {content.microarrayExplanation}

```

```

652:         </p>
653:     </DataCard>
654:     <DataCard title="Your Microarray Data" className="mt-4">
655:         <p className="text-justify whitespace-pre-line text-xs">
656:             {content.microarrayData}
657:         </p>
658:     </DataCard>
659: </div>
660:
661: <div className="flex justify-center items-center">
662:     
667: </div>
668: </div>
669: </div>
670: <Separator className="my-4" />
671: { /* Table of Contents */ }
672: <div className="text-center mb-3">
673:     <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded uppercase">
674:         Area Covered in the Report
675:     </h2>
676: </div>
677: {categories?.map((category) => {
678:     const columns: string[][] = [[], [], []];
679:     category.parameters.forEach((param, index) => {
680:         columns[index % 3].push(param);
681:     });
682:
683:     return (
684:         <div
685:             key={category.id}
686:             className="mb-4 border-b border-gray-300 pb-3"
687:         >
688:             <div className="flex gap-3">
689:                 <div className="w-1/4 flex flex-col items-center ml-3 justify-center gap-2 mb-2 min-h-20 md">
690:                     {category.imageUrl && (
691:                         <img
692:                             src={category.imageUrl || "/placeholder.svg"}
693:                             alt={category.category}
694:                             className="w-12 h-12 object-cover flex-shrink-0"
695:                         />
696:                     )}
697:                     <h2 className="p-2 sm:px-3 text-xs sm:text-sm md:text-base lg:text-sm font-bold text-gray">
698:                         {category.category}
699:                     </h2>
700:                 </div>
701:                 <div className="w-3/4">
702:                     <div className="grid grid-cols-3 gap-2">
703:                         {columns.map((col, colIndex) => (
704:                             <ul key={colIndex} className="list-none space-y-1">
705:                                 {col.map((param, idx) => (
706:                                     <li
707:                                         key={idx}
708:                                         className="text-gray-800 text-xs break-words"
709:                                     >
710:                                         {param}
711:                                     </li>
712:                                 ) )}
713:                             </ul>
714:                         ) )}
715:                     </div>
716:                 </div>
717:             </div>
718:         </div>
719:     );
720: })}
721:
722: <Separator className="my-4" />
723:
724: { /* Health Summary Section */ }

```

```

725:     {(healthSummary.description || healthSummary.data.length > 0) && (
726:       <div className="space-y-4">
727:         {/* Section Title */}
728:         <div className="text-center mb-3">
729:           <h2 className="text-base font-bold mb-2 bg-blue-300 text-white py-2 px-2 rounded uppercase">
730:             DISTRIBUTION OF YOUR GENOMIC ANALYSIS
731:           </h2>
732:         </div>
733:
734:         {/* Top description */}
735:         {healthSummary.description && (
736:           <p className="text-gray-700 text-sm text-justify p-3">
737:             {healthSummary.description}
738:           </p>
739:         )}
740:
741:         {/* Two-column table entries */}
742:         {healthSummary.data.map((entry, index) => (
743:           <div
744:             key={`health-summary-${index}`}
745:             className="grid grid-cols-12 gap-0 w-full"
746:           >
747:             {/* Left Column: Title */}
748:             <div className="col-span-3 flex items-center justify-center p-2 sm:p-4">
749:               <span
750:                 className={`text-white font-bold text-[10px] sm:text-sm px-2 sm:px-5 py-4 sm:py-7 rounded
751: $ {
752:   entry.title.toLowerCase() === "strengths"
753:     ? "bg-green-500"
754:     : entry.title.toLowerCase() === "weaknesses"
755:     ? "bg-orange-500"
756:     : entry.title.toLowerCase().includes("diet")
757:     ? "bg-blue-500"
758:     : entry.title.toLowerCase().includes("exercise")
759:     ? "bg-gray-400"
760:     : entry.title.toLowerCase().includes("preventive")
761:     ? "bg-yellow-500 text-black"
762:     : "bg-blue-500"
763: }}`
764:               >
765:                 {entry.title}
766:               </span>
767:             </div>
768:
769:             {/* Right Column: Description */}
770:             <div className="col-span-9 border border-gray-400 p-2 sm:p-4 flex flex-col justify-center r
771: {entry.description &&
772:   (() => {
773:     const lines = entry.description
774:       .split("\n")
775:       .map((l) => l.trim())
776:       .filter(Boolean);
777:     const isBulletList =
778:       lines.length > 1 &&
779:       (lines.every((line) => /^[^-\d]/.test(line)) ||
780:         lines.length <= 6);
781:
782:     return isBulletList ? (
783:       <ul className="list-disc list-inside text-gray-700 text-xs sm:text-sm normal-case l
784:         {lines.map((line, i) => (
785:           <li key={i}>{line.replace(/^[^-\d. ]+/, "")}</li>
786:         ))}
787:       </ul>
788:     ) : (
789:       <p className="text-gray-700 text-xs sm:text-sm normal-case leading-relaxed text-jus
790:         {entry.description}
791:       </p>
792:     );
793:   })()}
794:           </div>
795:         </div>
796:       )}
797:     </div>

```

```

798:     })
799:
800:     <Separator className="my-4" />
801:
802:     <div className="w-full max-w-4xl">
803:       <div className="relative w-full" style={{ paddingBottom: "141.4%" }}>
804:         
809:       </div>
810:     </div>
811:
812:     <Separator className="my-4" />
813:
814:     { /* Diet Section */ }
815:     <div className="text-center mb-3">
816:       <h2 className="text-base font-bold mb-2 bg-green-600 text-white py-2 px-2 rounded">
817:         GENOMICS & DIET
818:       </h2>
819:     </div>
820:     {Object.keys(groupedDietFields).length === 0 && (
821:       <p className="text-center text-gray-500 italic mb-4">
822:         No diet analysis results available.
823:       </p>
824:     )}
825:
826:     {reportData?.dynamicDietFieldDefinitions?.[0]?.meta?.quote && (
827:       <div className="mb-3 text-center">
828:         <p className="text-sm font-semibold italic text-gray-800">
829:           "{reportData.dynamicDietFieldDefinitions[0].meta.quote}"
830:         </p>
831:       </div>
832:     )}
833:
834:     {reportData?.dynamicDietFieldDefinitions?.[0]?.meta?.description && (
835:       <div className="mb-3">
836:         <p className="text-gray-700 p-3 text-xs text-justify">
837:           {reportData.dynamicDietFieldDefinitions[0].meta.description}
838:         </p>
839:       </div>
840:     )}
841:
842:     <div className="w-full max-w-4xl mx-auto mb-6 print:mb-0 print:max-w-none print:w-full">
843:       <div className="overflow-x-auto overflow-y-hidden">
844:         <table className="w-full min-w-[700px] table-fixed text-xs border-2 border-gray-300 bg-white pr
845:         <tbody>
846:           {Object.entries(groupedDietFields).flatMap(
847:             ([category, fields]) => {
848:               fields.map((data, index) => {
849:                 const score = Number(data.score ?? 0);
850:
851:                 // Determine selected level from score
852:                 let selectedLevel: "LOW" | "NORMAL" | "HIGH" = "NORMAL";
853:                 if (score <= 3) selectedLevel = "LOW";
854:                 else if (score >= 7) selectedLevel = "HIGH";
855:
856:                 const scoreColor =
857:                   score <= 3
858:                     ? "bg-green-600 print:bg-green-700"
859:                     : score <= 6
860:                     ? "bg-yellow-500 print:bg-yellow-600"
861:                     : "bg-red-600 print:bg-red-600";
862:
863:                 const renderAdviceCell = (
864:                   level: "LOW" | "NORMAL" | "HIGH"
865:                 ) => (
866:                   <td className="p-3 border border-gray-300 align-top w-[17%] print:p-2 print:text-[7
867:                     <div
868:                       className={`text-xs font-bold mb-2 uppercase tracking-wide text-center print:te
869:                       selectedLevel === level
870:                       ? "text-black"

```

```

871:         : "text-gray-400 print:text-gray-500"
872:     }` }
873: >
874:     {level}
875: </div>
876: <div
877:     className={`text-xs leading-relaxed text-center print:text-[6px] print:leading-
878:         selectedLevel === level
879:         ? "text-gray-900 font-medium"
880:         : "text-gray-400 print:text-gray-500"
881:     }` }
882:     style={{
883:         wordBreak: "break-word",
884:         overflowWrap: "break-word",
885:         hyphens: "auto",
886:     }}
887: >
888:     {(() => {
889:         // Prefer new flat field (e.g. lowRecommendation), fallback to legacy structur
890:         const key = (level.toLowerCase() +
891:             "Recommendation") as
892:             | "lowRecommendation"
893:             | "normalRecommendation"
894:             | "highRecommendation";
895:         return (
896:             (key in data
897:                 ? (data as any)[key]
898:                 : undefined) ||
899:             data?.recommendations?.[level] ||
900:             "-"
901:         );
902:     })()}
903: </div>
904: </td>
905: );
906:
907: return (
908:     <tr
909:         key={`${category}-${index}`}
910:         className="border-t border-gray-100 print:break-inside-avoid"
911:     >
912:         { /* Category cell - show only once per group */ }
913:         { index === 0 && (
914:             <td
915:                 rowspan={fields.length}
916:                 className="p-2 border border-gray-300 font-bold text-blue-800 bg-gray-50 text-
917:             >
918:                 <div className="text-center">
919:                     {category
920:                         .replace(/([a-z])([A-Z])/g, "$1 $2")
921:                         .replace(/([A-Z])([A-Z][a-z])/g, "$1 $2")
922:                         .replace(/&/g, " & ")
923:                         .replace(/\\s+/g, " ")
924:                         .trim()}
925:                     </div>
926:                 </td>
927:             )}
928:
929:             { /* Trait name */ }
930:             <td className="p-3 border border-gray-300 text-center font-semibold text-gray-800
931:                 <div
932:                     className="flex items-center justify-center break-words overflow-hidden text-
933:                     style={{
934:                         wordBreak: "break-word",
935:                         overflowWrap: "break-word",
936:                         hyphens: "auto",
937:                     }}
938:                 >
939:                     {data.label}
940:                 </div>
941:             </td>
942:
943:             { /* Score badge */ }

```

```

944:         <td className="p-3 border border-gray-300 text-center align-middle w-[10%] print:
945:             <span
946:                 className={`inline-block px-5 py-1 rounded-md text-white font-bold text-sm mi
947:             >
948:                 {score}
949:             </span>
950:         </td>
951:
952:         { /* Advice columns */ }
953:         {renderAdviceCell("LOW")}
954:         {renderAdviceCell("NORMAL")}
955:         {renderAdviceCell("HIGH")}
956:     </tr>
957:   );
958: })
959: })
960: </tbody>
961: </table>
962: </div>
963: </div>
964:
965: <Separator className="my-4" />
966:
967: <div className="w-full max-w-4xl">
968:   <div className="relative w-full" style={{ paddingBottom: "141.4%" }}>
969:     
974:   </div>
975: </div>
976:
977: <Separator className="my-4" />
978:
979: { /* Nutrition Section */ }
980: <div className="text-center mb-3">
981:   <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
982:     NUTRITION
983:   </h2>
984: </div>
985: <SectionTitle title="Nutrition Analysis" icon="" />
986: {reportData.nutritionData?.quote && (
987:   <div className="mb-3">
988:     <p className="text-sm font-semibold italic text-center text-gray-700">
989:       {reportData.nutritionData.quote}
990:     </p>
991:   </div>
992: )}
993: {reportData.nutritionData?.description && (
994:   <div className="mb-4">
995:     <p className="text-gray-600 text-justify leading-relaxed p-3 text-xs">
996:       {reportData.nutritionData.description}
997:     </p>
998:   </div>
999: )}
1000:
1001: <div className="w-full print:w-[700px] mx-auto mb-6 print:mb-4">
1002:   <div className="overflow-x-auto overflow-y-hidden">
1003:     <table className="w-full min-w-[700px] text-[13px] border border-gray-300 print:text-[12px] ta
1004:       <thead className="bg-gray-200 text-gray-700 uppercase text-[11px]">
1005:         <tr>
1006:           <th className="p-2 border border-gray-300 w-[18%] text-center align-middle">
1007:             Category
1008:           </th>
1009:           <th className="p-2 border border-gray-300 w-[21%] text-center align-middle">
1010:             Nutrient
1011:           </th>
1012:           <th className="p-2 border border-gray-300 w-[10%] text-center align-middle">
1013:             Score
1014:           </th>
1015:           <th className="p-2 border border-gray-300 w-[18%] text-center align-middle">
1016:             Health Impact

```



```

1017:         </th>
1018:         <th className="p-2 border border-gray-300 w-[18%] text-center align-middle">
1019:             Intake Level
1020:         </th>
1021:         <th className="p-2 border border-gray-300 w-[15%] text-center align-middle">
1022:             Source
1023:         </th>
1024:     </tr>
1025: </thead>
1026: <tbody>
1027:     {Object.entries(reportData.nutritionData?.data || {}).flatMap(
1028:       ([sectionName, sectionData]) => {
1029:         const entries = Object.entries(sectionData);
1030:
1031:         const formattedCategory = sectionName
1032:           .replace(/([A-Z])/g, " $1")
1033:           .replace(/^./, (str) => str.toUpperCase());
1034:
1035:         // Helper to format nutrient names
1036:         const formatNutrientName = (name: string) => {
1037:           let label = name
1038:             .replace(/([A-Z])/g, " $1") // space before capitals
1039:             .replace(/^./, (str) => str.toUpperCase()); // uppercase first letter
1040:
1041:           // Special handling for Vitamin patterns
1042:           label = label.replace(
1043:             /^Vitamin\s?([a-z])(\d+)?$/i,
1044:             (_, letter, number) =>
1045:               `Vitamin ${letter.toUpperCase()}${number || ""}`
1046:           );
1047:
1048:           return label;
1049:         };
1050:
1051:         return entries.map(([key, data], index) => {
1052:           const formattedLabel = formatNutrientName(key);
1053:
1054:           const score = Number(data.score ?? 0);
1055:           const scoreColor =
1056:             score <= 3
1057:               ? "bg-green-600"
1058:               : score <= 6
1059:               ? "bg-yellow-500"
1060:               : "bg-orange-500";
1061:
1062:           return (
1063:             <tr
1064:               key={`_${sectionName}-${key}`}
1065:               className="border-t border-gray-200 break-inside-avoid"
1066:             >
1067:               {index === 0 && (
1068:                 <td
1069:                   rowspan={entries.length}
1070:                   className="p-2 border border-gray-300 font-bold text-blue-800 uppercase bg-gray-100"
1071:                 >
1072:                   {formattedCategory}
1073:                 </td>
1074:               )}
1075:               <td className="p-2 border border-gray-300 font-medium text-gray-800 text-center" >
1076:                 {formattedLabel}
1077:               </td>
1078:               <td className="p-2 border border-gray-300 text-center align-middle">
1079:                 <span
1080:                   className={`inline-block px-5 py-1 rounded text-white font-bold ${scoreColor}`}
1081:                 >
1082:                   {score}
1083:                 </span>
1084:               </td>
1085:               <td className="p-2 border border-gray-300 text-center text-gray-700 break-words" >
1086:                 {data.healthImpact || "-"}
1087:               </td>
1088:               <td className="p-2 border border-gray-300 text-center text-gray-700 break-words" >
1089:                 {data.intakeLevel || "-"}

```

```

1090:         </td>
1091:         <td className="p-2 border border-gray-300 text-center text-gray-700 break-words"
1092:             {data.source || "-"}
1093:         </td>
1094:     </tr>
1095:   </tbody>
1096: </table>
1097: </div>
1098: </div>
1099: <Separator className="my-4" />
1100:
1101: <div className="w-full max-w-4xl">
1102:   <div className="relative w-full" style={{ paddingBottom: "141.4%" }}>
1103:     
1108:   </div>
1109: </div>
1110:
1111: <Separator className="my-4" />
1112:
1113: <div className="text-center mb-3">
1114:   <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
1115:     SPORTS AND FITNESS
1116:   </h2>
1117: </div>
1118:
1119: <div className="space-y-4 mb-4">
1120:   { /* Quote */
1121:   {sportsAndFitness.quote && (
1122:     <p className="text-sm font-semibold italic text-center text-red-600 mb-3">
1123:       {sportsAndFitness.quote}
1124:     </p>
1125:   )}
1126:
1127:   { /* Description */
1128:   {sportsAndFitness.description && (
1129:     <p className="text-gray-600 text-justify leading-relaxed mb-4 p-3 text-xs">
1130:       {sportsAndFitness.description}
1131:     </p>
1132:   )}
1133:
1134:   { /* Table */
1135:   <div className="w-full overflow-x-auto mb-4">
1136:     <table className="w-full min-w-[500px] text-xs border-collapse border border-gray-300">
1137:       <thead>
1138:         <tr className="bg-gray-100">
1139:           <th className="w-1/5 border border-gray-300 p-2 text-xs font-bold text-center uppercase">
1140:             CATEGORY
1141:           </th>
1142:           <th className="w-1/4 border border-gray-300 p-2 text-xs font-bold text-center uppercase">
1143:             TRAIT
1144:           </th>
1145:           <th className="w-1/6 border border-gray-300 p-2 text-xs font-bold text-center uppercase">
1146:             IMAGE
1147:           </th>
1148:           <th className="w-1/2 border border-gray-300 p-2 text-xs font-bold text-center uppercase">
1149:             DESCRIPTION
1150:           </th>
1151:         </tr>
1152:       </thead>
1153:       <tbody>
1154:         {Object.entries(sportsAndFitness)
1155:           .filter(
1156:             ([key]) =>
1157:               key !== "quote" &&
1158:               key !== "description" &&

```

```

1163:         key != "customImages"
1164:     )
1165:     .map(([categoryKey, fields]) => {
1166:         if (
1167:             !fields ||
1168:             typeof fields !== "object" ||
1169:             Array.isArray(fields)
1170:         ) {
1171:             return null;
1172:         }
1173:
1174:         const fieldEntries = Object.entries(
1175:             fields as Record<
1176:                 string,
1177:                 { level: string; description: string; label?: string }
1178:             >
1179:         );
1180:
1181:         return fieldEntries.map(([fieldKey, fieldData], index) => {
1182:             const selectedImageKey = fieldData.level
1183:                 ?.split("-")
1184:                 .pop()
1185:                 ?.toLowerCase();
1186:
1187:             const imageUrl =
1188:                 (
1189:                     sportsAndFitness.customImages as
1190:                         | Record<string, string>
1191:                         | undefined
1192:                 )?.[selectedImageKey ?? ""] ||
1193:                 defaultImageMap[selectedImageKey ?? ""] ||
1194:                 "/sports/default.png";
1195:
1196:             return (
1197:                 <tr
1198:                     key={`_${categoryKey}-${fieldKey}`}
1199:                     className="border-b border-gray-300 align-middle"
1200:                 >
1201:                     {index === 0 && (
1202:                         <td
1203:                             rowspan={fieldEntries.length}
1204:                             className="text-center font-semibold text-red-700 w-1/5 uppercase align-midd
1205:                         >
1206:                             {categoryKey
1207:                                 .replace(/_/g, " ") // <-- replace underscores with spaces
1208:                                 .replace(/[A-Z]/g, " $1")}
1209:                             </td>
1210:                         )}
1211:                     <td className="text-center font-medium uppercase text-gray-800 w-1/4 border bord
1212:                         {(fieldData.label || fieldKey).replace(/_/g, " ")}{" "}
1213:                         {/* <-- replace underscores */}
1214:                     </td>
1215:                     <td className="text-center w-1/5 border border-gray-300 p-1 align-middle">
1216:                         <img
1217:                             src={imageUrl || "/placeholder.svg"}
1218:                             alt={selectedImageKey}
1219:                             className="w-16 h-17 p-1 mx-auto rounded-lg"
1220:                         />
1221:                     </td>
1222:                     <td className="text-center text-gray-600 text-xs w-1/2 border border-gray-300 p-
1223:                         {fieldData.description}
1224:                     </td>
1225:                 </tr>
1226:             );
1227:         });
1228:     });
1229: </tbody>
1230: </table>
1231: </div>
1232: </div>
1233:
1234: <Separator className="my-4" />
1235:

```

```

1236: <div className="w-full max-w-4xl">
1237:   <div className="relative w-full" style={{ paddingBottom: "141.4%" }}>
1238:     
1243:   </div>
1244: </div>
1245:
1246: <Separator className="my-4" />
1247:
1248: { /* Lifestyle Conditions */ }
1249: <div className="text-center mb-3">
1250:   <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
1251:     LIFESTYLE CONDITIONS
1252:   </h2>
1253: </div>
1254:
1255: {reportData.lifestyleConditions && (
1256:   <div className="mb-6">
1257:     {reportData.lifestyleConditions.quote && (
1258:       <p className="text-sm font-semibold italic text-center text-red-600 mb-3">
1259:         "{reportData.lifestyleConditions.quote}"
1260:       </p>
1261:     )}
1262:
1263:     {reportData.lifestyleConditions.description && (
1264:       <p className="text-gray-600 text-justify leading-relaxed mb-4 p-3 text-xs">
1265:         {reportData.lifestyleConditions.description}
1266:       </p>
1267:     )}
1268:
1269:     <div className="w-full overflow-x-auto mb-6">
1270:       <table className="min-w-full border border-gray-300 bg-white rounded-lg overflow-hidden text
1271:         <thead className="bg-gray-100 text-left text-xs font-semibold text-gray-700">
1272:           <tr>
1273:             <th className="px-2 py-2 w-64 text-center align-middle">
1274:               Category
1275:             </th>
1276:             <th className="px-2 py-2 text-center align-middle">
1277:               Condition
1278:             </th>
1279:             <th className="px-2 py-2 text-center align-middle">
1280:               Your Strengths
1281:             </th>
1282:             <th className="px-2 py-2 text-center align-middle">
1283:               Improvement Needed
1284:             </th>
1285:           </tr>
1286:         </thead>
1287:         <tbody>
1288:           {Object.entries(reportData.lifestyleConditions)
1289:             .reverse() // ? reverse order
1290:             .map(([categoryId, categoryData]) => {
1291:               if (
1292:                 categoryId === "quote" ||
1293:                 categoryId === "description" ||
1294:                 typeof categoryData !== "object"
1295:               )
1296:                 return null;
1297:
1298:               const fields = Object.entries(
1299:                 categoryData as Record<string, HealthConditionStatus>
1300:               ).reverse(); // ? reverse conditions inside category
1301:
1302:               return fields.map(([condition, { status }], index) => (
1303:                 <tr>
1304:                   key={`_${categoryId}-${condition}_`}
1305:                   className="border-t text-center align-middle "
1306:                 >
1307:                   {index === 0 && (
1308:                     <td>

```

```

1309:         rowspan={fields.length}
1310:         className="px-2 py-5 font-medium text-center align-middle text-gray-900 "
1311:     >
1312:         <div className="flex items-center gap-2">
1313:             {reportData.lifestyleCategoryImages?.[
1314:                 categoryId
1315:             ] ? (
1316:                 <img
1317:                     src={
1318:                         reportData.lifestyleCategoryImages[
1319:                             categoryId
1320:                         ] || "/placeholder.svg"
1321:                     }
1322:                     alt={categoryId}
1323:                     className="w-12 h-12 rounded object-cover"
1324:                 />
1325:             ) : (
1326:                 <div className="w-4 h-4 bg-gray-300 rounded " />
1327:             )}
1328:             <span className="capitalize">
1329:                 {categoryId === categoryId.toUpperCase()
1330:                 ? categoryId
1331:                 : categoryId.replace(/([A-Z])/g, " $1")}
1332:             </span>
1333:         </div>
1334:     </td>
1335: }}
1336: <td className="px-3 py-3 ">{condition}</td>
1337: <td className="px-2 py-1 text-center">
1338:     {status === "strength" && (
1339:         <Check className="text-green-600 inline w-3 h-3" />
1340:     )}
1341: </td>
1342: <td className="px-2 py-1 text-center">
1343:     {status === "improvement" && (
1344:         <Check className="text-red-600 inline w-3 h-3" />
1345:     )}
1346: </td>
1347: </tr>
1348:     ));
1349:     }}}
1350: </tbody>
1351: </table>
1352: </div>
1353:
1354: <div className="text-center mb-7">
1355:     <h2 className="mb- bg-gray-300 text-white py-1 px-1 rounded"></h2>
1356: </div>
1357:
1358: {Object.entries(reportData.lifestyleConditions)
1359:   .filter(([key]) => !["quote", "description"].includes(key))
1360:   .map(([categoryId, conditions]) => (
1361:     <div key={categoryId} className="mb-6">
1362:       <div className="text-center">
1363:         <div className="text-base font-bold mb-3 bg-red-400 text-white py-0.5 px-2 rounded inline-block">
1364:           {categoryId
1365:             .replace(/([a-z])([A-Z])/g, "$1 $2")
1366:             .replace(/([A-Z])([A-Z][a-z])/g, "$1 $2")
1367:             .replace(/&/g, " & ")
1368:             .replace(/\\s+/g, " ")
1369:             .trim()}
1370:         </div>
1371:       </div>
1372:
1373:       <div className="grid grid-cols-1 gap-3">
1374:         {Object.entries(
1375:           conditions as Record<string, HealthConditionStatus>
1376:         ).map(([fieldKey, data]) => (
1377:           <div
1378:             key={fieldKey}
1379:             className="border rounded-lg border-gray-400 shadow-md bg-white p-3 space-y-2"
1380:           >
1381:             <div className="flex justify-between gap-2 items-stretch">

```

```

1382:         <div className="w-[76%]">
1383:             <h3 className="text-sm font-bold text-red-400 mb-1 uppercase">
1384:                 {fieldKey
1385:                     .replace(/([a-z])([A-Z])/g, "$1 $2")
1386:                     .replace(/([A-Z])([A-Z][a-z])/g, "$1 $2")
1387:                     .replace(/&/g, " & ")
1388:                     .replace(/\\s+/g, " ")
1389:                     .trim()}
1390:             </h3>
1391:
1392:             {data.description && (
1393:                 <p className="text-xs text-gray-700 leading-relaxed text-justify">
1394:                     {data.description}
1395:                 </p>
1396:             )}
1397:         </div>
1398:         <div className="w-[24%] flex items-center p-3 justify-center">
1399:             {data.sensitivity && (
1400:                 <img
1401:                     src={`/sensitivity/${data.sensitivity}.png`}
1402:                     alt={data.sensitivity}
1403:                     className="w-auto h-auto object-contain max-w-full max-h-29 print:max-h-25"
1404:                 </>
1405:             )}
1406:         </div>
1407:     </div>
1408:     <hr className="border-t border-gray-200" />
1409:     <div className="grid grid-cols-4 gap-2">
1410:         {(
1411:             ["avoid", "follow", "consume", "monitor"] as const
1412:         ).map((key) =>
1413:             data[key]?.length && data[key]?.length > 0 ? (
1414:                 <div key={key}>
1415:                     <p className="text-xs font-bold uppercase mb-1 ml-2">
1416:                         {data[`${key}Label`] || key.toUpperCase()}
1417:                     </p>
1418:                     <ul className="list-disc list-outside pl-5 text-xs text-gray-700 space-y-1">
1419:                         {data[key]?.flatMap((item, idx) =>
1420:                             typeof item === "string"
1421:                                 ? item
1422:                                   .split("\\n")
1423:                                   .filter((line) => line.trim() !== "")
1424:                                   .map((line, subIdx) => (
1425:                                       <li key={`_${idx}-${subIdx}`}>
1426:                                           {line.trim()}
1427:                                       </li>
1428:                                   ))
1429:                                 : null
1430:                         )}
1431:                     </ul>
1432:                 </div>
1433:             ) : null
1434:         )}
1435:     </div>
1436: </div>
1437: )})}
1438: </div>
1439: </div>
1440: )})}
1441: </div>
1442: )}
1443:
1444: <Separator className="my-4" />
1445:
1446: <div className="w-full max-w-4xl">
1447:     <div className="relative w-full" style={{ paddingBottom: "141.4%" }}>
1448:         
1453:     </div>
1454: </div>

```

```

1455:
1456: <Separator className="my-4" />
1457:
1458: { /* Metabolic Core */}
1459: <div className="text-center my-3">
1460:   <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
1461:     METABOLIC CORE
1462:   </h2>
1463: </div>
1464:
1465: <div className="space-y-4">
1466:   {metabolicCore.quote && (
1467:     <p className="text-sm font-semibold italic text-center text-red-600 mb-3">
1468:       "{metabolicCore.quote}"
1469:     </p>
1470:   )}
1471:   {metabolicCore.description && (
1472:     <p className="text-gray-600 text-justify leading-relaxed mb-4 p-3 text-xs">
1473:       {metabolicCore.description}
1474:     </p>
1475:   )}
1476:
1477: <div className="overflow-x-auto">
1478:   <table className="w-full border border-blue-500 text-xs">
1479:     <thead className="bg-blue-500 text-white text-center uppercase text-xs">
1480:       <tr>
1481:         <th className="py-1 px-2 w-[20%]">Metabolic Area</th>
1482:         <th className="py-1 px-2 w-[20%]">Gene (Alleles)</th>
1483:         <th className="py-1 px-2 w-[15%]">Your Genotype</th>
1484:         <th className="py-1 px-2 w-[15%]">Genetic Impact</th>
1485:         <th className="py-1 px-2 w-[30%]">Advice</th>
1486:       </tr>
1487:     </thead>
1488:     <tbody className="text-center">
1489:       {Object.entries(metabolicCore)
1490:         .filter(([key]) => key !== "quote" && key !== "description")
1491:         .map(([areaKey, areaData]) => {
1492:           const genes = areaData.genes || [];
1493:           const totalRows = Math.max(genes.length, 1);
1494:           return genes.length > 0 ? (
1495:             genes.map((gene, index) => (
1496:               <tr>
1497:                 <th>{` ${areaKey}-${index} `}
1498:                 <th>{`border-t border-slate-200`}
1499:               >
1500:                 {index === 0 ? (
1501:                   <>
1502:                     <td>
1503:                       {rowSpan={totalRows}
1504:                         className="font-bold align-middle px-2 py-2.5 text-blue-700 uppercase text
1505:                       }
1506:                     {areaKey
1507:                       .replace(/[a-z]([A-Z])/g, "$1 $2")
1508:                       .replace(/[A-Z]([A-Z][a-z])/g, "$1 $2")
1509:                       .replace(/&/g, " & ")
1510:                       .replace(/\\s+/g, " ")
1511:                       .trim()}
1512:                     </td>
1513:                   </>
1514:                 ) : null}
1515:                 <td className="px-2 py-1">{gene.name}</td>
1516:                 <td className="px-2 py-1 font-mono">
1517:                   {gene.genotype}
1518:                 </td>
1519:                 <td className="px-2 py-1">
1520:                   <div className="flex justify-center items-center gap-2">
1521:                     <span>
1522:                       className={`w-7 h-7 rounded-full inline-block ${
1523:                         gene.impact?.toLowerCase().includes("high")
1524:                           ? "bg-red-500"
1525:                           : gene.impact?.toLowerCase().includes("low")
1526:                           ? "bg-slate-400"
1527:                           : gene.impact

```

```

1528:         ?.toLowerCase()
1529:         .includes("moderate") ||
1530:         gene.impact
1531:         ?.toLowerCase()
1532:         .includes("average")
1533:         ? "bg-yellow-400"
1534:         : "bg-green-500"
1535:     }` }
1536:     ></span>
1537: </div>
1538: </td>
1539: {index === 0 ? (
1540:     <td
1541:         rowspan={totalRows}
1542:         className="px-2 py-1 text-left align-middle border-1 border-slate-200"
1543:     >
1544:         <div className=" text-center whitespace-pre-wrap text-xs">
1545:             {areaData.advice || "No advice provided"}
1546:         </div>
1547:     </td>
1548:     ) : null}
1549: </tr>
1550: ))
1551: ) : (
1552:     // Show row even if no genes, to display area advice
1553:     <tr key={areaKey} className="border-t border-slate-200">
1554:         <td className="font-bold px-2 py-2.5 text-blue-700 uppercase text-center border-r">
1555:             {areaKey
1556:                 .replace(/([a-z])([A-Z])/g, "$1 $2")
1557:                 .replace(/([A-Z])([A-Z][a-z])/g, "$1 $2")
1558:                 .replace(/&/g, " & ")
1559:                 .replace(/\\s+/g, " ")
1560:                 .trim()}
1561:         </td>
1562:         <td className="px-2 py-1 text-gray-400">No genes</td>
1563:         <td className="px-2 py-1 text-gray-400">-</td>
1564:         <td className="px-2 py-1 text-gray-400">-</td>
1565:         <td className="px-2 py-1 text-left">
1566:             <div className="whitespace-pre-wrap text-xs">
1567:                 {areaData.advice || "No advice provided"}
1568:             </div>
1569:         </td>
1570:     </tr>
1571: );
1572: }}}
1573: </tbody>
1574: </table>
1575: </div>
1576: </div>
1577:
1578: <Separator className="my-4" />
1579:
1580: <div className="w-full max-w-4xl">
1581:     <div className="relative w-full" style={{ paddingBottom: "141.4%" }}>
1582:         
1587:     </div>
1588: </div>
1589:
1590: <Separator className="my-4" />
1591:
1592: {/* Digestive Health */}
1593: <div className="text-center mb-3">
1594:     <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
1595:         DIGESTIVE HEALTH
1596:     </h2>
1597: </div>
1598:
1599: {digestiveHealth?.quote && (
1600:     <p className="text-sm font-semibold italic text-center text-red-600 mb-3">

```



```

1601:         "{digestiveHealth.quote}"
1602:     </p>
1603: )}
1604: {digestiveHealth?.description && (
1605:     <p className="text-gray-600 text-justify leading-relaxed mb-4 p-3 text-xs">
1606:         {digestiveHealth.description}
1607:     </p>
1608: )}
1609: <div className="overflow-x-auto rounded-sm shadow">
1610:     {digestiveHealth?.data &&
1611:         Object.keys(digestiveHealth.data).length > 0 ? (
1612:             <table className="min-w-full text-xs text-center border border-gray-300 table-fixed">
1613:                 <thead className="bg-red-500 text-white">
1614:                     <tr>
1615:                         <th className="w-[20%] p-2 border border-gray-300">Title</th>
1616:                         <th className="w-[20%] p-2 border border-gray-300">Icon</th>
1617:                         <th className="w-[30%] p-2 text-center border border-gray-300">
1618:                             LOW
1619:                         </th>
1620:                         <th className="w-[30%] p-2 text-center border border-gray-300">
1621:                             HIGH
1622:                         </th>
1623:                     </tr>
1624:                 </thead>
1625:                 <tbody>
1626:                     {Object.entries(digestiveHealth.data).map(([key, item]) => {
1627:                         const iconPath = item.icon
1628:                             ? `/digestive/${item.icon}.png`
1629:                             : null;
1630:                         const isLow = item.sensitivity === "Low";
1631:                         const isHigh = item.sensitivity === "High";
1632:
1633:                         return (
1634:                             <tr key={key} className="border-t border-gray-200">
1635:                                 <td className="p-2 border border-gray-300 font-medium text-center uppercase">
1636:                                     {item.title || <span className="italic">Untitled</span>}
1637:                                 </td>
1638:                                 <td className="p-2 border border-gray-300">
1639:                                     {iconPath ? (
1640:                                         <img
1641:                                             src={iconPath}
1642:                                             alt={item.title || "icon"}
1643:                                             className="w-15 h-auto mx-auto object-contain"
1644:                                             onError={(e) => {
1645:                                                 e.currentTarget.onerror = null;
1646:                                                 e.currentTarget.src =
1647:                                                     "/digestive/placeholder.png";
1648:                                             }}
1649:                                         />
1650:                                     ) : (
1651:                                         <span className="text-slate-400 italic">No Icon</span>
1652:                                     )}
1653:                                 </td>
1654:                                 <td
1655:                                     className={`p-2 border border-gray-300 text-center whitespace-pre-line ${
1656:                                         isLow ? "text-gray-900" : "text-gray-400"
1657:                                     }`}
1658:                                 >
1659:                                     {item.good || <span className="italic">?</span>}
1660:                                 </td>
1661:                                 <td
1662:                                     className={`p-2 border border-gray-300 text-center whitespace-pre-line ${
1663:                                         isHigh ? "text-gray-900" : "text-gray-400"
1664:                                     }`}
1665:                                 >
1666:                                     {item.bad || <span className="italic">?</span>}
1667:                                 </td>
1668:                             </tr>
1669:                         );
1670:                     })}
1671:                 </tbody>
1672:             </table>
1673:         ) : (

```

```

1674:         <p className="text-center text-sm text-gray-500 p-4 italic">
1675:             No digestive health data available.
1676:         </p>
1677:     )}
1678: </div>
1679:
1680: <Separator className="my-4" />
1681:
1682: <div className="w-full max-w-4xl">
1683:   <div className="relative w-full" style={{ paddingBottom: "141.4%" }}>
1684:     
1689:   </div>
1690: </div>
1691:
1692: <Separator className="my-" />
1693:
1694: { /* Genes & Addiction */ }
1695: <div className="text-center mb-3">
1696:   <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded uppercase">
1697:     Genes & Addiction
1698:   </h2>
1699: </div>
1700: <div className="mb-4">
1701:   {genesAndAddition.quote && (
1702:     <p className="text-sm font-semibold italic text-center text-red-600 mb-3">
1703:       "{genesAndAddition.quote}"
1704:     </p>
1705:   )}
1706:
1707:   {genesAndAddition.description && (
1708:     <p className="text-gray-600 text-justify leading-relaxed mb-4 p-3 text-xs">
1709:       {genesAndAddition.description}
1710:     </p>
1711:   )}
1712:
1713: <div className="w-full">
1714:   <table className="w-full table-fixed border border-gray-300 bg-white border-collapse text-xs">
1715:     <thead className="bg-blue-300">
1716:       <tr>
1717:         <th className="p-2 text-xs font-bold text-white text-left border border-gray-300 upperca
1718:           Addictive Habit
1719:         </th>
1720:         <th className="p-2 text-xs font-bold text-white text-center border border-gray-300 upper
1721:           Chance of Addiction
1722:         </th>
1723:       </tr>
1724:     </thead>
1725:     <tbody>
1726:       {Object.entries(genesAndAddition.data || {}).map(
1727:         ([key, entry], index) => (
1728:           <tr
1729:             key={key}
1730:             className={`transition-all duration-200 text-center ${{
1731:               index % 2 === 0 ? "bg-white" : "bg-gray-50"
1732:             }}`}
1733:           >
1734:             <td className="p-0 border border-gray-300">
1735:               <table className="w-full h-full table-fixed border-collapse">
1736:                 <tbody>
1737:                   <tr>
1738:                     <td className="w-1/3 p-2 align-middle bg-blue-100">
1739:                       <span className="text-xs font-semibold text-gray-800 uppercase">
1740:                         {entry.title}
1741:                       </span>
1742:                     </td>
1743:                     <td className="w-[2px] border-gray-300"></td>
1744:                     <td className="w-2/3 p-2 text-center">
1745:                       {entry.icon && (
1746:                         <img

```

```

1747:             src={` /addiction/icons/${entry.icon}.png`}
1748:             alt={entry.title}
1749:             className="w-20 h-24 object-contain mx-auto"
1750:         />
1751:     )}
1752: </td>
1753: </tr>
1754: </tbody>
1755: </table>
1756: </td>
1757: <td
1758:     className="text-center border border-gray-300 align-middle"
1759:     style={{ width: "200px" }} // lock column width
1760: >
1761:     {entry.sensitivityIcon && (
1762:         <div className="flex justify-center items-center h-full">
1763:             <img
1764:                 src={` /addiction/sensitivity/${entry.sensitivityIcon}.png`}
1765:                 alt="Chance of Addiction"
1766:                 className="max-w-[170px] max-h-[100px] object-contain"
1767:             />
1768:         </div>
1769:     )}
1770: </td>
1771: </tr>
1772: )
1773: )}
1774: </tbody>
1775: </table>
1776: </div>
1777: </div>
1778:
1779: <Separator className="mt-7" />
1780:
1781: <div className="w-full max-w-4xl">
1782:     <div className="relative w-full" style={{ paddingBottom: "141.4%" }}>
1783:         
1788:     </div>
1789: </div>
1790:
1791: <Separator className="my-4" />
1792: { /* Sleep & Rest */ }
1793:
1794: <div className="text-center mb-3">
1795:     <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded uppercase">
1796:         Sleep & Rest
1797:     </h2>
1798: </div>
1799: <SectionTitle title="Sleep & Rest" icon="" />
1800:
1801: {sleepAndRest?.quote && (
1802:     <p className="text-sm font-semibold italic text-center text-red-600 mb-3">
1803:         "{sleepAndRest.quote}"
1804:     </p>
1805: )}
1806:
1807: {sleepAndRest?.description && (
1808:     <p className="text-gray-600 text-justify leading-relaxed mb-4 p-3 text-xs">
1809:         {sleepAndRest.description}
1810:     </p>
1811: )}
1812:
1813: <div className="overflow-x-auto rounded-sm shadow mb-4">
1814:     <table className="min-w-full text-xs text-center border border-gray-300 table-fixed">
1815:         <thead className="bg-blue-600 text-white">
1816:             <tr>
1817:                 <th className="w-[25%] p-1 border border-gray-300">
1818:                     Sleep Parameter
1819:                 </th>

```

```

1820:         <th className="w-[25%] p-1 border border-gray-300">
1821:             Genetic Icon
1822:         </th>
1823:         <th className="w-[50%] p-1 border border-gray-300">
1824:             Intervention Advice
1825:         </th>
1826:     </tr>
1827: </thead>
1828: <tbody>
1829:     {sleepAndRest?.data &&
1830:       Object.entries(sleepAndRest.data).map(([key, entry], index) => (
1831:         <tr
1832:           key={key}
1833:           className={index % 2 === 0 ? "bg-white" : "bg-gray-50"}
1834:         >
1835:           {/* Sleep Parameter */}
1836:           <td className="p-1 border border-gray-300 bg-blue-200 font-medium text-center capitali
1837:             {entry.title?.trim()
1838:               ? entry.title
1839:               : key?.replace(/_/g, " ") || "No parameter"}
1840:           </td>
1841:
1842:           {/* Genetic Icon */}
1843:           <td className="p-1 border border-gray-300">
1844:             {entry.image ? (
1845:               <img
1846:                 src={`\sleep/${entry.image}.png`}
1847:                 alt={entry.title || key}
1848:                 className="w-24 h-26 object-contain mx-auto"
1849:                 onError={(e) => {
1850:                   e.currentTarget.onerror = null;
1851:                   e.currentTarget.src = "/sleep/placeholder.png";
1852:                 }}
1853:               />
1854:             ) : (
1855:               <span className="text-slate-400 italic">No Image</span>
1856:             )}
1857:           </td>
1858:
1859:           {/* Intervention Advice */}
1860:           <td className="p-1 border border-gray-300 text-left whitespace-pre-line text-gray-700"
1861:             {entry.intervention?.trim() ? (
1862:               entry.intervention
1863:             ) : (
1864:               <span className="italic text-gray-400">
1865:                 No advice available
1866:               </span>
1867:             )}
1868:           </td>
1869:         </tr>
1870:       )})
1871:   </tbody>
1872: </table>
1873: </div>
1874:
1875: <Separator className="my-4" />
1876:
1877: <div className="w-full max-w-4xl">
1878:   <div className="relative w-full" style={{ paddingBottom: "141.4%" }}>
1879:     
1884:   </div>
1885: </div>
1886:
1887: <Separator className="my-4" />
1888:
1889: {/* Allergies & Sensitivity */}
1890: <div className="text-center mb-3">
1891:   <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded uppercase">
1892:     Allergies & Sensitivity

```

```

1893:     </h2>
1894:   </div>
1895:
1896:   {typeof allergiesAndSensitivity === "object" &&
1897:     ("quote" in allergiesAndSensitivity ||
1898:       "description" in allergiesAndSensitivity) &&
1899:     ((allergiesAndSensitivity as any).quote ||
1900:       (allergiesAndSensitivity as any).description) && (
1901:       <div className="mb-3 space-y-2">
1902:         {"quote" in allergiesAndSensitivity &&
1903:           (allergiesAndSensitivity as any).quote && (
1904:             <div className="text-sm font-semibold italic text-center text-red-600 mb-3">
1905:               "{(allergiesAndSensitivity as any).quote}"
1906:             </div>
1907:           )}
1908:         {"description" in allergiesAndSensitivity &&
1909:           (allergiesAndSensitivity as any).description && (
1910:             <p className="text-gray-600 text-justify leading-relaxed mb-4 p-3 text-xs">
1911:               {(allergiesAndSensitivity as any).description}
1912:             </p>
1913:           )}
1914:       </div>
1915:     )}
1916:
1917:   <div className="overflow-x-auto shadow mb-4">
1918:     <table className="min-w-full text-xs text-center border border-gray-300 table-fixed">
1919:       <thead className="bg-orange-600 text-white">
1920:         <tr>
1921:           <th className="w-[35%] p-2 border border-gray-300">
1922:             ALLERGIES
1923:           </th>
1924:           <th className="w-[65%] p-2 border border-gray-300">
1925:             TENDENCY OF ALLERGY
1926:           </th>
1927:         </tr>
1928:       </thead>
1929:       <tbody>
1930:         {allergiesAndSensitivity.data &&
1931:           Object.entries(allergiesAndSensitivity.data).map(
1932:             ([key, entry], index) => (
1933:               <tr>
1934:                 <td>
1935:                   key={key}
1936:                   className={index % 2 === 0 ? "bg-white" : "bg-gray-50"}
1937:                 >
1938:                   <td className="p-2 border border-gray-300 font-medium text-center capitalize">
1939:                     {entry.title || `Allergy ${index + 1}`}
1940:                   </td>
1941:                   <td>
1942:                     className="p-2 border border-gray-300"
1943:                     style={{ width: "100px" }}
1944:                   >
1945:                     {entry.image ? (
1946:                       <img
1947:                         src={`../allergies/${entry.image}.png`}
1948:                         alt={entry.title}
1949:                         className="max-w-[170px] max-h-[130px] object-contain mx-auto"
1950:                         onError={(e) => {
1951:                           e.currentTarget.onerror = null;
1952:                           e.currentTarget.src =
1953:                             "/allergies/placeholder.png";
1954:                         }}
1955:                       />
1956:                     ) : (
1957:                       <span className="text-slate-400 italic">
1958:                         No Image
1959:                       </span>
1960:                     )}
1961:                   </td>
1962:                 </tr>
1963:               )
1964:             )}
1965:       </tbody>
1966:     </table>

```

```

1966:     </div>
1967:
1968:     { /* General Advice */ }
1969:     { typeof allergiesAndSensitivity === "object" &&
1970:       "generalAdvice" in allergiesAndSensitivity &&
1971:       typeof allergiesAndSensitivity.generalAdvice === "string" &&
1972:       allergiesAndSensitivity.generalAdvice.trim() && (
1973:         <div className="rounded shadow border border-gray-300 mt-4 overflow-hidden">
1974:           { /* <div className="bg-orange-600 text-white px-2 py-2 font-semibold text-xs text-left">
1975:             General Advice
1976:           </div> */ }
1977:           <div className="bg-white p-2 text-xs text-justify whitespace-pre-line break-words leading-re
1978:             { allergiesAndSensitivity.generalAdvice }
1979:           </div>
1980:         </div>
1981:       ) }
1982:
1983:     <Separator className="my-4" />
1984:
1985:     <div className="w-full max-w-4xl">
1986:       <div className="relative w-full" style={{ paddingBottom: "141.4%" }}>
1987:         
1992:       </div>
1993:     </div>
1994:
1995:     <Separator className="my-4" />
1996:     { /* Preventive Health */ }
1997:     <div className="text-center mb-3">
1998:       <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
1999:         PREVENTIVE HEALTH
2000:       </h2>
2001:     </div>
2002:
2003:     <div className="grid grid-cols-2 gap-6">
2004:       { /* Diagnostic Tests Table */ }
2005:       <table className="min-w-full border border-gray-300 text-sm text-left">
2006:         <thead>
2007:           <tr className="bg-gray-400 text-white uppercase">
2008:             <th
2009:               colspan={3}
2010:               className="px-3 py-2 border text-black border-gray-400 text-center"
2011:             >
2012:               Diagnostic Tests
2013:             </th>
2014:           </tr>
2015:           <tr className="bg-gray-300 text-gray-900 uppercase">
2016:             <th className="px-3 py-2 border border-gray-400 text-center">
2017:               Tests
2018:             </th>
2019:             <th className="px-3 py-2 border border-gray-400 text-center">
2020:               Half Yearly
2021:             </th>
2022:             <th className="px-3 py-2 border border-gray-400 text-center">
2023:               Yearly
2024:             </th>
2025:           </tr>
2026:         </thead>
2027:         <tbody>
2028:           { [
2029:             ...preventiveHealth.diagnosticTests.halfYearly.map((test) => ({
2030:               test,
2031:               half: "YES",
2032:               year: "",
2033:             })),
2034:             ...preventiveHealth.diagnosticTests.yearly.map((test) => ({
2035:               test,
2036:               half: "",
2037:               year: "YES",
2038:             })),

```

```

2039:         ].map((row, index) => (
2040:             <tr
2041:                 key={index}
2042:                 className={index % 2 === 0 ? "bg-gray-100" : "bg-white"}
2043:             >
2044:                 <td className="px-3 py-2 border border-gray-300">
2045:                     {row.test}
2046:                 </td>
2047:                 <td className="px-3 py-2 border border-gray-300 text-center">
2048:                     {row.half}
2049:                 </td>
2050:                 <td className="px-3 py-2 border border-gray-300 text-center">
2051:                     {row.year}
2052:                 </td>
2053:             </tr>
2054:         )})
2055:     </tbody>
2056: </table>
2057:
2058:     { /* Nutritional Supplements Table */ }
2059:     <table className="min-w-full border border-gray-300 text-sm text-left">
2060:         <thead>
2061:             <tr className="bg-blue-500 text-white uppercase">
2062:                 <th
2063:                     colSpan={2}
2064:                     className="px-3 py-2 border border-blue-500 text-center"
2065:                 >
2066:                     Nutritional Supplements
2067:                 </th>
2068:             </tr>
2069:             <tr className="bg-blue-300 text-blue-900 uppercase">
2070:                 <th className="px-3 py-2 border border-blue-300 text-center">
2071:                     Supplements
2072:                 </th>
2073:                 <th className="px-3 py-2 border border-blue-300 text-center">
2074:                     Yes/No
2075:                 </th>
2076:             </tr>
2077:         </thead>
2078:         <tbody>
2079:             {preventiveHealth.nutritionalSupplements.map(
2080:                 (supplement, index) => (
2081:                     <tr
2082:                         key={index}
2083:                         className={index % 2 === 0 ? "bg-blue-100" : "bg-white"}
2084:                     >
2085:                         <td className="px-3 py-2 border border-gray-300">
2086:                             {supplement.supplement}
2087:                         </td>
2088:                         <td className="px-3 py-2 border border-gray-300 text-center">
2089:                             {supplement.needed ? "YES" : ""}
2090:                         </td>
2091:                     </tr>
2092:                 )
2093:             )}
2094:         </tbody>
2095:     </table>
2096: </div>
2097:
2098:     { /* Description Section at the Bottom */ }
2099:
2100:     <p className="text-sm text-black whitespace-pre-line text-justify mt-6">
2101:         {preventiveHealth.description || "No description provided."}
2102:     </p>
2103:
2104:     <Separator className="my-4" />
2105:     { /* Family Genetic Impact */ }
2106:     <div className="text-center mb-3">
2107:         <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
2108:             FAMILY GENETIC IMPACT
2109:         </h2>
2110:     </div>
2111:     <div className="overflow-x-auto">

```

```

2112:     {familyGeneticImpactSection.description && (
2113:         <p className="text-gray-600 text-sm leading-relaxed mb-4 p-3 whitespace-pre-line">
2114:             {familyGeneticImpactSection.description}
2115:         </p>
2116:     )}
2117:
2118:     {familyGeneticImpactSection.impacts.length === 0 ? (
2119:         <p className="text-center text-gray-500 italic">
2120:             No family genetic impacts recorded.
2121:         </p>
2122:     ) : (
2123:         <table className="w-full border border-blue-500 text-xs">
2124:             <thead className="bg-blue-500 text-white text-center uppercase text-xs">
2125:                 <tr>
2126:                     <th className="py-2 px-2 w-[20%]">Genes</th>
2127:                     <th className="py-2 px-2 w-[30%]">Normal/Common Alleles</th>
2128:                     <th className="py-2 px-2 w-[15%]">Your Result</th>
2129:                     <th className="py-2 px-2 w-[35%]">Impact on Health</th>
2130:                 </tr>
2131:             </thead>
2132:             <tbody className="text-center">
2133:                 {familyGeneticImpactSection.impacts.map((impact, index) => (
2134:                     <tr key={index} className="border-t border-slate-200">
2135:                         <td className="px-2 py-2 font-bold text-blue-700 uppercase border-r border-slate-200">
2136:                             {impact.gene}
2137:                         </td>
2138:                         <td className="px-2 py-2">{impact.normalAlleles}</td>
2139:                         <td className="px-2 py-2 font-mono">{impact.yourResult}</td>
2140:                         <td className="px-2 py-2 text-center">
2141:                             {impact.healthImpact}
2142:                         </td>
2143:                     </tr>
2144:                 ))}
2145:             </tbody>
2146:         </table>
2147:     )}
2148:
2149:     <div className="flex justify-center items-center">
2150:         
2155:     </div>
2156: </div>
2157:
2158: <Separator className="my-4" />
2159: {/* ===== Genomic Analysis Section (Moved Up) ===== */}
2160: <div className="text-center mb-3">
2161:     <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
2162:         GENOMIC ANALYSIS
2163:     </h2>
2164: </div>
2165:
2166: {/* Show description if available */}
2167: {reportData.genomicAnalysisTable?.description && (
2168:     <p className="text-gray-600 text-justify leading-relaxed mb-4 p-3 text-xs">
2169:         {reportData.genomicAnalysisTable.description}
2170:     </p>
2171: )}
2172: <div className="mb-4 overflow-x-auto">
2173:     {reportData.genomicAnalysisTable?.categories?.length === 0 ? (
2174:         <p className="text-center text-gray-500 italic">
2175:             No genomic analysis data recorded.
2176:         </p>
2177:     ) : (
2178:         <table className="w-full border border-blue-500 text-xs text-center">
2179:             <thead className=" text-white bg-blue-400 uppercase text-xs">
2180:                 <tr>
2181:                     <th className="py-2 px-2 w-[20%] border border-blue-500">
2182:                         Category
2183:                     </th>
2184:                     <th className="py-2 px-2 w-[22%] border border-blue-500">

```



```

2185:         Area
2186:     </th>
2187:     <th className="py-2 px-2 w-[28%] border border-blue-500">
2188:         Traits
2189:     </th>
2190:     <th className="py-2 px-2 w-[30%] border border-blue-500">
2191:         Genes
2192:     </th>
2193: </tr>
2194: </thead>
2195: <tbody>
2196:     {reportData.genomicAnalysisTable?.categories?.map(
2197:       (cat, catIndex) => {
2198:         // Sort by area while keeping other data intact
2199:         const sortedSubs = [...(cat.subcategories || [])].sort(
2200:           (a, b) => (a.area || "").localeCompare(b.area || "")
2201:         );
2202:
2203:         const subLen = sortedSubs.length || 1;
2204:         return sortedSubs.length > 0 ? (
2205:           sortedSubs.map((sub, subIndex) => (
2206:             <tr>
2207:               key={`${catIndex}-${subIndex}`}
2208:               className="bg-white"
2209:             >
2210:               {subIndex === 0 && (
2211:                 <td>
2212:                   className="px-3 py-2 border border-gray-300 text-center font-bold align-middle"
2213:                   rowspan={subLen}
2214:                 >
2215:                   {cat.category}
2216:                 </td>
2217:               )}
2218:               <td className="px-3 py-2 border border-gray-300 text-center break-words text-xs">
2219:                 {sub.area || "-"}
2220:               </td>
2221:               <td className="px-3 py-2 border border-gray-300 text-center break-words text-xs">
2222:                 {sub.trait || "-"}
2223:               </td>
2224:               <td className="px-3 py-2 border border-gray-300 text-center break-words text-xs">
2225:                 {sub.genes?.join(", ") || "-"}
2226:               </td>
2227:             </tr>
2228:           )}
2229:         ) : (
2230:           <tr key={`${catIndex}-empty`} className="bg-white">
2231:             <td className="px-3 py-2 border border-gray-300 text-center font-bold align-middle">
2232:               {cat.category}
2233:             </td>
2234:             <td>
2235:               colSpan={3}
2236:               className="px-3 py-2 border border-gray-300 text-center italic text-gray-500 tex
2237:             >
2238:               No rows added
2239:             </td>
2240:           </tr>
2241:         );
2242:       }
2243:     )}
2244:   </tbody>
2245: </table>
2246: )}
2247: </div>
2248:
2249: {/* Now render the rest of the sections BELOW this */}
2250: <Separator className="my-4" />
2251: {/* Gene Test Results */}
2252: <SectionTitle title="Gene Test Results" icon="" />
2253: <div className="mb-4 overflow-x-auto">
2254:   {geneTestResults.length === 0 ? (
2255:     <p className="text-center text-gray-500 italic">
2256:       No gene test results recorded.
2257:     </p>

```

```

2258:     ) : (
2259:       <table className="min-w-full border border-gray-200 text-xs text-left">
2260:         <thead className="bg-blue-300 text-white">
2261:           <tr>
2262:             <th className="px-3 py-2 border text-center uppercase">
2263:               Gene Code
2264:             </th>
2265:             <th className="px-3 py-2 border text-center uppercase">
2266:               Gene Name
2267:             </th>
2268:             <th className="px-3 py-2 border text-center uppercase">
2269:               Variation
2270:             </th>
2271:             <th className="px-3 py-2 border text-center uppercase">
2272:               Result
2273:             </th>
2274:           </tr>
2275:         </thead>
2276:         <tbody>
2277:           {geneTestResults.map((result, index) => (
2278:             <tr key={index}>
2279:               <td className="px-3 py-2 border border-gray-300 font-mono whitespace-normal text-center">
2280:                 {result.genecode || "-"}
2281:               </td>
2282:               <td className="px-3 py-2 border border-gray-300 whitespace-normal text-center break-wor">
2283:                 {result.geneName || "-"}
2284:               </td>
2285:               <td className="px-3 py-2 border border-gray-300 font-mono whitespace-normal text-center">
2286:                 {result.variation || "-"}
2287:               </td>
2288:               <td className="px-3 py-2 border border-gray-300 font-mono font-bold whitespace-normal">
2289:                 {result.result || "-"}
2290:               </td>
2291:             </tr>
2292:           ))}
2293:         </tbody>
2294:       </table>
2295:     )}
2296:   </div>
2297:   <Separator className="my-4" />
2298:   { /* Enhanced Comprehensive Summary */ }
2299:   <div className="text-center mb-3">
2300:     <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
2301:       COMPREHENSIVE REPORT SUMMARY
2302:     </h2>
2303:   </div>
2304:   { /* Generate comprehensive summary based on all data */ }
2305:   { (() => {
2306:     // Analyze diet data in detail
2307:     const dietAnalysis = Object.entries(groupedDietFields).reduce(
2308:       (acc, [category, fields]) => {
2309:         const scores = fields.map((f) => f.score);
2310:         const levels = fields.map((f) => f.level);
2311:         const highScores = fields.filter((f) => f.score >= 7);
2312:         const mediumScores = fields.filter(
2313:           (f) => f.score >= 4 && f.score < 7
2314:         );
2315:         const lowScores = fields.filter((f) => f.score < 4);
2316:         const avgScore =
2317:           scores.reduce((sum, score) => sum + score, 0) / scores.length ||
2318:           0;
2319:
2320:         const recommendations = fields.reduce((recAcc, field) => {
2321:           if (field.recommendations) {
2322:             Object.entries(field.recommendations).forEach(
2323:               ([level, rec]) => {
2324:                 if (field.selectedLevel === level) {
2325:                   recAcc.push(`${field.label}: ${rec}`);
2326:                 }
2327:               }
2328:             );
2329:           } else if (field.recommendation) {
2330:             recAcc.push(`${field.label}: ${field.recommendation}`);

```

```

2331:         }
2332:         return recAcc;
2333:     }, [] as string[]);
2334:
2335:     acc[category] = {
2336:         highScores: highScores.length,
2337:         mediumScores: mediumScores.length,
2338:         lowScores: lowScores.length,
2339:         avgScore,
2340:         total: fields.length,
2341:         fields: fields,
2342:         recommendations: recommendations.slice(0, 3), // Top 3 recommendations
2343:     };
2344:     return acc;
2345: },
2346: {} as Record<
2347: string,
2348: {
2349:     highScores: number;
2350:     mediumScores: number;
2351:     lowScores: number;
2352:     avgScore: number;
2353:     total: number;
2354:     fields: any[];
2355:     recommendations: string[];
2356: }
2357: >
2358: );
2359:
2360: // Analyze nutrition data in detail
2361: const nutritionAnalysis = Object.entries(
2362:     reportData.nutritionData?.data || {}
2363: ).reduce(
2364:     (acc, [section, data]) => {
2365:         const entries = Object.entries(data);
2366:         const highImpact = entries.filter(
2367:             ([_, item]) =>
2368:                 item.healthImpact?.toLowerCase().includes("high") ||
2369:                 item.healthImpact?.toLowerCase().includes("increased")
2370:         );
2371:         const lowImpact = entries.filter(
2372:             ([_, item]) =>
2373:                 item.healthImpact?.toLowerCase().includes("low") ||
2374:                 item.healthImpact?.toLowerCase().includes("reduced")
2375:         );
2376:         const normalImpact = entries.filter(
2377:             ([_, item]) =>
2378:                 item.healthImpact?.toLowerCase().includes("normal") ||
2379:                 item.healthImpact?.toLowerCase().includes("average")
2380:         );
2381:         const avgScore =
2382:             entries.reduce((sum, [_, item]) => sum + (item.score || 0), 0) /
2383:             entries.length || 0;
2384:
2385:         const detailedItems = entries.map(([key, item]) => ({
2386:             name: key
2387:                 .replace(/[A-Z]/g, " $1")
2388:                 .replace(/^./, (str) => str.toUpperCase()),
2389:             score: item.score,
2390:             impact: item.healthImpact,
2391:             level: item.intakeLevel,
2392:             source: item.source,
2393:         }));
2394:
2395:         acc[section] = {
2396:             highImpact: highImpact.length,
2397:             lowImpact: lowImpact.length,
2398:             normalImpact: normalImpact.length,
2399:             avgScore,
2400:             total: entries.length,
2401:             items: detailedItems,
2402:         };
2403:         return acc;

```

```

2404:         },
2405:         {} as Record<
2406:             string,
2407:             {
2408:                 highImpact: number;
2409:                 lowImpact: number;
2410:                 normalImpact: number;
2411:                 avgScore: number;
2412:                 total: number;
2413:                 items: any[];
2414:             }
2415:         >
2416:     );
2417:
2418:     // Analyze lifestyle conditions in detail
2419:     const lifestyleAnalysis = Object.entries(
2420:         reportData.lifestyleConditions || {}
2421:     )
2422:     .filter(([key]) => ![ "quote", "description" ].includes(key))
2423:     .reduce(
2424:         (acc, [category, conditions]) => {
2425:             const entries = Object.entries(
2426:                 conditions as Record<string, HealthConditionStatus>
2427:             );
2428:             const strengths = entries.filter(
2429:                 ([_, data]) => data.status === "strength"
2430:             );
2431:             const improvements = entries.filter(
2432:                 ([_, data]) => data.status === "improvement"
2433:             );
2434:
2435:             const detailedConditions = entries.map(([condition, data]) => ({
2436:                 name: condition,
2437:                 status: data.status,
2438:                 description: data.description,
2439:                 sensitivity: data.sensitivity,
2440:                 recommendations: {
2441:                     avoid: data.avoid || [],
2442:                     follow: data.follow || [],
2443:                     consume: data.consume || [],
2444:                     monitor: data.monitor || [],
2445:                 },
2446:             }));
2447:
2448:             acc[category] = {
2449:                 strengths: strengths.length,
2450:                 improvements: improvements.length,
2451:                 total: entries.length,
2452:                 conditions: detailedConditions,
2453:                 hasImages: !!reportData.lifestyleCategoryImages?.[category],
2454:             };
2455:             return acc;
2456:         },
2457:         {} as Record<
2458:             string,
2459:             {
2460:                 strengths: number;
2461:                 improvements: number;
2462:                 total: number;
2463:                 conditions: any[];
2464:                 hasImages: boolean;
2465:             }
2466:         >
2467:     );
2468:
2469:     // Analyze metabolic core in detail
2470:     const metabolicAnalysis = Object.entries(metabolicCore)
2471:     .filter(([key]) => key !== "quote" && key !== "description")
2472:     .reduce(
2473:         (acc, [area, data]) => {
2474:             let genes: any[] = [];
2475:             if (
2476:                 typeof data === "object" &&

```

```

2477:         data !== null &&
2478:         "genes" in data &&
2479:         Array.isArray((data as any).genes)
2480:     ) {
2481:         genes = (data as any).genes;
2482:     }
2483:
2484:     // Filter genes by impact level
2485:     const highImpact = genes.filter((g) =>
2486:         g.impact?.toLowerCase().includes("high")
2487:     );
2488:
2489:     const lowImpact = genes.filter((g) =>
2490:         g.impact?.toLowerCase().includes("low")
2491:     );
2492:
2493:     const moderateImpact = genes.filter(
2494:         (g) =>
2495:             g.impact?.toLowerCase().includes("moderate") ||
2496:             g.impact?.toLowerCase().includes("average")
2497:     );
2498:
2499:     // Add normal impact filter
2500:     const normalImpact = genes.filter((g) =>
2501:         g.impact?.toLowerCase().includes("normal")
2502:     );
2503:
2504:     const detailedGenes = genes.map((gene) => ({
2505:         name: gene.name,
2506:         genotype: gene.genotype,
2507:         impact: gene.impact,
2508:         advice: gene.advice,
2509:     }));
2510:
2511:     acc[area] = {
2512:         highImpact: highImpact.length,
2513:         moderateImpact: moderateImpact.length,
2514:         normalImpact: normalImpact.length, // Add this line
2515:         lowImpact: lowImpact.length,
2516:         total: genes.length,
2517:         genes: detailedGenes,
2518:     };
2519:
2520:     return acc;
2521: },
2522: {} as Record<
2523:     string,
2524:     {
2525:         highImpact: number;
2526:         moderateImpact: number;
2527:         normalImpact: number; // Add this to type definition
2528:         lowImpact: number;
2529:         total: number;
2530:         genes: any[];
2531:     }
2532: >
2533: );
2534:
2535: // Analyze sports and fitness in detail
2536: const sportsAnalysis = Object.entries(sportsAndFitness)
2537:     .filter(
2538:         ([key]) => ![ "quote", "description", "customImages" ].includes(key)
2539:     )
2540:     .reduce((acc, [section, groups]) => {
2541:         if (Array.isArray(groups)) {
2542:             const allFields = groups.reduce((fieldAcc, group) => {
2543:                 return fieldAcc.concat(Object.entries(group.fields || {}));
2544:             }, [] as any[]);
2545:
2546:             const detailedTraits = allFields.map(
2547:                 ([trait, data]: [string, any]) => {
2548:                     const selectedImageKey = data.level
2549:                         ?.split("-")

```

```

2550:         .pop()
2551:         ?.toLowerCase();
2552:         const hasImage = !!(
2553:             reportData.sportsAndFitness.customImages?.[
2554:                 selectedImageKey
2555:             ] || defaultImageMap[selectedImageKey]
2556:         );
2557:
2558:         return {
2559:             name: trait,
2560:             level: data.level,
2561:             description: data.description,
2562:             label: data.label,
2563:             imageKey: selectedImageKey,
2564:             hasImage: hasImage,
2565:         };
2566:     }
2567: );
2568:
2569:     acc[section] = {
2570:         totalTraits: allFields.length,
2571:         traits: detailedTraits,
2572:     };
2573: }
2574: return acc;
2575: }, {} as Record<string, { totalTraits: number; traits: any[] }>);
2576:
2577: // Analyze digestive health in detail
2578: const digestiveAnalysis = Object.entries(
2579:     digestiveHealth.data || {}
2580: ).reduce((acc, [key, item]) => {
2581:     acc[key] = {
2582:         title: item.title,
2583:         sensitivity: item.sensitivity,
2584:         good: item.good,
2585:         bad: item.bad,
2586:         icon: item.icon,
2587:         hasIcon: !!item.icon,
2588:     };
2589:     return acc;
2590: }, {} as Record<string, any>);
2591:
2592: // Analyze genes & addiction in detail
2593: // Analyze genes & addiction in detail
2594: const addictionAnalysis = Object.entries(
2595:     genesAndAddiction.data || {}
2596: ).reduce((acc, [key, entry]) => {
2597:     acc[key] = {
2598:         title: entry.title,
2599:         icon: entry.icon, // Add this line
2600:         sensitivityIcon: entry.sensitivityIcon, // Add this line
2601:         hasIcon: !!entry.icon,
2602:         hasSensitivityIcon: !!entry.sensitivityIcon,
2603:     };
2604:     return acc;
2605: }, {} as Record<string, any>);
2606:
2607: // Analyze sleep & rest in detail
2608: const sleepAnalysis = Object.entries(sleepAndRest.data || {}).reduce(
2609:     (acc, [key, entry]) => {
2610:         acc[key] = {
2611:             title: entry.title,
2612:             intervention: entry.intervention,
2613:             image: (entry as any).image,
2614:             hasImage: !!entry.image,
2615:         };
2616:         return acc;
2617:     },
2618:     {} as Record<string, any>
2619: );
2620:
2621: // Analyze allergies in detail
2622: const allergyAnalysis = Object.entries(

```

```

2623:         allergiesAndSensitivity.data || {}
2624:     ).reduce((acc, [key, entry]) => {
2625:         const allergyEntry = entry as { title?: string; image?: string };
2626:         acc[key] = {
2627:             title: allergyEntry.title,
2628:             image: allergyEntry.image,
2629:             hasImage: !!allergyEntry.image,
2630:         };
2631:         return acc;
2632:     }, {} as Record<string, any>);
2633:
2634:     // Calculate overall scores and metrics
2635:     const totalDietScore =
2636:         Object.values(dietAnalysis).reduce(
2637:             (sum, cat) => sum + cat.avgScore,
2638:             0
2639:         ) / Object.keys(dietAnalysis).length || 0;
2640:     const totalNutritionScore =
2641:         Object.values(nutritionAnalysis).reduce(
2642:             (sum, cat) => sum + cat.avgScore,
2643:             0
2644:         ) / Object.keys(nutritionAnalysis).length || 0;
2645:     const totalLifestyleStrengths = Object.values(
2646:         lifestyleAnalysis
2647:     ).reduce((sum, cat) => sum + cat.strengths, 0);
2648:     const totalLifestyleImprovements = Object.values(
2649:         lifestyleAnalysis
2650:     ).reduce((sum, cat) => sum + cat.improvements, 0);
2651:     const totalMetabolicHigh = Object.values(metabolicAnalysis).reduce(
2652:         (sum, area) => sum + area.highImpact,
2653:         0
2654:     );
2655:     const totalMetabolicLow = Object.values(metabolicAnalysis).reduce(
2656:         (sum, area) => sum + area.lowImpact,
2657:         0
2658:     );
2659:     const totalMetabolicModerate = Object.values(
2660:         metabolicAnalysis
2661:     ).reduce((sum, area) => sum + area.moderateImpact, 0);
2662:
2663:     const totalMetabolicNormal = Object.values(metabolicAnalysis).reduce(
2664:         (total, area) => total + (area.normalImpact || 0),
2665:         0
2666:     );
2667:
2668:     return (
2669:         <div className="space-y-4 mb-4">
2670:             { /* Detailed Diet Analysis */ }
2671:             { Object.keys(dietAnalysis).length > 0 && (
2672:                 <Card className="border shadow-sm">
2673:                     <CardHeader className="pb-2">
2674:                         <CardTitle className="text-sm font-bold text-center mb-5 text-green-800">
2675:                             COMPREHENSIVE DIETARY GENETIC ANALYSIS
2676:                         </CardTitle>
2677:                     </CardHeader>
2678:
2679:                     <CardContent className="text-xs text-gray-700">
2680:                         <div className="w-full h-full text-center border rounded p-2 bg-gradient-to-br from-gr
2681:                         <p className="font-semibold text-green-700 mb-1">
2682:                             GENOMICS & DIET
2683:                         </p>
2684:                         <div className="h-80">
2685:                             { " " }
2686:                             { /* doubled height */ }
2687:                             <ResponsiveContainer width="100%" height="100%">
2688:                                 <LineChart
2689:                                     data={Object.values(groupedDietFields).flatMap(
2690:                                         (fields) =>
2691:                                             fields.map((data) => ({
2692:                                                 field: data.label,
2693:                                                 score: Number(
2694:                                                     data.score ??
2695:                                                     (data.high ? 8 : data.medium ? 6 : 4)

```

```

2696:         ),
2697:     })))
2698: })
2699: margin={{ top: 10, right: 8, left: 30, bottom: 60 }}
2700: >
2701: <XAxis
2702:   dataKey="field"
2703:   axisLine
2704:   tickLine
2705:   tick={{ fontSize: 9, fill: "#16a34a" }}
2706:   stroke="#16a34a"
2707:   height={40}
2708:   angle={-45}
2709:   textAnchor="end"
2710:   interval={0}
2711:   tickFormatter={(value) => {
2712:     const str = String(value).toLowerCase();
2713:     return (
2714:       str.charAt(0).toUpperCase() + str.slice(1)
2715:     );
2716:   }}
2717: />
2718:
2719: <YAxis
2720:   axisLine
2721:   tickLine
2722:   tick={{ fontSize: 9, fill: "#16a34a" }}
2723:   stroke="#16a34a"
2724:   width={30}
2725:   domain={[1, 10]}
2726:   ticks={[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]}
2727: />
2728: <Tooltip
2729:   contentStyle={{
2730:     backgroundColor: "#f0fdf4",
2731:     border: "1px solid #16a34a",
2732:     borderRadius: "4px",
2733:     fontSize: "10px",
2734:   }}
2735:   formatter={(value, name, props) => [
2736:     value,
2737:     props.payload.field,
2738:   ]}
2739: />
2740: <Line
2741:   type="monotone"
2742:   dataKey="score"
2743:   stroke="#16a34a"
2744:   strokeWidth={2}
2745:   dot={{ fill: "#16a34a", strokeWidth: 1, r: 3 }}
2746:   activeDot={{
2747:     r: 4,
2748:     fill: "#16a34a",
2749:     stroke: "#fff",
2750:     strokeWidth: 1,
2751:   }}
2752: />
2753: </LineChart>
2754: </ResponsiveContainer>
2755: </div>
2756: <p className="text-xs mt-1">
2757:   Diet-related genetic scores
2758: </p>
2759: </div>
2760:
2761: <div className="mb-3 p-2">
2762:   <p className="font-semibold my-3">
2763:     OVERALL DIET COMPATIBILITY: {totalDietScore.toFixed(1)}
2764:     /10
2765:   </p>
2766:   <div className="grid grid-cols-3 gap-2 text-center">
2767:     <div>
2768:       <p className="text-green-600 font-bold">

```



```

2769:         {Object.values(dietAnalysis).reduce(
2770:             (sum, cat) => sum + cat.highScores,
2771:             0
2772:         )}
2773:     </p>
2774:     <p className="text-xs">High Performers (7-10)</p>
2775: </div>
2776: <div>
2777:     <p className="text-yellow-600 font-bold">
2778:         {Object.values(dietAnalysis).reduce(
2779:             (sum, cat) => sum + cat.mediumScores,
2780:             0
2781:         )}
2782:     </p>
2783:     <p className="text-xs">Moderate (4-6)</p>
2784: </div>
2785: <div>
2786:     <p className="text-red-600 font-bold">
2787:         {Object.values(dietAnalysis).reduce(
2788:             (sum, cat) => sum + cat.lowScores,
2789:             0
2790:         )}
2791:     </p>
2792:     <p className="text-xs">Need Attention (0-3)</p>
2793: </div>
2794: </div>
2795: </div>
2796:
2797: { /* Visual Graph Section */ }
2798:
2799: <div className="mb-10 h-70">
2800:     <ResponsiveContainer width="100%" height="100%">
2801:         <LineChart
2802:             data={Object.entries(dietAnalysis).map(
2803:                 ([category, data]) => ({
2804:                     category:
2805:                         category.charAt(0).toUpperCase() +
2806:                         category.slice(1),
2807:                     avgScore: Number(data.avgScore.toFixed(1)),
2808:                     highScores: data.highScores,
2809:                     mediumScores: data.mediumScores,
2810:                     lowScores: data.lowScores,
2811:                 })
2812:             )}
2813:             margin={{ top: 10, right: 8, left: 35, bottom: 60 }}
2814:         >
2815:             <XAxis
2816:                 dataKey="category"
2817:                 axisLine
2818:                 tickLine
2819:                 tick={{ fontSize: 9, fill: "#16a34a" }}
2820:                 stroke="#16a34a"
2821:                 height={30}
2822:                 angle={-45}
2823:                 textAnchor="end"
2824:                 interval={0}
2825:             />
2826:             <YAxis
2827:                 axisLine
2828:                 tickLine
2829:                 tick={{ fontSize: 9, fill: "#16a34a" }}
2830:                 stroke="#16a34a"
2831:                 width={30}
2832:                 domain={[0, 10]}
2833:                 ticks={[0, 2, 4, 6, 8, 10]}
2834:             />
2835:             <Tooltip
2836:                 contentStyle={{
2837:                     backgroundColor: "#f0fdf4",
2838:                     border: "1px solid #16a34a",
2839:                     borderRadius: "4px",
2840:                     fontSize: "10px",
2841:                 }}

```

```

2842:         formatter={ (value, name, props) => {
2843:             if (name === "avgScore") {
2844:                 return [`${value}/10`, "Average Score"];
2845:             }
2846:             return [value, name];
2847:         }}
2848:     />
2849:     <Line
2850:         type="monotone"
2851:         dataKey="avgScore"
2852:         stroke="#16a34a"
2853:         strokeWidth={2}
2854:         dot={{ fill: "#16a34a", strokeWidth: 1, r: 4 }}
2855:         activeDot={{
2856:             r: 5,
2857:             fill: "#16a34a",
2858:             stroke: "#fff",
2859:             strokeWidth: 2,
2860:         }}
2861:     />
2862: </LineChart>
2863: </ResponsiveContainer>
2864: </div>
2865:
2866: <div className="grid grid-cols-2 gap-3">
2867:   {Object.entries(dietAnalysis).map(([category, data]) => (
2868:     <div key={category} className="rounded p-2">
2869:       <p className="font-semibold text-center capitalize mb-1">
2870:         {category}
2871:       </p>
2872:       <div className="mb-2">
2873:         <p>
2874:           Average Score: { " "}
2875:           <span
2876:             className={`font-bold ${
2877:               data.avgScore >= 6
2878:                 ? "text-green-600"
2879:                 : data.avgScore >= 4
2880:                 ? "text-yellow-600"
2881:                 : "text-red-600"
2882:             }`}
2883:           >
2884:             {data.avgScore.toFixed(1)}/10
2885:           </span>
2886:         </p>
2887:         <p>
2888:           Distribution: {data.highScores}High / { " "}
2889:           {data.mediumScores}Medium / {data.lowScores}Low
2890:         </p>
2891:       </div>
2892:
2893:       { /* Mini Bar Chart for Distribution */ }
2894:       <div className="p-1 h-auto mt-3 rounded">
2895:         <div className="h-40">
2896:           <ResponsiveContainer width="100%" height="100%">
2897:             <BarChart
2898:               data={[
2899:                 {
2900:                   name: "High",
2901:                   value: data.highScores,
2902:                   color: "#dc2626",
2903:                 },
2904:                 {
2905:                   name: "Medium",
2906:                   value: data.mediumScores,
2907:                   color: "#ca8a04",
2908:                 },
2909:                 {
2910:                   name: "Low",
2911:                   value: data.lowScores,
2912:                   color: "#16a34a",
2913:                 },
2914:               ]}

```

```

2915:         margin={{
2916:             top: 5,
2917:             right: 5,
2918:             left: 5,
2919:             bottom: 15,
2920:         }}
2921:     >
2922:     <XAxis
2923:         dataKey="name"
2924:         axisLine={false}
2925:         tickLine={false}
2926:         tick={{ fontSize: 8, fill: "#374151" }}
2927:     />
2928:     <YAxis hide />
2929:     <Tooltip
2930:         contentStyle={{
2931:             backgroundColor: "#f9fafb",
2932:             border: "1px solid #d1d5db",
2933:             borderRadius: "4px",
2934:             fontSize: "10px",
2935:             height: "full",
2936:             borderBottomColor: "black",
2937:         }}
2938:         formatter={(value, name) => [
2939:             `${value} genes`,
2940:             name,
2941:         ]}
2942:     />
2943:     <Bar
2944:         dataKey="value"
2945:         fill="#16a34a"
2946:         radius={[2, 2, 0, 0]}
2947:     >
2948:         {[
2949:             {
2950:                 name: "High",
2951:                 value: data.highScores,
2952:                 color: "#dc2626",
2953:             },
2954:             {
2955:                 name: "Medium",
2956:                 value: data.mediumScores,
2957:                 color: "#ca8a04",
2958:             },
2959:             {
2960:                 name: "Low",
2961:                 value: data.lowScores,
2962:                 color: "#16a34a",
2963:             },
2964:         ].map((entry, index) => (
2965:             <Cell
2966:                 key={`cell-${index}`}
2967:                 fill={entry.color}
2968:             />
2969:         )))
2970:     </Bar>
2971: </BarChart>
2972: </ResponsiveContainer>
2973: </div>
2974: </div>
2975:
2976: {data.recommendations.length > 0 && (
2977:     <div>
2978:         <p className="font-semibold text-xs mb-1">
2979:             Key Recommendations:
2980:         </p>
2981:         <ul className="list-disc list-inside space-y-0">
2982:             {data.recommendations.map((rec, idx) => (
2983:                 <li
2984:                     key={idx}
2985:                     className="text-xs truncate"
2986:                     title={rec}
2987:                 >

```

```

2988:         {rec}
2989:       </li>
2990:     )}
2991:   </ul>
2992: </div>
2993: )}
2994: </div>
2995: )}
2996: </div>
2997: </CardContent>
2998: </Card>
2999: )}
3000:
3001: {/* Detailed Nutrition Analysis */}
3002: {Object.keys(nutritionAnalysis).length > 0 && (
3003:   <Card className="border shadow-sm">
3004:     <CardHeader className="pb-2">
3005:       <CardTitle className="text-sm font-bold text-center text-blue-800">
3006:         DETAILED NUTRITIONAL GENOMICS PROFILE
3007:       </CardTitle>
3008:     </CardHeader>
3009:     <CardContent className="text-xs text-gray-700">
3010:       <div className="w-full print:w-[700px] mx-auto mb-6 print:mb-4">
3011:         {/* Chart for Nutrition */}
3012:         <div className="mb-4 text-center border rounded p-2 bg-gradient-to-br from-blue-50 to-white">
3013:           <p className="font-semibold text-blue-700 mb-1">
3014:             NUTRITION SCORE OVERVIEW
3015:           </p>
3016:           <div className="h-80">
3017:             <ResponsiveContainer width="100%" height="100%">
3018:               <LineChart
3019:                 data={Object.entries(
3020:                   reportData.nutritionData?.data || {}
3021:                 ).flatMap(([sectionName, sectionData]) =>
3022:                   Object.entries(sectionData).map(
3023:                     ([key, data]) => ({
3024:                       field: key
3025:                         .replace(/[A-Z]/g, " $1")
3026:                         .replace(/^../, (str) =>
3027:                           str.toUpperCase()
3028:                         )
3029:                       ,
3030:                       score: Number(
3031:                         data.score ??
3032:                         (data.high ? 8 : data.medium ? 6 : 4)
3033:                       )
3034:                     })
3035:                   )
3036:                 }
3037:                 margin={{
3038:                   top: 10,
3039:                   right: 8,
3040:                   left: 30,
3041:                   bottom: 60,
3042:                 }}
3043:               <XAxis
3044:                 dataKey="field"
3045:                 axisLine
3046:                 tickLine
3047:                 tick={{ fontSize: 9, fill: "#2563eb" }}
3048:                 stroke="#2563eb"
3049:                 height={40}
3050:                 angle={-45}
3051:                 textAnchor="end"
3052:                 interval={0}
3053:               />
3054:               <YAxis
3055:                 axisLine
3056:                 tickLine
3057:                 tick={{ fontSize: 9, fill: "#2563eb" }}
3058:                 stroke="#2563eb"
3059:                 width={30}
3060:                 domain={[1, 10]}

```

```

3061:         ticks={[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]}
3062:     />
3063:     <Tooltip
3064:       contentStyle={{
3065:         backgroundColor: "#eff6ff",
3066:         border: "1px solid #2563eb",
3067:         borderRadius: "4px",
3068:         fontSize: "10px",
3069:       }}
3070:       formatter={(value, name, props) => [
3071:         value,
3072:         props.payload.field,
3073:       ]}
3074:     />
3075:     <Line
3076:       type="monotone"
3077:       dataKey="score"
3078:       stroke="#2563eb"
3079:       strokeWidth={2}
3080:       dot={{ fill: "#2563eb", strokeWidth: 1, r: 3 }}
3081:       activeDot={{
3082:         r: 4,
3083:         fill: "#2563eb",
3084:         stroke: "#fff",
3085:         strokeWidth: 1,
3086:       }}
3087:     />
3088:   </LineChart>
3089: </ResponsiveContainer>
3090: </div>
3091: <p className="text-xs mt-1">
3092:   Nutrient-level genetic scores
3093: </p>
3094: </div>
3095: </div>
3096: <div className="mb-3 p-2 ">
3097:   <p className="font-semibold my-3">
3098:     NUTRITION GENETIC SCORE:{" " }
3099:     {totalNutritionScore.toFixed(1)}/10
3100:   </p>
3101:   <div className="grid grid-cols-3 gap-2 text-center">
3102:     <div>
3103:       <p className="text-red-600 font-bold">
3104:         {Object.values(nutritionAnalysis).reduce(
3105:           (sum, cat) => sum + cat.highImpact,
3106:           0
3107:         )}
3108:       </p>
3109:       <p className="text-xs">High Impact Factors</p>
3110:     </div>
3111:     <div>
3112:       <p className="text-yellow-600 font-bold">
3113:         {Object.values(nutritionAnalysis).reduce(
3114:           (sum, cat) => sum + cat.normalImpact,
3115:           0
3116:         )}
3117:       </p>
3118:       <p className="text-xs">Normal Impact</p>
3119:     </div>
3120:     <div>
3121:       <p className="text-green-600 font-bold">
3122:         {Object.values(nutritionAnalysis).reduce(
3123:           (sum, cat) => sum + cat.lowImpact,
3124:           0
3125:         )}
3126:       </p>
3127:       <p className="text-xs">Low Impact Factors</p>
3128:     </div>
3129:   </div>
3130: </div>
3131:
3132: <div className="grid grid-cols-2 gap-3">
3133:   {Object.entries(nutritionAnalysis).map(

```

```

3134:         ([section, data]) => (
3135:             <div
3136:                 key={section}
3137:                 className="border rounded p-2 bg-white"
3138:             >
3139:                 <p className="font-semibold text-center uppercase mb-1">
3140:                     {section.replace(/([A-Z])/g, " $1")}
3141:                 </p>
3142:                 <div className="mb-2">
3143:                     <p>
3144:                         Avg Score:{" "}
3145:                         <span className="font-bold text-blue-600">
3146:                             {data.avgScore.toFixed(1)}/10
3147:                         </span>
3148:                     </p>
3149:                     <p>
3150:                         Impact Distribution: {data.highImpact}H /{" "}
3151:                         {data.normalImpact}N / {data.lowImpact}L
3152:                     </p>
3153:                 </div>
3154:
3155:                 {data.items.length > 0 && (
3156:                     <div>
3157:                         <p className="font-semibold text-xs mb-1">
3158:                             Key Nutrients:
3159:                         </p>
3160:                         <div className="space-y-1">
3161:                             {data.items.slice(0, 3).map((item, idx) => (
3162:                                 <div
3163:                                     key={idx}
3164:                                     className="text-xs flex items-center"
3165:                                 >
3166:                                     <div
3167:                                         className={`w-2 h-2 rounded-full mr-1 ${
3168:                                             item.score >= 7
3169:                                             ? "bg-green-500"
3170:                                             : item.score >= 4
3171:                                             ? "bg-yellow-500"
3172:                                             : "bg-red-500"
3173:                                         }`}
3174:                                     ></div>
3175:                                     <span className="font-medium">
3176:                                         {item.name}:
3177:                                     </span>{" "}
3178:                                     {item.impact} ({item.score}/10)
3179:                                 </div>
3180:                             )]}
3181:                         </div>
3182:                     </div>
3183:                 )}
3184:             </div>
3185:         )
3186:     )}
3187: </div>
3188: </CardContent>
3189: </Card>
3190: )}
3191:
3192: /* Detailed Lifestyle Analysis */
3193: {Object.keys(lifestyleAnalysis).length > 0 && (
3194:     <Card className="border shadow-sm">
3195:         <CardHeader className="pb-2">
3196:             <CardTitle className="text-sm text-center font-bold mb-4 text-yellow-800">
3197:                 COMPREHENSIVE LIFESTYLE GENETIC ASSESSMENT
3198:             </CardTitle>
3199:         </CardHeader>
3200:         <CardContent className="text-xs text-gray-700">
3201:             <div className="mb-3 p-2">
3202:                 <p className="font-semibold mb-3">
3203:                     LIFESTYLE BALANCE OVERVIEW
3204:                 </p>
3205:                 <div className="grid grid-cols-2 gap-2 text-center">
3206:                     <div>

```

```

3207:         <p className="text-green-600 font-bold text-lg">
3208:             {totalLifestyleStrengths}
3209:         </p>
3210:         <p className="text-xs">
3211:             Genetic Strengths to Leverage
3212:         </p>
3213:     </div>
3214:     <div>
3215:         <p className="text-red-600 font-bold text-lg">
3216:             {totalLifestyleImprovements}
3217:         </p>
3218:         <p className="text-xs">Areas Requiring Attention</p>
3219:     </div>
3220: </div>
3221: </div>
3222:
3223: { /* Simple Overview Chart */ }
3224: <div className="mb-4 p-2 bg-gradient-to-br from-yellow-50 to-yellow-100 rounded border
3225:     <p className="font-semibold text-yellow-700 mb-2 text-center">
3226:         YOUR LIFESTYLE BALANCE
3227:     </p>
3228:     <div className="h-48">
3229:         <ResponsiveContainer width="100%" height="100%">
3230:             <BarChart
3231:                 data={Object.entries(lifestyleAnalysis).map(
3232:                     ([category, data]) => ({
3233:                         category: category
3234:                             .replace(/[a-z]([A-Z])/g, "$1 $2")
3235:                             .replace(/[A-Z]([A-Z][a-z])/g, "$1 $2")
3236:                             .replace(/&/g, " & ")
3237:                             .replace(/\\s+/g, " ")
3238:                             .trim(),
3239:                         strengths: data.strengths,
3240:                         improvements: data.improvements,
3241:                     })
3242:                 )}
3243:                 margin={{ top: 10, right: 8, left: 15, bottom: 40 }}
3244:             >
3245:                 <XAxis
3246:                     dataKey="category"
3247:                     axisLine
3248:                     tickLine
3249:                     tick={{ fontSize: 9, fill: "#a16207" }}
3250:                     stroke="#a16207"
3251:                     angle={-45}
3252:                     textAnchor="end"
3253:                     interval={0}
3254:                 />
3255:                 <YAxis
3256:                     axisLine
3257:                     tickLine
3258:                     tick={{ fontSize: 9, fill: "#a16207" }}
3259:                     stroke="#a16207"
3260:                 />
3261:                 <Tooltip
3262:                     contentStyle={{
3263:                         backgroundColor: "#ffffbe",
3264:                         border: "1px solid #a16207",
3265:                         borderRadius: "4px",
3266:                         fontSize: "10px",
3267:                     }}
3268:                 />
3269:                 <Bar
3270:                     dataKey="strengths"
3271:                     fill="#16a34a"
3272:                     name="Strengths"
3273:                     radius={[2, 2, 0, 0]}
3274:                 />
3275:                 <Bar
3276:                     dataKey="improvements"
3277:                     fill="#dc2626"
3278:                     name="Need Attention"
3279:                     radius={[2, 2, 0, 0]}

```

```

3280:         />
3281:     </BarChart>
3282: </ResponsiveContainer>
3283: </div>
3284: <div className="flex justify-center gap-4 mt-1">
3285:     <div className="flex items-center gap-1">
3286:         <div className="w-3 h-3 bg-green-500 rounded"></div>
3287:         <span className="text-xs text-yellow-700">
3288:             Your Strengths
3289:         </span>
3290:     </div>
3291:     <div className="flex items-center gap-1">
3292:         <div className="w-3 h-3 bg-red-500 rounded"></div>
3293:         <span className="text-xs text-yellow-700">
3294:             Need Attention
3295:         </span>
3296:     </div>
3297: </div>
3298: </div>
3299:
3300: <div className="grid grid-cols-2 gap-3">
3301:     {Object.entries(lifestyleAnalysis).map(
3302:         ([category, data]) => (
3303:             <div
3304:                 key={category}
3305:                 className="border rounded p-2 bg-white"
3306:             >
3307:                 <div className="flex items-center gap-2 mb-1">
3308:                     {data.hasImages &&
3309:                         reportData.lifestyleCategoryImages?.[
3310:                             category
3311:                         ] && (
3312:                             <img
3313:                                 src={
3314:                                     reportData.lifestyleCategoryImages[
3315:                                         category
3316:                                     ] || "/placeholder.svg"
3317:                                 }
3318:                                 alt={category}
3319:                                 className="w-4 h-4 rounded object-cover"
3320:                             />
3321:                         )}
3322:                     <p className="font-semibold capitalize">
3323:                         {category.replace(/[A-Z]/g, " $1")}
3324:                     </p>
3325:                 </div>
3326:
3327:                 <div className="mb-2">
3328:                     <p>
3329:                         Balance:{ " "}
3330:                         <span className="text-green-600 font-bold">
3331:                             {data.strengths}
3332:                         </span>{ " "}
3333:                         strengths /{ " "}
3334:                         <span className="text-red-600 font-bold">
3335:                             {data.improvements}
3336:                         </span>{ " "}
3337:                         improvements
3338:                     </p>
3339:                     <p>Total Conditions: {data.total}</p>
3340:                 </div>
3341:
3342:                 { /* Simple Balance Visualization */ }
3343:                 <div className="mb-2">
3344:                     <div className="flex items-center justify-between text-xs mb-1">
3345:                         <span className="text-green-600">
3346:                             Strengths
3347:                         </span>
3348:                         <span className="text-red-600">
3349:                             Need Attention
3350:                         </span>
3351:                     </div>
3352:                     <div className="flex h-3 bg-gray-200 rounded-full overflow-hidden">

```



```

3353:         <div
3354:             className="bg-green-500"
3355:             style={{
3356:                 width:
3357:                     data.total > 0
3358:                     ? `${
3359:                         (data.strengths / data.total) * 100
3360:                     }%`
3361:                     : "0%",
3362:             }}
3363:         ></div>
3364:         <div
3365:             className="bg-red-500"
3366:             style={{
3367:                 width:
3368:                     data.total > 0
3369:                     ? `${
3370:                         (data.improvements / data.total) *
3371:                         100
3372:                     }%`
3373:                     : "0%",
3374:             }}
3375:         ></div>
3376:     </div>
3377:     <div className="flex justify-between text-xs mt-1">
3378:         <span className="text-green-600">
3379:             {data.strengths}
3380:         </span>
3381:         <span className="text-red-600">
3382:             {data.improvements}
3383:         </span>
3384:     </div>
3385: </div>
3386:
3387: {data.conditions.length > 0 && (
3388:     <div>
3389:         <p className="font-semibold text-xs mb-1">
3390:             Key Conditions:
3391:         </p>
3392:         <div className="space-y-1">
3393:             {data.conditions
3394:                 .slice(0, 3)
3395:                 .map((condition, idx) => (
3396:                     <div
3397:                         key={idx}
3398:                         className="text-xs flex items-center gap-1"
3399:                     >
3400:                         {condition.status === "strength" ? (
3401:                             <Check className="text-green-600 w-2 h-2" />
3402:                         ) : (
3403:                             <Check className="text-red-600 w-2 h-2" />
3404:                         )}
3405:                         <span className="truncate">
3406:                             {condition.name}
3407:                         </span>
3408:                     </div>
3409:                 )))
3410:             </div>
3411:         </div>
3412:     </div>
3413: )
3414: )
3415: }}
3416: </div>
3417: </CardContent>
3418: </Card>
3419: )}
3420:
3421: {/* Detailed Metabolic Analysis */}
3422: {Object.keys(metabolicAnalysis).length > 0 && (
3423:     <Card className="border shadow-sm">
3424:         {/* HEADER SECTION */}
3425:         <CardHeader className="pb-2">

```

[illegible]

```

3499:         totalMetabolicLow)) *
3500:         100 || 0
3501:     }%,
3502: }}
3503: >
3504:     { /* Only show number if greater than 0 */
3505:     {totalMetabolicHigh > 0 && totalMetabolicHigh}
3506: </div>
3507:
3508: { /* Moderate Impact Section - Yellow */
3509: <div
3510:     className="bg-yellow-500 flex items-center justify-center text-white text-xs font
3511:     style={{
3512:         width: `${
3513:             (totalMetabolicModerate /
3514:             (totalMetabolicHigh +
3515:                 totalMetabolicModerate +
3516:                 totalMetabolicNormal +
3517:                 totalMetabolicLow)) *
3518:             100 || 0
3519:         }%`,
3520:         }}
3521: >
3522:     {totalMetabolicModerate > 0 &&
3523:     totalMetabolicModerate}
3524: </div>
3525:
3526: { /* Normal Impact Section - Blue (NEW) */
3527: <div
3528:     className="bg-blue-500 flex items-center justify-center text-white text-xs font
3529:     style={{
3530:         width: `${
3531:             (totalMetabolicNormal /
3532:             (totalMetabolicHigh +
3533:                 totalMetabolicModerate +
3534:                 totalMetabolicNormal +
3535:                 totalMetabolicLow)) *
3536:             100 || 0
3537:         }%`,
3538:         }}
3539: >
3540:     {totalMetabolicNormal > 0 && totalMetabolicNormal}
3541: </div>
3542:
3543: { /* Low Impact Section - Green */
3544: <div
3545:     className="bg-green-500 flex items-center justify-center text-white text-xs font
3546:     style={{
3547:         width: `${
3548:             (totalMetabolicLow /
3549:             (totalMetabolicHigh +
3550:                 totalMetabolicModerate +
3551:                 totalMetabolicNormal +
3552:                 totalMetabolicLow)) *
3553:             100 || 0
3554:         }%`,
3555:         }}
3556: >
3557:     {totalMetabolicLow > 0 && totalMetabolicLow}
3558: </div>
3559: </div>
3560: </div>
3561:
3562: { /* STACKED BAR CHART BY METABOLIC AREA */
3563: <div className="h-60">
3564:     <ResponsiveContainer width="100%" height="100%">
3565:         <BarChart
3566:             data={Object.entries(metabolicAnalysis).map(
3567:                 ([area, data]) => ({
3568:                     // Format area name for display
3569:                     area: area
3570:                     .replace(/([a-z])([A-Z])/g, "$1 $2")
3571:                     .replace(/([A-Z])([A-Z][a-z])/g, "$1 $2")

```

```

3572:         .replace(/&/g, " & ")
3573:         .replace(/\\s+/g, " ")
3574:         .trim(),
3575:         // Data for each impact level
3576:         high: data.highImpact,
3577:         moderate: data.moderateImpact,
3578:         normal: data.normalImpact, // NEW
3579:         low: data.lowImpact,
3580:         total: data.total,
3581:     })
3582: })
3583: margin={{
3584:     top: 10,
3585:     right: 8,
3586:     left: 15,
3587:     bottom: 100,
3588: }}
3589: >
3590: { /* X-Axis - Metabolic Area Names */}
3591: <XAxis
3592:     dataKey="area"
3593:     axisLine
3594:     tickLine
3595:     tick={{ fontSize: 9, fill: "#991b1b" }}
3596:     stroke="#991b1b"
3597:     angle={-45}
3598:     textAnchor="end"
3599:     interval={0}
3600: />
3601:
3602: { /* Y-Axis - Gene Count */}
3603: <YAxis
3604:     axisLine
3605:     tickLine
3606:     tick={{ fontSize: 9, fill: "#991b1b" }}
3607:     stroke="#991b1b"
3608:     label={{
3609:         value: "Gene Count",
3610:         angle: -90,
3611:         position: "insideLeft",
3612:     }}
3613: />
3614:
3615: { /* Tooltip on hover */}
3616: <Tooltip
3617:     contentStyle={{
3618:         backgroundColor: "#fef2f2",
3619:         border: "1px solid #991b1b",
3620:         borderRadius: "4px",
3621:         fontSize: "10px",
3622:     }}
3623: />
3624:
3625: { /* Stacked Bars - Each impact level */}
3626: { /* High Impact Bar - Red */}
3627: <Bar
3628:     dataKey="high"
3629:     stackId="impact"
3630:     fill="#dc2626"
3631:     name="High Impact"
3632:     radius={[0, 0, 0, 0]}
3633: />
3634:
3635: { /* Moderate Impact Bar - Yellow */}
3636: <Bar
3637:     dataKey="moderate"
3638:     stackId="impact"
3639:     fill="#eab308"
3640:     name="Moderate Impact"
3641:     radius={[0, 0, 0, 0]}
3642: />
3643:
3644: { /* Normal Impact Bar - Blue (NEW) */}

```

```

3645:         <Bar
3646:             dataKey="normal"
3647:             stackId="impact"
3648:             fill="#3b82f6"
3649:             name="Normal"
3650:             radius={[0, 0, 0, 0]}
3651:         />
3652:
3653:         {/* Low Impact Bar - Green */}
3654:         <Bar
3655:             dataKey="low"
3656:             stackId="impact"
3657:             fill="#16a34a"
3658:             name="Low Impact"
3659:             radius={[2, 2, 0, 0]} // Rounded top corners
3660:         />
3661:     </BarChart>
3662: </ResponsiveContainer>
3663: </div>
3664:
3665: {/* CHART LEGEND */}
3666: <div className="flex justify-center gap-3 mt-1">
3667:     {/* High Impact Legend */}
3668:     <div className="flex items-center gap-1">
3669:         <div className="w-3 h-3 bg-red-500 rounded"></div>
3670:         <span className="text-xs text-red-700">
3671:             High Impact
3672:         </span>
3673:     </div>
3674:
3675:     {/* Moderate Impact Legend */}
3676:     <div className="flex items-center gap-1">
3677:         <div className="w-3 h-3 bg-yellow-500 rounded"></div>
3678:         <span className="text-xs text-red-700">Moderate</span>
3679:     </div>
3680:
3681:     {/* Normal Impact Legend - NEW */}
3682:     <div className="flex items-center gap-1">
3683:         <div className="w-3 h-3 bg-blue-500 rounded"></div>
3684:         <span className="text-xs text-red-700">Normal</span>
3685:     </div>
3686:
3687:     {/* Low Impact Legend */}
3688:     <div className="flex items-center gap-1">
3689:         <div className="w-3 h-3 bg-green-500 rounded"></div>
3690:         <span className="text-xs text-red-700">
3691:             Low Impact
3692:         </span>
3693:     </div>
3694: </div>
3695: </div>
3696:
3697: {/* DETAILED AREA BREAKDOWN CARDS */}
3698: <div className="grid grid-cols-2 gap-3">
3699:     {Object.entries(metabolicAnalysis).map(([area, data]) => (
3700:         <div key={area} className="border rounded p-2 bg-white">
3701:             {/* Area Name */}
3702:             <p className="font-semibold capitalize mb-1">
3703:                 {area
3704:                     .replace(/[a-z]([A-Z])/g, "$1 $2")
3705:                     .replace(/[A-Z]([A-Z][a-z])/g, "$1 $2")
3706:                     .replace(/&/g, " & ")
3707:                     .replace(/\\s+/g, " ")
3708:                     .trim()}
3709:             </p>
3710:
3711:             {/* Area Statistics */}
3712:             <div className="mb-2">
3713:                 <p>Total Genes: {data.total}</p>
3714:                 <p>
3715:                     Impact:{" "}
3716:                     <span className="text-red-600">
3717:                         {data.highImpact}H

```

```

3718:         </span>{ " " }
3719:     /{ " " }
3720:     <span className="text-yellow-600">
3721:         {data.moderateImpact}M
3722:     </span>{ " " }
3723:     /{ " " }
3724:     <span className="text-blue-600">
3725:         {data.normalImpact}N
3726:     </span>{ " " }
3727:     /{ " " }
3728:     <span className="text-green-600">
3729:         {data.lowImpact}L
3730:     </span>
3731: </p>
3732: </div>
3733:
3734: { /* Area-Specific Progress Bar */ }
3735: <div className="mb-2">
3736:     <div className="flex h-4 bg-gray-200 rounded-full overflow-hidden">
3737:         { /* High Impact Section */ }
3738:         <div
3739:             className="bg-red-500"
3740:             style={{
3741:                 width:
3742:                     data.total > 0
3743:                     ? `${
3744:                         (data.highImpact / data.total) * 100
3745:                     }%`
3746:                     : "0%",
3747:             }}
3748:         ></div>
3749:
3750:         { /* Moderate Impact Section */ }
3751:         <div
3752:             className="bg-yellow-500"
3753:             style={{
3754:                 width:
3755:                     data.total > 0
3756:                     ? `${
3757:                         (data.moderateImpact / data.total) *
3758:                         100
3759:                     }%`
3760:                     : "0%",
3761:             }}
3762:         ></div>
3763:
3764:         { /* Normal Impact Section - NEW */ }
3765:         <div
3766:             className="bg-blue-500"
3767:             style={{
3768:                 width:
3769:                     data.total > 0
3770:                     ? `${
3771:                         (data.normalImpact / data.total) * 100
3772:                     }%`
3773:                     : "0%",
3774:             }}
3775:         ></div>
3776:
3777:         { /* Low Impact Section */ }
3778:         <div
3779:             className="bg-green-500"
3780:             style={{
3781:                 width:
3782:                     data.total > 0
3783:                     ? `${
3784:                         (data.lowImpact / data.total) * 100
3785:                     }%`
3786:                     : "0%",
3787:             }}
3788:         ></div>
3789:     </div>
3790:

```

```

3791:         { /* Numbers under the progress bar */ }
3792:     <div className="flex justify-between text-xs mt-1">
3793:         <span className="text-red-600">
3794:             {data.highImpact}
3795:         </span>
3796:         <span className="text-yellow-600">
3797:             {data.moderateImpact}
3798:         </span>
3799:         <span className="text-blue-600">
3800:             {data.normalImpact}
3801:         </span>
3802:         <span className="text-green-600">
3803:             {data.lowImpact}
3804:         </span>
3805:     </div>
3806: </div>
3807:
3808: { /* KEY GENES LIST */ }
3809: {data.genes.length > 0 && (
3810:     <div>
3811:         <p className="font-semibold text-xs mb-1">
3812:             Key Genes:
3813:         </p>
3814:         <div className="space-y-1">
3815:             { /* Show only first 3 genes */ }
3816:             {data.genes.slice(0, 3).map((gene, idx) => (
3817:                 <div key={idx} className="text-xs">
3818:                     <div className="flex items-center gap-1">
3819:                         { /* Color-coded impact indicator */ }
3820:                         <span
3821:                             className={`w-2 h-2 rounded-full ${
3822:                                 gene.impact
3823:                                     ?.toLowerCase()
3824:                                     .includes("high")
3825:                                     ? "bg-red-500"
3826:                                     : gene.impact
3827:                                     ?.toLowerCase()
3828:                                     .includes("moderate") ||
3829:                                     gene.impact
3830:                                     ?.toLowerCase()
3831:                                     .includes("average")
3832:                                     ? "bg-yellow-500"
3833:                                     : gene.impact
3834:                                     ?.toLowerCase()
3835:                                     .includes("normal")
3836:                                     ? "bg-blue-500" // NEW
3837:                                     : "bg-green-500" // Low impact
3838:                             }`}
3839:                         ></span>
3840:                         { /* Gene name and genotype */ }
3841:                         <span className="font-medium">
3842:                             {gene.name}:
3843:                         </span>{ " " }
3844:                         {gene.genotype}
3845:                     </div>
3846:                 </div>
3847:             )})
3848:         </div>
3849:     </div>
3850: )}
3851: </div>
3852: )})
3853: </div>
3854: </CardContent>
3855: </Card>
3856: )}
3857:
3858: { /* Sports & Fitness Analysis (only if data exists) */ }
3859: {Object.keys(sportsAnalysis).length > 0 && (
3860:     <Card className="border shadow-sm">
3861:         <CardHeader className="pb-2">
3862:             <CardTitle className="text-sm font-bold text-orange-800">
3863:                 DETAILED FITNESS GENETIC POTENTIAL

```

```

3864:         </CardTitle>
3865:     </CardHeader>
3866:     <CardContent className="text-xs text-gray-700">
3867:         <div className="mb-3 p-2 bg-white rounded border">
3868:             <p className="font-semibold mb-1">
3869:                 FITNESS GENETIC OVERVIEW
3870:             </p>
3871:             <p>
3872:                 Total Traits Analyzed:{" "}
3873:                 {Object.values(sportsAnalysis).reduce(
3874:                     (sum, section) => sum + section.totalTraits,
3875:                     0
3876:                 )}{"}
3877:                 across {Object.keys(sportsAnalysis).length} categories
3878:             </p>
3879:         </div>
3880:
3881:         <div className="grid grid-cols-2 gap-3">
3882:             {Object.entries(sportsAnalysis).map(([section, data]) => (
3883:                 <div
3884:                     key={section}
3885:                     className="border rounded p-2 bg-white"
3886:                 >
3887:                     <p className="font-semibold capitalize mb-1">
3888:                         {section.replace(/[A-Z]/g, " $1")}
3889:                     </p>
3890:                     <p className="mb-2">Traits: {data.totalTraits}</p>
3891:
3892:                     {data.traits.length > 0 && (
3893:                         <div>
3894:                             <p className="font-semibold text-xs mb-1">
3895:                                 Key Traits:
3896:                             </p>
3897:                             <div className="space-y-1">
3898:                                 {data.traits.slice(0, 3).map((trait, idx) => (
3899:                                     <div
3900:                                         key={idx}
3901:                                         className="text-xs flex items-center gap-2"
3902:                                     >
3903:                                         {trait.hasImage && (
3904:                                             <img
3905:                                                 src={
3906:                                                     reportData.sportsAndFitness
3907:                                                         .customImages?.[trait.imageKey] ||
3908:                                                     defaultImageMap[trait.imageKey] ||
3909:                                                     "/sports/default.png"
3910:                                                 }
3911:                                                 alt={trait.imageKey}
3912:                                                 className="w-4 h-4 object-contain"
3913:                                             />
3914:                                         )}
3915:                                     <div>
3916:                                         <span className="font-medium">
3917:                                             {trait.label}:
3918:                                         </span>{" "}
3919:                                         {trait.level
3920:                                             ?.split("-")
3921:                                             .pop()
3922:                                             ?.toUpperCase()}
3923:                                     </div>
3924:                                 </div>
3925:                             )}
3926:                         </div>
3927:                     </div>
3928:                 )}
3929:             </div>
3930:         )}
3931:     </div>
3932: </CardContent>
3933: </Card>
3934: )}
3935:
3936: {/* Digestive Health Analysis (only if data exists) */}

```



```

3937:         {Object.keys(digestiveAnalysis).length > 0 && (
3938:             <Card className="border shadow-sm">
3939:                 <CardHeader className="pb-2">
3940:                     <CardTitle className="text-sm font-bold text-pink-800">
3941:                         DIGESTIVE HEALTH GENETIC ANALYSIS
3942:                     </CardTitle>
3943:                 </CardHeader>
3944:                 <CardContent className="text-xs text-gray-700">
3945:                     <div className="mb-3 p-2">
3946:                         <p className="font-semibold mb-1">
3947:                             DIGESTIVE SENSITIVITY OVERVIEW
3948:                         </p>
3949:                         <p>
3950:                             Total Parameters:{ " " }
3951:                             {Object.keys(digestiveAnalysis).length}
3952:                         </p>
3953:                         <div className="grid grid-cols-2 gap-2 text-center mt-2">
3954:                             <div>
3955:                                 <p className="text-red-600 font-bold">
3956:                                     {
3957:                                         Object.values(digestiveAnalysis).filter(
3958:                                             (item) => item.sensitivity === "High"
3959:                                         ).length
3960:                                     }
3961:                                 </p>
3962:                                 <p className="text-xs">High Sensitivity</p>
3963:                             </div>
3964:                             <div>
3965:                                 <p className="text-green-600 font-bold">
3966:                                     {
3967:                                         Object.values(digestiveAnalysis).filter(
3968:                                             (item) => item.sensitivity === "Low"
3969:                                         ).length
3970:                                     }
3971:                                 </p>
3972:                                 <p className="text-xs">Low Sensitivity</p>
3973:                             </div>
3974:                         </div>
3975:                     </div>
3976:
3977:                     { /* Simple Overview Chart */ }
3978:                     <div className="mb-4 p-2 bg-gradient-to-br from-pink-50 to-pink-100 rounded border">
3979:                         <p className="font-semibold text-pink-700 mb-2 text-center">
3980:                             YOUR DIGESTIVE SENSITIVITY PROFILE
3981:                         </p>
3982:
3983:                         { /* Overall Sensitivity Bar */ }
3984:                         <div className="mb-3">
3985:                             <div className="flex justify-between text-xs mb-1">
3986:                                 <span className="text-red-600">High Sensitivity</span>
3987:                                 <span className="text-green-600">
3988:                                     Low Sensitivity
3989:                                 </span>
3990:                             </div>
3991:                             <div className="flex h-6 bg-gray-200 rounded-full overflow-hidden">
3992:                                 <div
3993:                                     className="bg-red-500 flex items-center justify-center text-white text-xs font
3994:                                     style={{
3995:                                         width: `${
3996:                                             (Object.values(digestiveAnalysis).filter(
3997:                                                 (item) => item.sensitivity === "High"
3998:                                             ).length /
3999:                                             Object.keys(digestiveAnalysis).length) *
4000:                                             100 || 0
4001:                                         }%`,
4002:                                     }}
4003:                                 >
4004:                                     {Object.values(digestiveAnalysis).filter(
4005:                                         (item) => item.sensitivity === "High"
4006:                                     ).length > 0 &&
4007:                                     Object.values(digestiveAnalysis).filter(
4008:                                         (item) => item.sensitivity === "High"
4009:                                     ).length}

```

```

4010:         </div>
4011:     <div
4012:         className="bg-green-500 flex items-center justify-center text-white text-xs font
4013:         style={{
4014:             width: `${
4015:                 (Object.values(digestiveAnalysis).filter(
4016:                     (item) => item.sensitivity === "Low"
4017:                 ).length /
4018:                 Object.keys(digestiveAnalysis).length) *
4019:                 100 || 0
4020:             }%`,
4021:         }}
4022:     >
4023:         {Object.values(digestiveAnalysis).filter(
4024:             (item) => item.sensitivity === "Low"
4025:         ).length > 0 &&
4026:             Object.values(digestiveAnalysis).filter(
4027:                 (item) => item.sensitivity === "Low"
4028:             ).length}
4029:     </div>
4030: </div>
4031: <p className="text-xs text-center mt-1 text-pink-600">
4032:     {Object.values(digestiveAnalysis).filter(
4033:         (item) => item.sensitivity === "High"
4034:     ).length >
4035:         Object.values(digestiveAnalysis).filter(
4036:             (item) => item.sensitivity === "Low"
4037:         ).length
4038:         ? "You may need to be more careful with certain foods"
4039:         : "You have good genetic tolerance for most foods"}
4040: </p>
4041: </div>
4042:
4043: { /* Simple Visual List */ }
4044: <div className="h-full overflow-y-auto">
4045:     <div className="space-y-2">
4046:         {Object.entries(digestiveAnalysis)
4047:             .sort(([, a], [, b]) =>
4048:                 a.sensitivity === "High" ? -1 : 1
4049:             )
4050:             .map(([key, item]) => (
4051:                 <div
4052:                     key={key}
4053:                     className="flex items-center justify-between bg-white rounded p-2 border"
4054:                 >
4055:                     <div className="flex items-center gap-2">
4056:                         {item.icon && (
4057:                             <img
4058:                                 src={`~/digestive/${item.icon}.png`}
4059:                                 alt={item.title}
4060:                                 className="w-3 h-3 object-contain"
4061:                             />
4062:                         )}
4063:                         <span className="text-xs font-medium">
4064:                             {item.title}
4065:                         </span>
4066:                     </div>
4067:                     <div className="flex items-center gap-2">
4068:                         <div
4069:                             className={`w-3 h-3 rounded-full ${
4070:                                 item.sensitivity === "High"
4071:                                 ? "bg-red-500"
4072:                                 : "bg-green-500"
4073:                             }`}
4074:                         ></div>
4075:                         <span
4076:                             className={`text-xs font-bold ${
4077:                                 item.sensitivity === "High"
4078:                                 ? "text-red-600"
4079:                                 : "text-green-600"
4080:                             }`}
4081:                         >
4082:                             {item.sensitivity}

```

```

4083:         </span>
4084:     </div>
4085: </div>
4086:     )})
4087: </div>
4088: </div>
4089:
4090: <div className="flex justify-center gap-4 mt-2">
4091:     <div className="flex items-center gap-1">
4092:         <div className="w-3 h-3 bg-red-500 rounded-full"></div>
4093:         <span className="text-xs text-pink-700">
4094:             High Sensitivity
4095:         </span>
4096:     </div>
4097:     <div className="flex items-center gap-1">
4098:         <div className="w-3 h-3 bg-green-500 rounded-full"></div>
4099:         <span className="text-xs text-pink-700">
4100:             Low Sensitivity
4101:         </span>
4102:     </div>
4103: </div>
4104: </div>
4105:
4106: <div className="grid grid-cols-2 gap-3 h-full">
4107:     {Object.entries(digestiveAnalysis).map(([key, item]) => (
4108:         <div
4109:             key={key}
4110:             className="border rounded p-2 h-full bg-white"
4111:         >
4112:             <div className="flex items-center gap-2 mb-1">
4113:                 {item.icon && (
4114:                     <img
4115:                         src={`\digestive/${item.icon}.png`}
4116:                         alt={item.title}
4117:                         className="w-4 h-4 object-contain"
4118:                     />
4119:                 )}
4120:                 <p className="font-semibold">{item.title}</p>
4121:             </div>
4122:
4123:             {/\* Sensitivity Indicator Bar \*/}
4124:             <div className="mb-2">
4125:                 <div className="flex justify-between items-center mb-1">
4126:                     <span className="text-xs">
4127:                         Sensitivity Level:
4128:                     </span>
4129:                     <span
4130:                         className={`text-xs font-bold ${
4131:                             item.sensitivity === "High"
4132:                             ? "text-red-600"
4133:                             : "text-green-600"
4134:                         }`}
4135:                     >
4136:                         {item.sensitivity}
4137:                     </span>
4138:                 </div>
4139:                 <div className="h-3 bg-gray-200 rounded-full overflow-hidden">
4140:                     <div
4141:                         className={`h-full ${
4142:                             item.sensitivity === "High"
4143:                             ? "bg-red-500"
4144:                             : "bg-green-500"
4145:                         }`}
4146:                         style={{
4147:                             width:
4148:                                 item.sensitivity === "High" ? "85%" : "25%",
4149:                         }}
4150:                     ></div>
4151:                 </div>
4152:             </div>
4153:
4154:             {item.sensitivity === "High" && item.bad && (
4155:                 <p className="text-xs text-red-700">

```

```

4156:             {item.bad.substring(0, 5000)}...
4157:         </p>
4158:     )}
4159:     {item.sensitivity === "Low" && item.good && (
4160:         <p className="text-xs text-green-700">
4161:             {item.good.substring(0, 5000)}...
4162:         </p>
4163:     )}
4164: </div>
4165: )}}
4166: </div>
4167: </CardContent>
4168: </Card>
4169: )}
4170:
4171: {/* Addiction Analysis (only if data exists) */}
4172: {Object.keys(addictionAnalysis).length > 0 && (
4173:     <Card className="border shadow-sm">
4174:         <CardHeader className="pb-2">
4175:             <CardTitle className="text-sm font-bold text-indigo-800">
4176:                 ADDICTION SUSCEPTIBILITY GENETIC PROFILE
4177:             </CardTitle>
4178:         </CardHeader>
4179:         <CardContent className="text-xs text-gray-700">
4180:             <div className="mb-3 p-2 bg-white rounded ">
4181:                 <p className="font-semibold mb-1">
4182:                     ADDICTION GENETIC FACTORS
4183:                 </p>
4184:                 <p>
4185:                     Total Substances Analyzed:{" "}
4186:                     {Object.keys(addictionAnalysis).length}
4187:                 </p>
4188:             </div>
4189:
4190:             {/* Simple Risk Overview */}
4191:             <div className="mb-4 p-2 bg-gradient-to-br from-indigo-50 to-indigo-100 rounded border">
4192:                 <p className="font-semibold text-indigo-700 mb-2 text-center">
4193:                     YOUR GENETIC SUSCEPTIBILITY OVERVIEW
4194:                 </p>
4195:
4196:                 {/* Risk Level Summary */}
4197:                 <div className="mb-3 p-2 bg-white rounded border">
4198:                     <div className="grid grid-cols-3 gap-2 text-center">
4199:                         <div>
4200:                             <p className="text-red-600 font-bold text-lg">
4201:                                 {
4202:                                     Object.values(addictionAnalysis).filter(
4203:                                         (item) =>
4204:                                             item.sensitivityIcon?.includes("high") ||
4205:                                             item.title?.toLowerCase().includes("high")
4206:                                         ).length
4207:                                 }
4208:                             </p>
4209:                             <p className="text-xs">Higher Risk</p>
4210:                         </div>
4211:                         <div>
4212:                             <p className="text-yellow-600 font-bold text-lg">
4213:                                 {
4214:                                     Object.values(addictionAnalysis).filter(
4215:                                         (item) =>
4216:                                             item.sensitivityIcon?.includes("medium") ||
4217:                                             item.sensitivityIcon?.includes(
4218:                                                 "moderate"
4219:                                             ) ||
4220:                                             item.title
4221:                                                 ?.toLowerCase()
4222:                                                 .includes("medium") ||
4223:                                             item.title
4224:                                                 ?.toLowerCase()
4225:                                                 .includes("moderate")
4226:                                         ).length
4227:                                 }
4228:                             </p>

```

```

4229:         <p className="text-xs">Moderate Risk</p>
4230:     </div>
4231: <div>
4232:     <p className="text-green-600 font-bold text-lg">
4233:         {
4234:             Object.values(addictionAnalysis).filter(
4235:                 (item) =>
4236:                     item.sensitivityIcon?.includes("low") ||
4237:                     item.title?.toLowerCase().includes("low")
4238:                 ).length
4239:             }
4240:         </p>
4241:     <p className="text-xs">Lower Risk</p>
4242: </div>
4243: </div>
4244: </div>
4245:
4246: { /* Awareness Message */ }
4247: <div className="bg-blue-50 border border-blue-200 rounded p-2 mb-3">
4248:     <p className="text-xs text-blue-800 text-center">
4249:         <strong>Important:</strong> Genetics is just one
4250:         factor. Environment, lifestyle, and personal choices
4251:         play major roles in addiction risk.
4252:     </p>
4253: </div>
4254:
4255: { /* Simple Visual List with Risk Levels */ }
4256: <div className="h-48 overflow-y-auto">
4257:     <div className="space-y-2">
4258:         {Object.entries(addictionAnalysis)
4259:             .sort(([, a], [, b]) => {
4260:                 // Sort by risk level (high risk first)
4261:                 const getRiskLevel = (item) => {
4262:                     if (
4263:                         item.sensitivityIcon?.includes("high") ||
4264:                         item.title?.toLowerCase().includes("high")
4265:                     )
4266:                         return 3;
4267:                     if (
4268:                         item.sensitivityIcon?.includes("medium") ||
4269:                         item.sensitivityIcon?.includes("moderate") ||
4270:                         item.title
4271:                             ?.toLowerCase()
4272:                             .includes("medium") ||
4273:                         item.title?.toLowerCase().includes("moderate")
4274:                     )
4275:                         return 2;
4276:                     return 1;
4277:                 };
4278:                 return getRiskLevel(b) - getRiskLevel(a);
4279:             })
4280:             .map(([key, item]) => {
4281:                 const isHighRisk =
4282:                     item.sensitivityIcon?.includes("high") ||
4283:                     item.title?.toLowerCase().includes("high");
4284:                 const isMediumRisk =
4285:                     item.sensitivityIcon?.includes("medium") ||
4286:                     item.sensitivityIcon?.includes("moderate") ||
4287:                     item.title?.toLowerCase().includes("medium") ||
4288:                     item.title?.toLowerCase().includes("moderate");
4289:                 const riskLevel = isHighRisk
4290:                     ? "Higher"
4291:                     : isMediumRisk
4292:                     ? "Moderate"
4293:                     : "Lower";
4294:                 const riskColor = isHighRisk
4295:                     ? "red"
4296:                     : isMediumRisk
4297:                     ? "yellow"
4298:                     : "green";
4299:
4300:                 return (
4301:

```

```

4302:         key={key}
4303:         className="flex items-center justify-between bg-white rounded p-2 border
4304:     >
4305:     <div className="flex items-center gap-2">
4306:         {item.icon && (
4307:             <img
4308:                 src={` /addiction/icons/${item.icon}.png`}
4309:                 alt={item.title}
4310:                 className="w-4 h-4 object-contain"
4311:             />
4312:         )}
4313:         <span className="text-xs font-medium">
4314:             {item.title}
4315:         </span>
4316:     </div>
4317:     <div className="flex items-center gap-2">
4318:         <div className="flex items-center gap-1">
4319:             <div
4320:                 className={`w-3 h-3 rounded-full ${
4321:                     riskColor === "red"
4322:                     ? "bg-red-500"
4323:                     : riskColor === "yellow"
4324:                     ? "bg-yellow-500"
4325:                     : "bg-green-500"
4326:                 }`}
4327:             ></div>
4328:             <span
4329:                 className={`text-xs font-bold ${
4330:                     riskColor === "red"
4331:                     ? "text-red-600"
4332:                     : riskColor === "yellow"
4333:                     ? "text-yellow-600"
4334:                     : "text-green-600"
4335:                 }`}
4336:             >
4337:                 {riskLevel}
4338:             </span>
4339:         </div>
4340:     </div>
4341: </div>
4342: );
4343: }}}
4344: </div>
4345: </div>
4346:
4347: <div className="flex justify-center gap-4 mt-2">
4348:     <div className="flex items-center gap-1">
4349:         <div className="w-3 h-3 bg-red-500 rounded-full"></div>
4350:         <span className="text-xs text-indigo-700">
4351:             Higher Risk
4352:         </span>
4353:     </div>
4354:     <div className="flex items-center gap-1">
4355:         <div className="w-3 h-3 bg-yellow-500 rounded-full"></div>
4356:         <span className="text-xs text-indigo-700">
4357:             Moderate Risk
4358:         </span>
4359:     </div>
4360:     <div className="flex items-center gap-1">
4361:         <div className="w-3 h-3 bg-green-500 rounded-full"></div>
4362:         <span className="text-xs text-indigo-700">
4363:             Lower Risk
4364:         </span>
4365:     </div>
4366: </div>
4367: </div>
4368:
4369: <div className="grid grid-cols-2 gap-3">
4370:     {Object.entries(addictionAnalysis).map(([key, item]) => {
4371:         const isHighRisk =
4372:             item.sensitivityIcon?.includes("high") ||
4373:             item.title?.toLowerCase().includes("high");
4374:         const isMediumRisk =

```

```

4375:         item.sensitivityIcon?.includes("medium") ||
4376:         item.sensitivityIcon?.includes("moderate") ||
4377:         item.title?.toLowerCase().includes("medium") ||
4378:         item.title?.toLowerCase().includes("moderate");
4379:     const riskLevel = isHighRisk
4380:         ? "Higher Risk"
4381:         : isMediumRisk
4382:         ? "Moderate Risk"
4383:         : "Lower Risk";
4384:     const riskColor = isHighRisk
4385:         ? "red"
4386:         : isMediumRisk
4387:         ? "yellow"
4388:         : "green";
4389:
4390:     return (
4391:         <div
4392:             key={key}
4393:             className="border rounded p-2 bg-white"
4394:         >
4395:             <div className="flex items-center gap-2 mb-2">
4396:                 {item.icon && (
4397:                     <img
4398:                         src={`\addiction/icons/${item.icon}.png`}
4399:                         alt={item.title}
4400:                         className="w-6 h-6 object-contain"
4401:                     />
4402:                 )}
4403:                 <p className="font-semibold">{item.title}</p>
4404:             </div>
4405:
4406:             {/* Risk Level Indicator */}
4407:             <div className="mb-2">
4408:                 <div className="flex justify-between items-center mb-1">
4409:                     <span className="text-xs">Genetic Risk:</span>
4410:                     <span
4411:                         className={`text-xs font-bold ${
4412:                             riskColor === "red"
4413:                                 ? "text-red-600"
4414:                                 : riskColor === "yellow"
4415:                                 ? "text-yellow-600"
4416:                                 : "text-green-600"
4417:                         }`}
4418:                     >
4419:                         {riskLevel}
4420:                     </span>
4421:                 </div>
4422:                 <div className="h-3 bg-gray-200 rounded-full overflow-hidden">
4423:                     <div
4424:                         className={`h-full ${
4425:                             riskColor === "red"
4426:                                 ? "bg-red-500"
4427:                                 : riskColor === "yellow"
4428:                                 ? "bg-yellow-500"
4429:                                 : "bg-green-500"
4430:                         }`}
4431:                         style={{
4432:                             width: isHighRisk
4433:                                 ? "80%"
4434:                                 : isMediumRisk
4435:                                 ? "55%"
4436:                                 : "30%",
4437:                         }}
4438:                     ></div>
4439:                 </div>
4440:             </div>
4441:
4442:             <div className="flex justify-center">
4443:                 {item.sensitivityIcon && (
4444:                     <img
4445:                         src={`\addiction/sensitivity/${item.sensitivityIcon}.png`}
4446:                         alt="Addiction Sensitivity"
4447:                         className="w-16 h-auto object-contain"

```

```

4448:         />
4449:     })
4450: </div>
4451:
4452:     { /* Helpful Context */ }
4453:     <div className="mt-2 p-1 bg-gray-50 rounded">
4454:         <p className="text-xs text-gray-600 text-center">
4455:             {isHighRisk
4456:                 ? "Extra awareness recommended"
4457:                 : isMediumRisk
4458:                 ? "Moderate caution advised"
4459:                 : "Lower genetic predisposition"}
4460:         </p>
4461:     </div>
4462: </div>
4463:     );
4464:   })}
4465: </div>
4466: </CardContent>
4467: </Card>
4468: )}
4469:
4470: { /* Sleep Analysis (only if data exists) */ }
4471: {Object.keys(sleepAnalysis).length > 0 && (
4472:   <Card className="border shadow-sm">
4473:     <CardHeader className="pb-2">
4474:       <CardTitle className="text-sm font-bold text-teal-800">
4475:         SLEEP & REST GENETIC ANALYSIS
4476:       </CardTitle>
4477:     </CardHeader>
4478:     <CardContent className="text-xs text-gray-700">
4479:       <div className="mb-3 p-2 bg-white rounded border">
4480:         <p className="font-semibold mb-1">
4481:           SLEEP GENETIC PARAMETERS
4482:         </p>
4483:         <p>
4484:           Total Sleep Factors: {Object.keys(sleepAnalysis).length}
4485:         </p>
4486:       </div>
4487:
4488:       <div className="grid grid-cols-2 gap-3">
4489:         {Object.entries(sleepAnalysis).map(([key, item]) => (
4490:           <div key={key} className="border rounded p-2 bg-white">
4491:             <div className="flex items-center gap-2 mb-1">
4492:               {item.hasImage && (
4493:                 <img
4494:                   src={`\sleep/${item.image}.png`}
4495:                   alt={item.title}
4496:                   className="w-6 h-6 object-contain"
4497:                 />
4498:               )}
4499:               <p className="font-semibold">{item.title}</p>
4500:             </div>
4501:             {item.intervention && (
4502:               <p className="text-xs">
4503:                 {item.intervention.substring(0, 80)}...
4504:               </p>
4505:             )}
4506:           </div>
4507:         )}
4508:       </div>
4509:     </CardContent>
4510:   </Card>
4511: )}
4512:
4513: { /* Allergy Analysis (only if data exists) */ }
4514: {Object.keys(allergyAnalysis).length > 0 && (
4515:   <Card className="border shadow-sm">
4516:     <CardHeader className="pb-2">
4517:       <CardTitle className="text-sm font-bold text-amber-800">
4518:         ALLERGY & SENSITIVITY GENETIC PROFILE
4519:       </CardTitle>
4520:     </CardHeader>

```



```

4521:         <CardContent className="text-xs text-gray-700">
4522:             <div className="mb-3 p-2 bg-white rounded border">
4523:                 <p className="font-semibold mb-1">
4524:                     ALLERGY GENETIC FACTORS
4525:                 </p>
4526:                 <p>
4527:                     Total Allergies Analyzed:{" "}
4528:                     {Object.keys(allergyAnalysis).length}
4529:                 </p>
4530:             </div>
4531:
4532:             <div className="grid grid-cols-2 gap-3">
4533:                 {Object.entries(allergyAnalysis).map(([key, item]) => (
4534:                     <div key={key} className="border rounded p-2 bg-white">
4535:                         <div className="flex items-center gap-2 mb-1">
4536:                             {item.hasImage && (
4537:                                 <img
4538:                                     src={`~/allergies/${item.image}.png`}
4539:                                     alt={item.title}
4540:                                     className="w-6 h-6 object-contain"
4541:                                 />
4542:                             )}
4543:                             <p className="font-semibold">{item.title}</p>
4544:                         </div>
4545:                     </div>
4546:                 )]}
4547:             </div>
4548:         </CardContent>
4549:     </Card>
4550: )}
4551:
4552: {/* Comprehensive Recommendations */}
4553: <Card className="border-2 shadow-sm">
4554:     <CardHeader className="pb-2">
4555:         <CardTitle className="text-sm font-bold text-indigo-800">
4556:             COMPREHENSIVE GENETIC RECOMMENDATIONS
4557:         </CardTitle>
4558:     </CardHeader>
4559:     <CardContent className="text-xs text-gray-700">
4560:         <div className="grid grid-cols-2 gap-3">
4561:             <div className="space-y-2">
4562:                 <p className="font-semibold text-red-700">
4563:                     IMMEDIATE PRIORITY ACTIONS
4564:                 </p>
4565:                 <ul className="list-disc list-inside space-y-1 text-xs">
4566:                     {totalLifestyleImprovements > 0 && (
4567:                         <li>
4568:                             Address {totalLifestyleImprovements} lifestyle
4569:                             genetic improvement areas immediately
4570:                         </li>
4571:                     )}
4572:                     {totalMetabolicHigh > 0 && (
4573:                         <li>
4574:                             Closely monitor and manage {totalMetabolicHigh}{" "}
4575:                             high-impact metabolic genetic factors
4576:                         </li>
4577:                     )}
4578:                     {totalDietScore < 5 && (
4579:                         <li>
4580:                             Implement comprehensive dietary modifications based
4581:                             on genetic profile
4582:                         </li>
4583:                     )}
4584:                     {Object.values(dietAnalysis).some(
4585:                         (cat) => cat.lowScores > cat.highScores
4586:                     ) && (
4587:                         <li>
4588:                             Focus intensive effort on underperforming dietary
4589:                             genetic categories
4590:                         </li>
4591:                     )}
4592:                     {Object.values(nutritionAnalysis).reduce(
4593:                         (sum, cat) => sum + cat.highImpact,

```

```

4594:         0
4595:     ) > 0 && (
4596:         <li>
4597:             Address{" "}
4598:             {Object.values(nutritionAnalysis).reduce(
4599:                 (sum, cat) => sum + cat.highImpact,
4600:                 0
4601:             )}{"}
4602:             high-impact nutritional genetic factors
4603:         </li>
4604:     )}
4605: </ul>
4606: </div>
4607:
4608: <div className="space-y-2">
4609:     <p className="font-semibold text-green-700">
4610:         GENETIC STRENGTHS TO LEVERAGE
4611:     </p>
4612:     <ul className="list-disc list-inside space-y-1 text-xs">
4613:         {totalLifestyleStrengths > 0 && (
4614:             <li>
4615:                 Maximize and build upon {totalLifestyleStrengths}{"}
4616:                 genetic lifestyle strengths
4617:             </li>
4618:         )}
4619:         {totalMetabolicLow > 0 && (
4620:             <li>
4621:                 Maintain current approach for {totalMetabolicLow}{"}
4622:                 favorable metabolic genetic traits
4623:             </li>
4624:         )}
4625:         {totalDietScore >= 6 && (
4626:             <li>
4627:                 Continue and optimize current dietary approach -
4628:                 excellent genetic compatibility
4629:             </li>
4630:         )}
4631:         {Object.values(dietAnalysis).some(
4632:             (cat) => cat.highScores > cat.lowScores
4633:         ) && (
4634:             <li>
4635:                 Further optimize high-performing dietary genetic
4636:                 categories for maximum benefit
4637:             </li>
4638:         )}
4639:         {Object.values(nutritionAnalysis).reduce(
4640:             (sum, cat) => sum + cat.lowImpact,
4641:             0
4642:         ) > 0 && (
4643:             <li>
4644:                 Leverage{" "}
4645:                 {Object.values(nutritionAnalysis).reduce(
4646:                     (sum, cat) => sum + cat.lowImpact,
4647:                     0
4648:                 )}{"}
4649:                 low-impact nutritional factors as genetic advantages
4650:             </li>
4651:         )}
4652:     </ul>
4653: </div>
4654: </div>
4655:
4656: <div className="mt-3 p-3 bg-white rounded border">
4657:     <p className="font-semibold mb-3">
4658:         GENETIC HEALTH SCORE BREAKDOWN
4659:     </p>
4660:
4661:     {/* Chart Container */}
4662:     <div className="h-64 mb-4">
4663:         <svg viewBox="0 0 600 240" className="w-full h-full">
4664:             {/* Grid lines */}
4665:             <defs>
4666:                 <pattern

```

```

4667:         id="grid"
4668:         width="60"
4669:         height="24"
4670:         patternUnits="userSpaceOnUse"
4671:     >
4672:         <path
4673:             d="M 60 0 L 0 0 0 24"
4674:             fill="none"
4675:             stroke="#e5e7eb"
4676:             strokeWidth="0.5"
4677:         />
4678:     </pattern>
4679: </defs>
4680: <rect width="600" height="240" fill="url(#grid)" />
4681:
4682: { /* Y-axis labels */}
4683: <text
4684:     x="20"
4685:     y="20"
4686:     fontSize="10"
4687:     fill="#6b7280"
4688:     textAnchor="middle"
4689: >
4690:     10
4691: </text>
4692: <text
4693:     x="20"
4694:     y="44"
4695:     fontSize="10"
4696:     fill="#6b7280"
4697:     textAnchor="middle"
4698: >
4699:     8
4700: </text>
4701: <text
4702:     x="20"
4703:     y="68"
4704:     fontSize="10"
4705:     fill="#6b7280"
4706:     textAnchor="middle"
4707: >
4708:     6
4709: </text>
4710: <text
4711:     x="20"
4712:     y="92"
4713:     fontSize="10"
4714:     fill="#6b7280"
4715:     textAnchor="middle"
4716: >
4717:     4
4718: </text>
4719: <text
4720:     x="20"
4721:     y="116"
4722:     fontSize="10"
4723:     fill="#6b7280"
4724:     textAnchor="middle"
4725: >
4726:     2
4727: </text>
4728: <text
4729:     x="20"
4730:     y="140"
4731:     fontSize="10"
4732:     fill="#6b7280"
4733:     textAnchor="middle"
4734: >
4735:     0
4736: </text>
4737:
4738: { /* Bars */}
4739: { /* Diet Compatibility */}

```

```

4740:         <rect
4741:             x="60"
4742:             y={140 - totalDietScore * 12}
4743:             width="100"
4744:             height={totalDietScore * 12}
4745:             fill={
4746:                 totalDietScore >= 6
4747:                 ? "#10b981"
4748:                 : totalDietScore >= 4
4749:                 ? "#f59e0b"
4750:                 : "#ef4444"
4751:             }
4752:             rx="4"
4753:         />
4754:         <text
4755:             x="110"
4756:             y="160"
4757:             fontSize="11"
4758:             fill="#374151"
4759:             textAnchor="middle"
4760:         >
4761:             Diet
4762:         </text>
4763:         <text
4764:             x="110"
4765:             y="175"
4766:             fontSize="11"
4767:             fill="#374151"
4768:             textAnchor="middle"
4769:         >
4770:             Compatibility
4771:         </text>
4772:         <text
4773:             x="110"
4774:             y="192"
4775:             fontSize="12"
4776:             fontWeight="bold"
4777:             fill={
4778:                 totalDietScore >= 6
4779:                 ? "#059669"
4780:                 : totalDietScore >= 4
4781:                 ? "#d97706"
4782:                 : "#dc2626"
4783:             }
4784:             textAnchor="middle"
4785:         >
4786:             {totalDietScore.toFixed(1)}
4787:         </text>
4788:
4789:         { /* Nutrition Alignment */ }
4790:         <rect
4791:             x="180"
4792:             y={140 - totalNutritionScore * 12}
4793:             width="100"
4794:             height={totalNutritionScore * 12}
4795:             fill={
4796:                 totalNutritionScore >= 6
4797:                 ? "#10b981"
4798:                 : totalNutritionScore >= 4
4799:                 ? "#f59e0b"
4800:                 : "#ef4444"
4801:             }
4802:             rx="4"
4803:         />
4804:         <text
4805:             x="230"
4806:             y="160"
4807:             fontSize="11"
4808:             fill="#374151"
4809:             textAnchor="middle"
4810:         >
4811:             Nutrition
4812:         </text>

```

```

4813:         <text
4814:             x="230"
4815:             y="175"
4816:             fontSize="11"
4817:             fill="#374151"
4818:             textAnchor="middle"
4819:         >
4820:             Alignment
4821:         </text>
4822:         <text
4823:             x="230"
4824:             y="192"
4825:             fontSize="12"
4826:             fontWeight="bold"
4827:             fill={
4828:                 totalNutritionScore >= 6
4829:                 ? "#059669"
4830:                 : totalNutritionScore >= 4
4831:                 ? "#d97706"
4832:                 : "#dc2626"
4833:             }
4834:             textAnchor="middle"
4835:         >
4836:             {totalNutritionScore.toFixed(1)}
4837:         </text>
4838:
4839:         { /* Lifestyle Balance */ }
4840:         <rect
4841:             x="300"
4842:             y={
4843:                 140 -
4844:                 (totalLifestyleStrengths > 0
4845:                 ? (totalLifestyleStrengths /
4846:                     (totalLifestyleStrengths +
4847:                         totalLifestyleImprovements)) *
4848:                     10
4849:                 : 0) *
4850:                 12
4851:             }
4852:             width="100"
4853:             height={
4854:                 (totalLifestyleStrengths > 0
4855:                 ? (totalLifestyleStrengths /
4856:                     (totalLifestyleStrengths +
4857:                         totalLifestyleImprovements)) *
4858:                     10
4859:                 : 0) * 12
4860:             }
4861:             fill={
4862:                 totalLifestyleStrengths > totalLifestyleImprovements
4863:                 ? "#10b981"
4864:                 : totalLifestyleStrengths ===
4865:                   totalLifestyleImprovements
4866:                 ? "#f59e0b"
4867:                 : "#ef4444"
4868:             }
4869:             rx="4"
4870:         />
4871:         <text
4872:             x="350"
4873:             y="160"
4874:             fontSize="11"
4875:             fill="#374151"
4876:             textAnchor="middle"
4877:         >
4878:             Lifestyle
4879:         </text>
4880:         <text
4881:             x="350"
4882:             y="175"
4883:             fontSize="11"
4884:             fill="#374151"
4885:             textAnchor="middle"

```

```

4886: >
4887:     Balance
4888: </text>
4889: <text
4890:     x="350"
4891:     y="192"
4892:     fontSize="12"
4893:     fontWeight="bold"
4894:     fill={
4895:         totalLifestyleStrengths > totalLifestyleImprovements
4896:         ? "#059669"
4897:         : totalLifestyleStrengths ===
4898:           totalLifestyleImprovements
4899:         ? "#d97706"
4900:         : "#dc2626"
4901:     }
4902:     textAnchor="middle"
4903: >
4904:     {totalLifestyleStrengths > 0
4905:       ? (
4906:         (totalLifestyleStrengths /
4907:          (totalLifestyleStrengths +
4908:           totalLifestyleImprovements)) *
4909:         10
4910:       ).toFixed(1)
4911:       : "0.0"}
4912: </text>
4913:
4914: {/* Metabolic Risk - inverted scale (lower is better) */}
4915: <rect
4916:     x="420"
4917:     y={
4918:         140 -
4919:         (totalMetabolicHigh === 0
4920:          ? 10
4921:          : totalMetabolicHigh <= 2
4922:            ? 6
4923:            : 3) *
4924:         12
4925:     }
4926:     width="100"
4927:     height={
4928:         (totalMetabolicHigh === 0
4929:          ? 10
4930:          : totalMetabolicHigh <= 2
4931:            ? 6
4932:            : 3) * 12
4933:     }
4934:     fill={
4935:         totalMetabolicHigh === 0
4936:         ? "#10b981"
4937:         : totalMetabolicHigh <= 2
4938:         ? "#f59e0b"
4939:         : "#ef4444"
4940:     }
4941:     rx="4"
4942: />
4943: <text
4944:     x="470"
4945:     y="160"
4946:     fontSize="11"
4947:     fill="#374151"
4948:     textAnchor="middle"
4949: >
4950:     Metabolic
4951: </text>
4952: <text
4953:     x="470"
4954:     y="175"
4955:     fontSize="11"
4956:     fill="#374151"
4957:     textAnchor="middle"
4958: >

```

```

4959:         Risk
4960:     </text>
4961:     <text
4962:         x="470"
4963:         y="192"
4964:         fontSize="12"
4965:         fontWeight="bold"
4966:         fill={
4967:             totalMetabolicHigh === 0
4968:               ? "#059669"
4969:             : totalMetabolicHigh <= 2
4970:               ? "#d97706"
4971:             : "#dc2626"
4972:         }
4973:         textAnchor="middle"
4974:     >
4975:         {totalMetabolicHigh === 0
4976:           ? "Low"
4977:           : totalMetabolicHigh <= 2
4978:           ? "Med"
4979:           : "High"}
4980:     </text>
4981:
4982:     { /* Reference lines */ }
4983:     <line
4984:         x1="45"
4985:         y1="68"
4986:         x2="540"
4987:         y2="68"
4988:         stroke="#10b981"
4989:         strokeWidth="1"
4990:         strokeDasharray="3,3"
4991:         opacity="0.6"
4992:     />
4993:     <text x="545" y="72" fontSize="9" fill="#10b981">
4994:         Good (6+)
4995:     </text>
4996:
4997:     <line
4998:         x1="45"
4999:         y1="92"
5000:         x2="540"
5001:         y2="92"
5002:         stroke="#f59e0b"
5003:         strokeWidth="1"
5004:         strokeDasharray="3,3"
5005:         opacity="0.6"
5006:     />
5007:     <text x="545" y="96" fontSize="9" fill="#f59e0b">
5008:         Fair (4+)
5009:     </text>
5010: </svg>
5011: </div>
5012:
5013: { /* Legend */ }
5014: <div className="flex justify-center space-x-4 text-xs">
5015:     <div className="flex items-center">
5016:         <div className="w-3 h-3 bg-green-500 rounded mr-1"></div>
5017:         <span>Excellent/Low Risk</span>
5018:     </div>
5019:     <div className="flex items-center">
5020:         <div className="w-3 h-3 bg-yellow-500 rounded mr-1"></div>
5021:         <span>Good/Medium Risk</span>
5022:     </div>
5023:     <div className="flex items-center">
5024:         <div className="w-3 h-3 bg-red-500 rounded mr-1"></div>
5025:         <span>Needs Attention/High Risk</span>
5026:     </div>
5027: </div>
5028: </div>
5029: </CardContent>
5030: </Card>
5031:

```

```

5032:      {/* Expert Summaries with Enhanced Detail */}
5033:      <div className="grid grid-cols-2 gap-3">
5034:        <DataCard title="Expert Nutrigenomics Analysis">
5035:          <div className="space-y-2">
5036:            <div className="p-2 bg-blue-50 rounded">
5037:              <p className="font-semibold text-xs mb-1">
5038:                Professional Assessment:
5039:              </p>
5040:              <p className="text-justify break-words text-xs">
5041:                {summaries.nutrigenomicsSummary}
5042:              </p>
5043:            </div>
5044:            <div className="text-xs">
5045:              <p className="font-semibold">Key Metrics:</p>
5046:              <ul className="list-disc list-inside space-y-0">
5047:                <li>
5048:                  Nutritional genetic factors:{" "}
5049:                  {Object.values(nutritionAnalysis).reduce(
5050:                    (sum, cat) => sum + cat.total,
5051:                    0
5052:                  )}
5053:                </li>
5054:                <li>
5055:                  Average compatibility score:{" "}
5056:                  {totalNutritionScore.toFixed(1)}/10
5057:                </li>
5058:                <li>
5059:                  High-impact factors requiring attention:{" "}
5060:                  {Object.values(nutritionAnalysis).reduce(
5061:                    (sum, cat) => sum + cat.highImpact,
5062:                    0
5063:                  )}
5064:                </li>
5065:              </ul>
5066:            </div>
5067:          </div>
5068:        </DataCard>
5069:
5070:        <DataCard title="Expert Exercise Genomics Analysis">
5071:          <div className="space-y-2">
5072:            <div className="p-2 bg-orange-50 rounded">
5073:              <p className="font-semibold text-xs mb-1">
5074:                Professional Assessment:
5075:              </p>
5076:              <p className="text-justify break-words text-xs">
5077:                {summaries.exerciseGenomicsSummary}
5078:              </p>
5079:            </div>
5080:            <div className="text-xs">
5081:              <p className="font-semibold">Key Metrics:</p>
5082:              <ul className="list-disc list-inside space-y-0">
5083:                {Object.keys(sportsAnalysis).length > 0 && (
5084:                  <li>
5085:                    Fitness genetic traits analyzed:{" "}
5086:                    {Object.values(sportsAnalysis).reduce(
5087:                      (sum, section) => sum + section.totalTraits,
5088:                      0
5089:                    )}
5090:                  </li>
5091:                )}
5092:                <li>
5093:                  Lifestyle genetic strengths: {totalLifestyleStrengths}
5094:                </li>
5095:                <li>
5096:                  Areas for improvement: {totalLifestyleImprovements}
5097:                </li>
5098:                <li>
5099:                  Metabolic genetic considerations:{" "}
5100:                  {totalMetabolicHigh +
5101:                    totalMetabolicModerate +
5102:                    totalMetabolicLow}{"}"}
5103:                  genes
5104:                </li>

```



```

5105:         </ul>
5106:     </div>
5107: </div>
5108: </DataCard>
5109: </div>
5110: </div>
5111: );
5112: })(){}
5113:
5114: <Separator className="my-4" />
5115:
5116: { /* Final Thank You Page */ }
5117: <div className="h-[270mm] bg-gradient-to-br from-blue-100 rounded-lg via-purple-100 to-pink-100 pa
5118: { /* Main content section */ }
5119: <div className="flex flex-1 pt-16">
5120: { /* Left - Thank you text */ }
5121: <div className="flex-1 flex flex-col justify-start px-8">
5122:   <h1 className="text-5xl tracking-wide mb-8 text-gray-800">
5123:     Thank You...
5124:   </h1>
5125:   <div className="mb-6">
5126:     <p className="text-lg font-medium mb-2 text-gray-700">To,</p>
5127:     <p className="text-xl font-semibold text-gray-900">
5128:       {patientInfo.name}
5129:     </p>
5130:   </div>
5131:   <p className="text-base leading-relaxed mb-8 text-gray-800 font-light max-w-lg text-justify">
5132:     Thank you for taking the time to review our nutrigenomics
5133:     report. We appreciate the opportunity to conduct this analysis
5134:     for you. For further inquiries or personalized guidance, please
5135:     feel free to connect with us using the contact details provided
5136:     below.
5137:   </p>
5138:   <div className="text-sm leading-relaxed text-gray-700 space-y-4">
5139:     <div>
5140:       <p className="font-bold text-gray-900 mb-1">
5141:         Molsys Private Limited
5142:       </p>
5143:       <p className="font-light leading-relaxed">
5144:         Regd. Office: 1st Floor, #524, 13th Main, 23rd Cross,
5145:         Judicial Layout, Yelahanka
5146:       </p>
5147:       <p className="font-light leading-relaxed">
5148:         { " " }
5149:         Bengaluru, Karnataka, Pin: 560065
5150:       </p>
5151:       <a
5152:         href="https://www.molsys.in"
5153:         className="font-light "
5154:         target="_blank"
5155:         rel="noopener noreferrer"
5156:       >
5157:         Website: https://www.molsys.in
5158:       </a>
5159:
5160:       <br />
5161:
5162:       <a href="mailto:wellness@molsys.in" className="font-light ">
5163:         Email: wellness@molsys.in
5164:       </a>
5165:
5166:       <p className="font-light">
5167:         Contact no: 8271502582 / 8073193805
5168:       </p>
5169:     </div>
5170:
5171:     <div className="pt-5">
5172:       <p className="font-bold text-gray-900 mb-1">
5173:         Molsys Private Limited (R&D)
5174:       </p>
5175:       <p className="font-light leading-relaxed">
5176:         Yenepoya Technology Incubator, Yenepoya (deemed-to-be)
5177:         University,

```

```

5178:         </p>
5179:         <p className="font-light leading-relaxed">
5180:             Deralakatte, Ullal, DK Pin: 575020
5181:         </p>
5182:     </div>
5183: </div>
5184: </div>
5185: </div>
5186: { /* Bottom logos */}
5187: <div className="absolute bottom-4 left-4">
5188:     
5193: </div>
5194: <div className="absolute bottom-4 right-4">
5195:     
5200: </div>
5201: </div>
5202:
5203: { /* <footer className="text-center mt-4 py-3 text-gray-600 text-xs border-t border-gray-200">
5204:     <div className="space-y-1">
5205:         <p>
5206:             &copy; {new Date().getFullYear()} {settings.companyName}. All
5207:             rights reserved.
5208:         </p>
5209:         <p className="break-words">
5210:             Report Authentication: {patientInfo.reportAuth}
5211:         </p>
5212:     </div>
5213: </footer> */}
5214: </div>
5215: </>
5216: );
5217: }

```

## ■ File: components\theme-provider.tsx

```

=====
1: "use client"
2: import { ThemeProvider as NextThemesProvider } from "next-themes"
3: import type { ThemeProviderProps } from "next-themes"
4:
5: export function ThemeProvider({ children, ...props }: ThemeProviderProps) {
6:     return <NextThemesProvider {...props}>{children}</NextThemesProvider>
7: }

```

## ■ File: components\ui\accordion.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import * as AccordionPrimitive from "@radix-ui/react-accordion"
5: import { ChevronDown } from "lucide-react"
6:
7: import { cn } from "@lib/utils"
8:
9: const Accordion = AccordionPrimitive.Root
10:
11: const AccordionItem = React.forwardRef<
12:     React.ElementRef<typeof AccordionPrimitive.Item>,
13:     React.ComponentPropsWithoutRef<typeof AccordionPrimitive.Item>
14: >(({ className, ...props }, ref) => (
15:     <AccordionPrimitive.Item
16:         ref={ref}

```

```

17:     className={cn("border-b", className)}
18:     {...props}
19:   />
20: ))
21: AccordionItem.displayName = "AccordionItem"
22:
23: const AccordionTrigger = React.forwardRef<
24:   React.ElementRef<typeof AccordionPrimitive.Trigger>,
25:   React.ComponentPropsWithoutRef<typeof AccordionPrimitive.Trigger>
26: >(({ className, children, ...props }, ref) => (
27:   <AccordionPrimitive.Header className="flex">
28:     <AccordionPrimitive.Trigger
29:       ref={ref}
30:       className={cn(
31:         "flex flex-1 items-center justify-between py-4 font-medium transition-all hover:underline [&[data-s
32:         className
33:       )}
34:       {...props}
35:     >
36:       {children}
37:     <ChevronDown className="h-4 w-4 shrink-0 transition-transform duration-200" />
38:   </AccordionPrimitive.Trigger>
39: </AccordionPrimitive.Header>
40: ))
41: AccordionTrigger.displayName = AccordionPrimitive.Trigger.displayName
42:
43: const AccordionContent = React.forwardRef<
44:   React.ElementRef<typeof AccordionPrimitive.Content>,
45:   React.ComponentPropsWithoutRef<typeof AccordionPrimitive.Content>
46: >(({ className, children, ...props }, ref) => (
47:   <AccordionPrimitive.Content
48:     ref={ref}
49:     className="overflow-hidden text-sm transition-all data-[state=closed]:animate-accordion-up data-[state=
50:     {...props}
51:   >
52:     <div className={cn("pb-4 pt-0", className)}>{children}</div>
53:   </AccordionPrimitive.Content>
54: ))
55:
56: AccordionContent.displayName = AccordionPrimitive.Content.displayName
57:
58: export { Accordion, AccordionItem, AccordionTrigger, AccordionContent }

```

## ■ File: components/ui/alert-dialog.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import * as AlertDialogPrimitive from "@radix-ui/react-alert-dialog"
5:
6: import { cn } from "@lib/Utils"
7: import { buttonVariants } from "@components/ui/button"
8:
9: const AlertDialog = AlertDialogPrimitive.Root
10:
11: const AlertDialogTrigger = AlertDialogPrimitive.Trigger
12:
13: const AlertDialogPortal = AlertDialogPrimitive.Portal
14:
15: const AlertDialogOverlay = React.forwardRef<
16:   React.ElementRef<typeof AlertDialogPrimitive.Overlay>,
17:   React.ComponentPropsWithoutRef<typeof AlertDialogPrimitive.Overlay>
18: >(({ className, ...props }, ref) => (
19:   <AlertDialogPrimitive.Overlay
20:     className={cn(
21:       "fixed inset-0 z-50 bg-black/80 data-[state=open]:animate-in data-[state=closed]:animate-out data-[s
22:     className
23:   )}
24:   {...props}
25:   ref={ref}
26: />

```

```

27: ))
28: AlertDialogOverlay.displayName = AlertDialogPrimitive.Overlay.displayName
29:
30: const AlertDialogContent = React.forwardRef<
31:   React.ElementRef<typeof AlertDialogPrimitive.Content>,
32:   React.ComponentPropsWithoutRef<typeof AlertDialogPrimitive.Content>
33: >(({ className, ...props }, ref) => (
34:   <AlertDialogPortal>
35:     <AlertDialogOverlay />
36:     <AlertDialogPrimitive.Content
37:       ref={ref}
38:       className={cn(
39:         "fixed left-[50%] top-[50%] z-50 grid w-full max-w-lg translate-x-[-50%] translate-y-[-50%] gap-4 b
40:         className
41:       )}
42:       {...props}
43:     />
44:   </AlertDialogPortal>
45: ))
46: AlertDialogContent.displayName = AlertDialogPrimitive.Content.displayName
47:
48: const AlertDialogHeader = ({
49:   className,
50:   ...props
51: }: React.HTMLAttributes<HTMLDivElement>) => (
52:   <div
53:     className={cn(
54:       "flex flex-col space-y-2 text-center sm:text-left",
55:       className
56:     )}
57:     {...props}
58:   />
59: )
60: AlertDialogHeader.displayName = "AlertDialogHeader"
61:
62: const AlertDialogFooter = ({
63:   className,
64:   ...props
65: }: React.HTMLAttributes<HTMLDivElement>) => (
66:   <div
67:     className={cn(
68:       "flex flex-col-reverse sm:flex-row sm:justify-end sm:space-x-2",
69:       className
70:     )}
71:     {...props}
72:   />
73: )
74: AlertDialogFooter.displayName = "AlertDialogFooter"
75:
76: const AlertDialogTitle = React.forwardRef<
77:   React.ElementRef<typeof AlertDialogPrimitive.Title>,
78:   React.ComponentPropsWithoutRef<typeof AlertDialogPrimitive.Title>
79: >(({ className, ...props }, ref) => (
80:   <AlertDialogPrimitive.Title
81:     ref={ref}
82:     className={cn("text-lg font-semibold", className)}
83:     {...props}
84:   />
85: ))
86: AlertDialogTitle.displayName = AlertDialogPrimitive.Title.displayName
87:
88: const AlertDialogDescription = React.forwardRef<
89:   React.ElementRef<typeof AlertDialogPrimitive.Description>,
90:   React.ComponentPropsWithoutRef<typeof AlertDialogPrimitive.Description>
91: >(({ className, ...props }, ref) => (
92:   <AlertDialogPrimitive.Description
93:     ref={ref}
94:     className={cn("text-sm text-muted-foreground", className)}
95:     {...props}
96:   />
97: ))
98: AlertDialogDescription.displayName =
99:   AlertDialogPrimitive.Description.displayName

```

```

100:
101: const AlertDialogAction = React.forwardRef<
102:   React.ElementRef<typeof AlertDialogPrimitive.Action>,
103:   React.ComponentPropsWithoutRef<typeof AlertDialogPrimitive.Action>
104: >(({ className, ...props }, ref) => (
105:   <AlertDialogPrimitive.Action
106:     ref={ref}
107:     className={cn(buttonVariants(), className)}
108:     {...props}
109:   />
110: ))
111: AlertDialogAction.displayName = AlertDialogPrimitive.Action.displayName
112:
113: const AlertDialogCancel = React.forwardRef<
114:   React.ElementRef<typeof AlertDialogPrimitive.Cancel>,
115:   React.ComponentPropsWithoutRef<typeof AlertDialogPrimitive.Cancel>
116: >(({ className, ...props }, ref) => (
117:   <AlertDialogPrimitive.Cancel
118:     ref={ref}
119:     className={cn(
120:       buttonVariants({ variant: "outline" }),
121:       "mt-2 sm:mt-0",
122:       className
123:     )}
124:     {...props}
125:   />
126: ))
127: AlertDialogCancel.displayName = AlertDialogPrimitive.Cancel.displayName
128:
129: export {
130:   AlertDialog,
131:   AlertDialogPortal,
132:   AlertDialogOverlay,
133:   AlertDialogTrigger,
134:   AlertDialogContent,
135:   AlertDialogHeader,
136:   AlertDialogFooter,
137:   AlertDialogTitle,
138:   AlertDialogDescription,
139:   AlertDialogAction,
140:   AlertDialogCancel,
141: }

```

## ■ File: components/ui/alert.tsx

```

=====
1: import * as React from "react"
2: import { cva, type VariantProps } from "class-variance-authority"
3:
4: import { cn } from "@/lib/utils"
5:
6: const alertVariants = cva(
7:   "relative w-full rounded-lg border p-4 [&svg~*]:pl-7 [&svg+div]:translate-y-[-3px] [&svg]:absolute [&
8:   {
9:     variants: {
10:       variant: {
11:         default: "bg-background text-foreground",
12:         destructive:
13:           "border-destructive/50 text-destructive dark:border-destructive [&svg]:text-destructive",
14:       },
15:     },
16:     defaultVariants: {
17:       variant: "default",
18:     },
19:   }
20: )
21:
22: const Alert = React.forwardRef<
23:   HTMLDivElement,
24:   React.HTMLAttributes<HTMLDivElement> & VariantProps<typeof alertVariants>
25: >(({ className, variant, ...props }, ref) => (
26:   <div

```

```

27:     ref={ref}
28:     role="alert"
29:     className={cn(alertVariants({ variant })), className)}
30:     {...props}
31:   />
32: ))
33: Alert.displayName = "Alert"
34:
35: const AlertTitle = React.forwardRef<
36:   HTMLParagraphElement,
37:   React.HTMLAttributes<HTMLHeadingElement>
38: >(({ className, ...props }, ref) => (
39:   <h5
40:     ref={ref}
41:     className={cn("mb-1 font-medium leading-none tracking-tight", className)}
42:     {...props}
43:   />
44: ))
45: AlertTitle.displayName = "AlertTitle"
46:
47: const AlertDescription = React.forwardRef<
48:   HTMLParagraphElement,
49:   React.HTMLAttributes<HTMLParagraphElement>
50: >(({ className, ...props }, ref) => (
51:   <div
52:     ref={ref}
53:     className={cn("text-sm [&_p]:leading-relaxed", className)}
54:     {...props}
55:   />
56: ))
57: AlertDescription.displayName = "AlertDescription"
58:
59: export { Alert, AlertTitle, AlertDescription }

```

## ■ File: components/ui/aspect-ratio.tsx

```

1: "use client"
2:
3: import * as AspectRatioPrimitive from "@radix-ui/react-aspect-ratio"
4:
5: const AspectRatio = AspectRatioPrimitive.Root
6:
7: export { AspectRatio }

```

## ■ File: components/ui/avatar.tsx

```

1: "use client"
2:
3: import * as React from "react"
4: import * as AvatarPrimitive from "@radix-ui/react-avatar"
5:
6: import { cn } from "@/lib/utils"
7:
8: const Avatar = React.forwardRef<
9:   React.ElementRef<typeof AvatarPrimitive.Root>,
10:   React.ComponentPropsWithoutRef<typeof AvatarPrimitive.Root>
11: >(({ className, ...props }, ref) => (
12:   <AvatarPrimitive.Root
13:     ref={ref}
14:     className={cn(
15:       "relative flex h-10 w-10 shrink-0 overflow-hidden rounded-full",
16:       className
17:     )}
18:     {...props}
19:   />
20: ))
21: Avatar.displayName = AvatarPrimitive.Root.displayName
22:
23: const AvatarImage = React.forwardRef<

```

```

24:   React.ElementRef<typeof AvatarPrimitive.Image>,
25:   React.ComponentPropsWithoutRef<typeof AvatarPrimitive.Image>
26: >(({ className, ...props }, ref) => (
27:   <AvatarPrimitive.Image
28:     ref={ref}
29:     className={cn("aspect-square h-full w-full", className)}
30:     {...props}
31:   />
32: ))
33: AvatarImage.displayName = AvatarPrimitive.Image.displayName
34:
35: const AvatarFallback = React.forwardRef<
36:   React.ElementRef<typeof AvatarPrimitive.Fallback>,
37:   React.ComponentPropsWithoutRef<typeof AvatarPrimitive.Fallback>
38: >(({ className, ...props }, ref) => (
39:   <AvatarPrimitive.Fallback
40:     ref={ref}
41:     className={cn(
42:       "flex h-full w-full items-center justify-center rounded-full bg-muted",
43:       className
44:     )}
45:     {...props}
46:   />
47: ))
48: AvatarFallback.displayName = AvatarPrimitive.Fallback.displayName
49:
50: export { Avatar, AvatarImage, AvatarFallback }

```

## ■ File: components/ui/badge.tsx

```

1: import * as React from "react"
2: import { cva, type VariantProps } from "class-variance-authority"
3:
4: import { cn } from "@/lib/utils"
5:
6: const badgeVariants = cva(
7:   "inline-flex items-center rounded-full border px-2.5 py-0.5 text-xs font-semibold transition-colors focus:
8:   {
9:     variants: {
10:       variant: {
11:         default:
12:           "border-transparent bg-primary text-primary-foreground hover:bg-primary/80",
13:         secondary:
14:           "border-transparent bg-secondary text-secondary-foreground hover:bg-secondary/80",
15:         destructive:
16:           "border-transparent bg-destructive text-destructive-foreground hover:bg-destructive/80",
17:         outline: "text-foreground",
18:       },
19:     },
20:     defaultVariants: {
21:       variant: "default",
22:     },
23:   }
24: )
25:
26: export interface BadgeProps
27:   extends React.HTMLAttributes<HTMLDivElement>,
28:   VariantProps<typeof badgeVariants> {}
29:
30: function Badge({ className, variant, ...props }: BadgeProps) {
31:   return (
32:     <div className={cn(badgeVariants({ variant }), className)} {...props} />
33:   )
34: }
35:
36: export { Badge, badgeVariants }

```

## ■ File: components/ui/breadcrumb.tsx

```
=====
1: import * as React from "react"
2: import { Slot } from "@radix-ui/react-slot"
3: import { ChevronRight, MoreHorizontal } from "lucide-react"
4:
5: import { cn } from "@lib/Utils"
6:
7: const Breadcrumb = React.forwardRef<
8:   HTMLDivElement,
9:   React.ComponentPropsWithoutRef<"nav"> & {
10:     separator?: React.ReactNode
11:   }
12: >(({ ...props }, ref) => <nav ref={ref} aria-label="breadcrumb" {...props} />)
13: Breadcrumb.displayName = "Breadcrumb"
14:
15: const BreadcrumbList = React.forwardRef<
16:   HTMLListElement,
17:   React.ComponentPropsWithoutRef<"ol">
18: >(({ className, ...props }, ref) => (
19:   <ol
20:     ref={ref}
21:     className={cn(
22:       "flex flex-wrap items-center gap-1.5 break-words text-sm text-muted-foreground sm:gap-2.5",
23:       className
24:     )}
25:     {...props}
26:   />
27: ))
28: BreadcrumbList.displayName = "BreadcrumbList"
29:
30: const BreadcrumbItem = React.forwardRef<
31:   HTMLLIElement,
32:   React.ComponentPropsWithoutRef<"li">
33: >(({ className, ...props }, ref) => (
34:   <li
35:     ref={ref}
36:     className={cn("inline-flex items-center gap-1.5", className)}
37:     {...props}
38:   />
39: ))
40: BreadcrumbItem.displayName = "BreadcrumbItem"
41:
42: const BreadcrumbLink = React.forwardRef<
43:   HTMLAnchorElement,
44:   React.ComponentPropsWithoutRef<"a"> & {
45:     asChild?: boolean
46:   }
47: >(({ asChild, className, ...props }, ref) => {
48:   const Comp = asChild ? Slot : "a"
49:
50:   return (
51:     <Comp
52:       ref={ref}
53:       className={cn("transition-colors hover:text-foreground", className)}
54:       {...props}
55:     />
56:   )
57: })
58: BreadcrumbLink.displayName = "BreadcrumbLink"
59:
60: const BreadcrumbPage = React.forwardRef<
61:   HTMLSpanElement,
62:   React.ComponentPropsWithoutRef<"span">
63: >(({ className, ...props }, ref) => (
64:   <span
65:     ref={ref}
66:     role="link"
67:     aria-disabled="true"
68:     aria-current="page"
69:     className={cn("font-normal text-foreground", className)}
70:     {...props}
71:   />
=====
```



```

72: ))
73: BreadcrumbPage.displayName = "BreadcrumbPage"
74:
75: const BreadcrumbSeparator = ({
76:   children,
77:   className,
78:   ...props
79: }: React.ComponentProps<"li">) => (
80:   <li
81:     role="presentation"
82:     aria-hidden="true"
83:     className={cn("[&svg]:w-3.5 [&svg]:h-3.5", className)}
84:     {...props}
85:   >
86:     {children ?? <ChevronRight />}
87:   </li>
88: )
89: BreadcrumbSeparator.displayName = "BreadcrumbSeparator"
90:
91: const BreadcrumbEllipsis = ({
92:   className,
93:   ...props
94: }: React.ComponentProps<"span">) => (
95:   <span
96:     role="presentation"
97:     aria-hidden="true"
98:     className={cn("flex h-9 w-9 items-center justify-center", className)}
99:     {...props}
100:   >
101:     <MoreHorizontal className="h-4 w-4" />
102:     <span className="sr-only">More</span>
103:   </span>
104: )
105: BreadcrumbEllipsis.displayName = "BreadcrumbElipssis"
106:
107: export {
108:   Breadcrumb,
109:   BreadcrumbList,
110:   BreadcrumbItem,
111:   BreadcrumbLink,
112:   BreadcrumbPage,
113:   BreadcrumbSeparator,
114:   BreadcrumbEllipsis,
115: }

```

## ■ File: components\ui\button.tsx

```

1: import * as React from "react";
2: import { Slot } from "@radix-ui/react-slot";
3: import { cva, type VariantProps } from "class-variance-authority";
4:
5: import { cn } from "@lib/Utils";
6:
7: const buttonVariants = cva(
8:   "inline-flex items-center justify-center gap-2 whitespace-nowrap rounded-md text-sm font-medium ring-off",
9:   {
10:     variants: {
11:       variant: {
12:         default:
13:           "bg-gradient-to-r from-blue-400 to-purple-500 text-white hover:from-blue-500 hover:to-purple-600",
14:         destructive:
15:           "bg-gradient-to-r from-red-400 to-pink-500 text-white hover:from-red-500 hover:to-pink-600 shadow",
16:         outline:
17:           "border border-gradient bg-white text-gray-700 hover:bg-gradient-to-r hover:from-blue-50 hover:to-purple-600",
18:         secondary:
19:           "bg-gradient-to-r from-gray-200 to-gray-300 text-gray-800 hover:from-gray-300 hover:to-gray-400",
20:         ghost:
21:           "bg-gradient-to-r from-transparent to-transparent hover:from-blue-50 hover:to-purple-50 text-gray-700",
22:         link: "bg-gradient-to-r from-blue-600 to-purple-600 bg-clip-text text-transparent underline-offset-4",
23:         success:
24:           "bg-gradient-to-r from-green-400 to-emerald-500 text-white hover:from-green-500 hover:to-emerald-600",

```

```

25:         warning:
26:           "bg-gradient-to-r from-yellow-400 to-orange-500 text-white hover:from-yellow-500 hover:to-orange-
27:         info: "bg-gradient-to-r from-cyan-400 to-blue-500 text-white hover:from-cyan-500 hover:to-blue-600
28:       },
29:       size: {
30:         default: "h-10 px-4 py-2",
31:         sm: "h-9 rounded-md px-3",
32:         lg: "h-11 rounded-md px-8",
33:         icon: "h-10 w-10",
34:       },
35:     },
36:     defaultVariants: {
37:       variant: "default",
38:       size: "default",
39:     },
40:   }
41: );
42:
43: export interface ButtonProps
44:   extends React.ButtonHTMLAttributes<HTMLButtonElement>,
45:     VariantProps<typeof buttonVariants> {
46:   asChild?: boolean;
47: }
48:
49: const Button = React.forwardRef<HTMLButtonElement, ButtonProps>(
50:   ({ className, variant, size, asChild = false, ...props }, ref) => {
51:     const Comp = asChild ? Slot : "button";
52:     return (
53:       <Comp
54:         className={cn(buttonVariants({ variant, size, className }))}
55:         ref={ref}
56:         {...props}
57:       />
58:     );
59:   }
60: );
61: Button.displayName = "Button";
62:
63: export { Button, buttonVariants };

```

## ■ File: components/ui/calendar.tsx

```

1: "use client"
2:
3: import * as React from "react"
4: import { ChevronLeft, ChevronRight } from "lucide-react"
5: import { DayPicker } from "react-day-picker"
6:
7: import { cn } from "@/lib/utlis"
8: import { buttonVariants } from "@/components/ui/button"
9:
10: export type CalendarProps = React.ComponentProps<typeof DayPicker>
11:
12: function Calendar({
13:   className,
14:   classNames,
15:   showOutsideDays = true,
16:   ...props
17: }: CalendarProps) {
18:   return (
19:     <DayPicker
20:       showOutsideDays={showOutsideDays}
21:       className={cn("p-3", className)}
22:       classNames={{
23:         months: "flex flex-col sm:flex-row space-y-4 sm:space-x-4 sm:space-y-0",
24:         month: "space-y-4",
25:         caption: "flex justify-center pt-1 relative items-center",
26:         caption_label: "text-sm font-medium",
27:         nav: "space-x-1 flex items-center",
28:         nav_button: cn(
29:           buttonVariants({ variant: "outline" }),

```

```

30:         "h-7 w-7 bg-transparent p-0 opacity-50 hover:opacity-100"
31:     ),
32:     nav_button_previous: "absolute left-1",
33:     nav_button_next: "absolute right-1",
34:     table: "w-full border-collapse space-y-1",
35:     head_row: "flex",
36:     head_cell:
37:         "text-muted-foreground rounded-md w-9 font-normal text-[0.8rem]",
38:     row: "flex w-full mt-2",
39:     cell: "h-9 w-9 text-center text-sm p-0 relative [&:has([aria-selected].day-range-end)]:rounded-r-md",
40:     day: cn(
41:         buttonVariants({ variant: "ghost" }),
42:         "h-9 w-9 p-0 font-normal aria-selected:opacity-100"
43:     ),
44:     day_range_end: "day-range-end",
45:     day_selected:
46:         "bg-primary text-primary-foreground hover:bg-primary hover:text-primary-foreground focus:bg-primary",
47:     day_today: "bg-accent text-accent-foreground",
48:     day_outside:
49:         "day-outside text-muted-foreground aria-selected:bg-accent/50 aria-selected:text-muted-foreground",
50:     day_disabled: "text-muted-foreground opacity-50",
51:     day_range_middle:
52:         "aria-selected:bg-accent aria-selected:text-accent-foreground",
53:     day_hidden: "invisible",
54:     ...classNames,
55:   }}
56:   components={{
57:     IconLeft: ({ ...props }) => <ChevronLeft className="h-4 w-4" />,
58:     IconRight: ({ ...props }) => <ChevronRight className="h-4 w-4" />,
59:   }}
60:   {...props}
61: />
62: )
63: }
64: Calendar.displayName = "Calendar"
65:
66: export { Calendar }

```

## ■ File: components\ui\card.tsx

```

1: import * as React from "react"
2:
3: import { cn } from "@/lib/utils"
4:
5: const Card = React.forwardRef<
6:   HTMLDivElement,
7:   React.HTMLAttributes<HTMLDivElement>
8: >(({ className, ...props }, ref) => (
9:   <div
10:     ref={ref}
11:     className={cn(
12:       "rounded-lg border bg-card text-card-foreground shadow-sm",
13:       className
14:     )}
15:     {...props}
16:   />
17: ))
18: Card.displayName = "Card"
19:
20: const CardHeader = React.forwardRef<
21:   HTMLDivElement,
22:   React.HTMLAttributes<HTMLDivElement>
23: >(({ className, ...props }, ref) => (
24:   <div
25:     ref={ref}
26:     className={cn("flex flex-col space-y-1.5 p-6", className)}
27:     {...props}
28:   />
29: ))
30: CardHeader.displayName = "CardHeader"
31:

```

```

32: const CardTitle = React.forwardRef<
33:   HTMLDivElement,
34:   React.HTMLAttributes<HTMLDivElement>
35: >(({ className, ...props }, ref) => (
36:   <div
37:     ref={ref}
38:     className={cn(
39:       "text-2xl font-semibold leading-none tracking-tight",
40:       className
41:     )}
42:     {...props}
43:   />
44: ))
45: CardTitle.displayName = "CardTitle"
46:
47: const CardDescription = React.forwardRef<
48:   HTMLDivElement,
49:   React.HTMLAttributes<HTMLDivElement>
50: >(({ className, ...props }, ref) => (
51:   <div
52:     ref={ref}
53:     className={cn("text-sm text-muted-foreground", className)}
54:     {...props}
55:   />
56: ))
57: CardDescription.displayName = "CardDescription"
58:
59: const CardContent = React.forwardRef<
60:   HTMLDivElement,
61:   React.HTMLAttributes<HTMLDivElement>
62: >(({ className, ...props }, ref) => (
63:   <div ref={ref} className={cn("p-6 pt-0", className)} {...props} />
64: ))
65: CardContent.displayName = "CardContent"
66:
67: const CardFooter = React.forwardRef<
68:   HTMLDivElement,
69:   React.HTMLAttributes<HTMLDivElement>
70: >(({ className, ...props }, ref) => (
71:   <div
72:     ref={ref}
73:     className={cn("flex items-center p-6 pt-0", className)}
74:     {...props}
75:   />
76: ))
77: CardFooter.displayName = "CardFooter"
78:
79: export { Card, CardHeader, CardFooter, CardTitle, CardDescription, CardContent }

```

## ■ File: components/ui/carousel.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import useEmblaCarousel, {
5:   type UseEmblaCarouselType,
6: } from "embla-carousel-react"
7: import { ArrowLeft, ArrowRight } from "lucide-react"
8:
9: import { cn } from "@/lib/utils"
10: import { Button } from "@/components/ui/button"
11:
12: type CarouselApi = UseEmblaCarouselType[1]
13: type UseCarouselParameters = Parameters<typeof useEmblaCarousel>
14: type CarouselOptions = UseCarouselParameters[0]
15: type CarouselPlugin = UseCarouselParameters[1]
16:
17: type CarouselProps = {
18:   opts?: CarouselOptions
19:   plugins?: CarouselPlugin
20:   orientation?: "horizontal" | "vertical"

```

```

21:   setApi?: (api: CarouselApi) => void
22: }
23:
24: type CarouselContextProps = {
25:   carouselRef: ReturnType<typeof useEmblaCarousel>[0]
26:   api: ReturnType<typeof useEmblaCarousel>[1]
27:   scrollPrev: () => void
28:   scrollNext: () => void
29:   canScrollPrev: boolean
30:   canScrollNext: boolean
31: } & CarouselProps
32:
33: const CarouselContext = React.createContext<CarouselContextProps | null>(null)
34:
35: function useCarousel() {
36:   const context = React.useContext(CarouselContext)
37:
38:   if (!context) {
39:     throw new Error("useCarousel must be used within a <Carousel />")
40:   }
41:
42:   return context
43: }
44:
45: const Carousel = React.forwardRef<
46:   HTMLDivElement,
47:   React.HTMLAttributes<HTMLDivElement> & CarouselProps
48: >(
49:   (
50:     {
51:       orientation = "horizontal",
52:       opts,
53:       setApi,
54:       plugins,
55:       className,
56:       children,
57:       ...props
58:     },
59:     ref
60:   ) => {
61:     const [carouselRef, api] = useEmblaCarousel(
62:       {
63:         ...opts,
64:         axis: orientation === "horizontal" ? "x" : "y",
65:       },
66:       plugins
67:     )
68:     const [canScrollPrev, setCanScrollPrev] = React.useState(false)
69:     const [canScrollNext, setCanScrollNext] = React.useState(false)
70:
71:     const onSelect = React.useCallback((api: CarouselApi) => {
72:       if (!api) {
73:         return
74:       }
75:
76:       setCanScrollPrev(api.canScrollPrev())
77:       setCanScrollNext(api.canScrollNext())
78:     }, [])
79:
80:     const scrollPrev = React.useCallback(() => {
81:       api?.scrollPrev()
82:     }, [api])
83:
84:     const scrollNext = React.useCallback(() => {
85:       api?.scrollNext()
86:     }, [api])
87:
88:     const handleKeyDown = React.useCallback(
89:       (event: React.KeyboardEvent<HTMLDivElement>) => {
90:         if (event.key === "ArrowLeft") {
91:           event.preventDefault()
92:           scrollPrev()
93:         } else if (event.key === "ArrowRight") {

```

```

94:         event.preventDefault()
95:         scrollNext()
96:     }
97: },
98: [scrollPrev, scrollNext]
99: )
100:
101: React.useEffect(() => {
102:     if (!api || !setApi) {
103:         return
104:     }
105:
106:     setApi(api)
107: }, [api, setApi])
108:
109: React.useEffect(() => {
110:     if (!api) {
111:         return
112:     }
113:
114:     onSelect(api)
115:     api.on("reInit", onSelect)
116:     api.on("select", onSelect)
117:
118:     return () => {
119:         api?.off("select", onSelect)
120:     }
121: }, [api, onSelect])
122:
123: return (
124:     <CarouselContext.Provider
125:         value={{
126:             carouselRef,
127:             api: api,
128:             opts,
129:             orientation:
130:                 orientation || (opts?.axis === "y" ? "vertical" : "horizontal"),
131:             scrollPrev,
132:             scrollNext,
133:             canScrollPrev,
134:             canScrollNext,
135:         }}
136:     >
137:         <div
138:             ref={ref}
139:             onKeyDownCapture={handleKeyDown}
140:             className={cn("relative", className)}
141:             role="region"
142:             aria-roledescription="carousel"
143:             {...props}
144:         >
145:             {children}
146:         </div>
147:     </CarouselContext.Provider>
148: )
149: }
150: )
151: Carousel.displayName = "Carousel"
152:
153: const CarouselContent = React.forwardRef<
154:     HTMLDivElement,
155:     React.HTMLAttributes<HTMLDivElement>
156: >(({ className, ...props }, ref) => {
157:     const { carouselRef, orientation } = useCarousel()
158:
159:     return (
160:         <div ref={carouselRef} className="overflow-hidden">
161:             <div
162:                 ref={ref}
163:                 className={cn(
164:                     "flex",
165:                     orientation === "horizontal" ? "-ml-4" : "-mt-4 flex-col",
166:                     className

```

```

167:         })
168:         {...props}
169:       />
170:     </div>
171:   )
172: })
173: CarouselContent.displayName = "CarouselContent"
174:
175: const CarouselItem = React.forwardRef<
176:   HTMLDivElement,
177:   React.HTMLAttributes<HTMLDivElement>
178: >(({ className, ...props }, ref) => {
179:   const { orientation } = useCarousel()
180:
181:   return (
182:     <div
183:       ref={ref}
184:       role="group"
185:       aria-roledescription="slide"
186:       className={cn(
187:         "min-w-0 shrink-0 grow-0 basis-full",
188:         orientation === "horizontal" ? "pl-4" : "pt-4",
189:         className
190:       )}
191:       {...props}
192:     />
193:   )
194: })
195: CarouselItem.displayName = "CarouselItem"
196:
197: const CarouselPrevious = React.forwardRef<
198:   HTMLButtonElement,
199:   React.ComponentProps<typeof Button>
200: >(({ className, variant = "outline", size = "icon", ...props }, ref) => {
201:   const { orientation, scrollPrev, canScrollPrev } = useCarousel()
202:
203:   return (
204:     <Button
205:       ref={ref}
206:       variant={variant}
207:       size={size}
208:       className={cn(
209:         "absolute h-8 w-8 rounded-full",
210:         orientation === "horizontal"
211:           ? "-left-12 top-1/2 -translate-y-1/2"
212:           : "-top-12 left-1/2 -translate-x-1/2 rotate-90",
213:         className
214:       )}
215:       disabled={!canScrollPrev}
216:       onClick={scrollPrev}
217:       {...props}
218:     >
219:       <ArrowLeft className="h-4 w-4" />
220:       <span className="sr-only">Previous slide</span>
221:     </Button>
222:   )
223: })
224: CarouselPrevious.displayName = "CarouselPrevious"
225:
226: const CarouselNext = React.forwardRef<
227:   HTMLButtonElement,
228:   React.ComponentProps<typeof Button>
229: >(({ className, variant = "outline", size = "icon", ...props }, ref) => {
230:   const { orientation, scrollNext, canScrollNext } = useCarousel()
231:
232:   return (
233:     <Button
234:       ref={ref}
235:       variant={variant}
236:       size={size}
237:       className={cn(
238:         "absolute h-8 w-8 rounded-full",
239:         orientation === "horizontal"

```

```

240:         ? "-right-12 top-1/2 -translate-y-1/2"
241:         : "-bottom-12 left-1/2 -translate-x-1/2 rotate-90",
242:         className
243:     )}
244:     disabled={!canScrollNext}
245:     onClick={scrollNext}
246:     {...props}
247: >
248:     <ArrowRight className="h-4 w-4" />
249:     <span className="sr-only">Next slide</span>
250: </Button>
251: )
252: })
253: CarouselNext.displayName = "CarouselNext"
254:
255: export {
256:     type CarouselApi,
257:     Carousel,
258:     CarouselContent,
259:     CarouselItem,
260:     CarouselPrevious,
261:     CarouselNext,
262: }

```

## ■ File: components/ui/chart.tsx

```

1: "use client"
2:
3: import * as React from "react"
4: import * as RechartsPrimitive from "recharts"
5:
6: import { cn } from "@lib/utils"
7:
8: // Format: { THEME_NAME: CSS_SELECTOR }
9: const THEMES = { light: "", dark: ".dark" } as const
10:
11: export type ChartConfig = {
12:   [k in string]: {
13:     label?: React.ReactNode
14:     icon?: React.ComponentType
15:   } & (
16:     | { color?: string; theme?: never }
17:     | { color?: never; theme: Record<keyof typeof THEMES, string> }
18:   )
19: }
20:
21: type ChartContextProps = {
22:   config: ChartConfig
23: }
24:
25: const ChartContext = React.createContext<ChartContextProps | null>(null)
26:
27: function useChart() {
28:   const context = React.useContext(ChartContext)
29:
30:   if (!context) {
31:     throw new Error("useChart must be used within a <ChartContainer />")
32:   }
33:
34:   return context
35: }
36:
37: const ChartContainer = React.forwardRef<
38:   HTMLDivElement,
39:   React.ComponentProps<"div"> & {
40:     config: ChartConfig
41:     children: React.ComponentProps<
42:       typeof RechartsPrimitive.ResponsiveContainer
43:     >["children"]
44:   }
45: >(({ id, className, children, config, ...props }, ref) => {

```



```

46:   const uniqueId = React.useId()
47:   const chartId = `chart-${id || uniqueId.replace(/:/g, "")}`
48:
49:   return (
50:     <ChartContext.Provider value={{ config }}>
51:       <div
52:         data-chart={chartId}
53:         ref={ref}
54:         className={cn(
55:           "flex aspect-video justify-center text-xs [&_.recharts-cartesian-axis-tick_text]:fill-muted-foreground",
56:           className
57:         )}
58:         {...props}
59:       >
60:         <ChartStyle id={chartId} config={config} />
61:         <RechartsPrimitive.ResponsiveContainer>
62:           {children}
63:         </RechartsPrimitive.ResponsiveContainer>
64:       </div>
65:     </ChartContext.Provider>
66:   )
67: })
68: ChartContainer.displayName = "Chart"
69:
70: const ChartStyle = ({ id, config }: { id: string; config: ChartConfig }) => {
71:   const colorConfig = Object.entries(config).filter(
72:     ([_, config]) => config.theme || config.color
73:   )
74:
75:   if (!colorConfig.length) {
76:     return null
77:   }
78:
79:   return (
80:     <style
81:       dangerouslySetInnerHTML={{
82:         __html: Object.entries(THEMES)
83:           .map(
84:             ([theme, prefix]) => `
85:   ${prefix} [data-chart=${id}] {
86:     ${colorConfig
87:       .map(([key, itemConfig]) => {
88:         const color =
89:           itemConfig.theme?.[theme as keyof typeof itemConfig.theme] ||
90:           itemConfig.color
91:         return color ? `  --color-${key}: ${color};` : null
92:       })
93:       .join("\n")}
94:   }
95: `
96:       )
97:       .join("\n"),
98:     >
99:   />
100: )
101: }
102:
103: const ChartTooltip = RechartsPrimitive.Tooltip
104:
105: const ChartTooltipContent = React.forwardRef<
106:   HTMLDivElement,
107:   React.ComponentProps<typeof RechartsPrimitive.Tooltip> &
108:   React.ComponentProps<"div"> & {
109:     hideLabel?: boolean
110:     hideIndicator?: boolean
111:     indicator?: "line" | "dot" | "dashed"
112:     nameKey?: string
113:     labelKey?: string
114:   }
115: >((
116:   {
117:     {
118:       active,

```

```

119:     payload,
120:     className,
121:     indicator = "dot",
122:     hideLabel = false,
123:     hideIndicator = false,
124:     label,
125:     labelFormatter,
126:     labelClassName,
127:     formatter,
128:     color,
129:     nameKey,
130:     labelKey,
131:   },
132:   ref
133: ) => {
134:   const { config } = useChart()
135:
136:   const tooltipLabel = React.useMemo(() => {
137:     if (hideLabel || !payload?.length) {
138:       return null
139:     }
140:
141:     const [item] = payload
142:     const key = `${labelKey || item.dataKey || item.name || "value"}`
143:     const itemConfig = getPayloadConfigFromPayload(config, item, key)
144:     const value =
145:       !labelKey && typeof label === "string"
146:         ? config[label as keyof typeof config]?.label || label
147:         : itemConfig?.label
148:
149:     if (labelFormatter) {
150:       return (
151:         <div className={cn("font-medium", labelClassName)}>
152:           {labelFormatter(value, payload)}
153:         </div>
154:       )
155:     }
156:
157:     if (!value) {
158:       return null
159:     }
160:
161:     return <div className={cn("font-medium", labelClassName)}>{value}</div>
162:   }, [
163:     label,
164:     labelFormatter,
165:     payload,
166:     hideLabel,
167:     labelClassName,
168:     config,
169:     labelKey,
170:   ])
171:
172:   if (!active || !payload?.length) {
173:     return null
174:   }
175:
176:   const nestLabel = payload.length === 1 && indicator !== "dot"
177:
178:   return (
179:     <div
180:       ref={ref}
181:       className={cn(
182:         "grid min-w-[8rem] items-start gap-1.5 rounded-lg border border-border/50 bg-background px-2.5 py
183:       className
184:     )}
185:   >
186:     {!nestLabel ? tooltipLabel : null}
187:     <div className="grid gap-1.5">
188:       {payload.map((item, index) => {
189:         const key = `${nameKey || item.name || item.dataKey || "value"}`
190:         const itemConfig = getPayloadConfigFromPayload(config, item, key)
191:         const indicatorColor = color || item.payload.fill || item.color

```

```

192:
193:     return (
194:         <div
195:             key={item.dataKey}
196:             className={cn(
197:                 "flex w-full flex-wrap items-stretch gap-2 [>svg]:h-2.5 [>svg]:w-2.5 [>svg]:text-muted-foreground",
198:                 indicator === "dot" && "items-center"
199:             )}
200:         >
201:             {formatter && item?.value !== undefined && item.name ? (
202:                 formatter(item.value, item.name, item, index, item.payload)
203:             ) : (
204:                 <>
205:                     {itemConfig?.icon ? (
206:                         <itemConfig.icon />
207:                     ) : (
208:                         !hideIndicator && (
209:                             <div
210:                                 className={cn(
211:                                     "shrink-0 rounded-[2px] border-[--color-border] bg-[--color-bg]",
212:                                     {
213:                                         "h-2.5 w-2.5": indicator === "dot",
214:                                         "w-1": indicator === "line",
215:                                         "w-0 border-[1.5px] border-dashed bg-transparent":
216:                                             indicator === "dashed",
217:                                         "my-0.5": nestLabel && indicator === "dashed",
218:                                     }
219:                                 )}
220:                                 style={
221:                                     {
222:                                         "--color-bg": indicatorColor,
223:                                         "--color-border": indicatorColor,
224:                                     } as React.CSSProperties
225:                                 }
226:                             />
227:                         )
228:                     )}
229:                     <div
230:                         className={cn(
231:                             "flex flex-1 justify-between leading-none",
232:                             nestLabel ? "items-end" : "items-center"
233:                         )}
234:                     >
235:                         <div className="grid gap-1.5">
236:                             {nestLabel ? tooltipLabel : null}
237:                             <span className="text-muted-foreground">
238:                                 {itemConfig?.label || item.name}
239:                             </span>
240:                         </div>
241:                         {item.value && (
242:                             <span className="font-mono font-medium tabular-nums text-foreground">
243:                                 {item.value.toLocaleString()}
244:                             </span>
245:                         )}
246:                     </div>
247:                 </>
248:             )}
249:         </div>
250:     )
251: }}
252: </div>
253: </div>
254: )
255: }
256: )
257: ChartTooltipContent.displayName = "ChartTooltip"
258:
259: const ChartLegend = RechartsPrimitive.Legend
260:
261: const ChartLegendContent = React.forwardRef<
262:     HTMLDivElement,
263:     React.ComponentProps<"div"> &
264:         Pick<RechartsPrimitive.LegendProps, "payload" | "verticalAlign"> & {

```

```

265:     hideIcon?: boolean
266:     nameKey?: string
267:   }
268: >(
269:   (
270:     { className, hideIcon = false, payload, verticalAlign = "bottom", nameKey },
271:     ref
272:   ) => {
273:     const { config } = useChart()
274:
275:     if (!payload?.length) {
276:       return null
277:     }
278:
279:     return (
280:       <div
281:         ref={ref}
282:         className={cn(
283:           "flex items-center justify-center gap-4",
284:           verticalAlign === "top" ? "pb-3" : "pt-3",
285:           className
286:         )}
287:       >
288:         {payload.map((item) => {
289:           const key = `${nameKey} || item.dataKey || "value"}`
290:           const itemConfig = getPayloadConfigFromPayload(config, item, key)
291:
292:           return (
293:             <div
294:               key={item.value}
295:               className={cn(
296:                 "flex items-center gap-1.5 [&svg]:h-3 [&svg]:w-3 [&svg]:text-muted-foreground"
297:               )}
298:             >
299:               {itemConfig?.icon && !hideIcon ? (
300:                 <itemConfig.icon />
301:               ) : (
302:                 <div
303:                   className="h-2 w-2 shrink-0 rounded-[2px]"
304:                   style={{
305:                     backgroundColor: item.color,
306:                   }}
307:                 />
308:               )}
309:               {itemConfig?.label}
310:             </div>
311:           )
312:         })}
313:       </div>
314:     )
315:   }
316: )
317: ChartLegendContent.displayName = "ChartLegend"
318:
319: // Helper to extract item config from a payload.
320: function getPayloadConfigFromPayload(
321:   config: ChartConfig,
322:   payload: unknown,
323:   key: string
324: ) {
325:   if (typeof payload !== "object" || payload === null) {
326:     return undefined
327:   }
328:
329:   const payloadPayload =
330:     "payload" in payload &&
331:     typeof payload.payload === "object" &&
332:     payload.payload !== null
333:       ? payload.payload
334:       : undefined
335:
336:   let configLabelKey: string = key
337:

```

```

338:   if (
339:     key in payload &&
340:     typeof payload[key as keyof typeof payload] === "string"
341:   ) {
342:     configLabelKey = payload[key as keyof typeof payload] as string
343:   } else if (
344:     payloadPayload &&
345:     key in payloadPayload &&
346:     typeof payloadPayload[key as keyof typeof payloadPayload] === "string"
347:   ) {
348:     configLabelKey = payloadPayload[
349:       key as keyof typeof payloadPayload
350:     ] as string
351:   }
352:
353:   return configLabelKey in config
354:     ? config[configLabelKey]
355:     : config[key as keyof typeof config]
356: }
357:
358: export {
359:   ChartContainer,
360:   ChartTooltip,
361:   ChartTooltipContent,
362:   ChartLegend,
363:   ChartLegendContent,
364:   ChartStyle,
365: }

```

## ■ File: components/ui/checkbox.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import * as CheckboxPrimitive from "@radix-ui/react-checkbox"
5: import { Check } from "lucide-react"
6:
7: import { cn } from "@lib/utils"
8:
9: const Checkbox = React.forwardRef<
10:   React.ElementRef<typeof CheckboxPrimitive.Root>,
11:   React.ComponentPropsWithoutRef<typeof CheckboxPrimitive.Root>
12: >(({ className, ...props }, ref) => (
13:   <CheckboxPrimitive.Root
14:     ref={ref}
15:     className={cn(
16:       "peer h-4 w-4 shrink-0 rounded-sm border border-primary ring-offset-background focus-visible:outline-1
17:       className
18:     )}
19:     {...props}
20:   >
21:     <CheckboxPrimitive.Indicator
22:       className={cn("flex items-center justify-center text-current")}
23:     >
24:       <Check className="h-4 w-4" />
25:     </CheckboxPrimitive.Indicator>
26:   </CheckboxPrimitive.Root>
27: ))
28: Checkbox.displayName = CheckboxPrimitive.Root.displayName
29:
30: export { Checkbox }

```

## ■ File: components/ui/collapsible.tsx

```

=====
1: "use client"
2:
3: import * as CollapsiblePrimitive from "@radix-ui/react-collapsible"
4:
5: const Collapsible = CollapsiblePrimitive.Root

```

```

6:
7: const CollapsibleTrigger = CollapsiblePrimitive.CollapsibleTrigger
8:
9: const CollapsibleContent = CollapsiblePrimitive.CollapsibleContent
10:
11: export { Collapsible, CollapsibleTrigger, CollapsibleContent }

```

## ■ File: components/ui/command.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import { type DialogProps } from "@radix-ui/react-dialog"
5: import { Command as CommandPrimitive } from "cmdk"
6: import { Search } from "lucide-react"
7:
8: import { cn } from "@/lib/utils"
9: import { Dialog, DialogContent } from "@/components/ui/dialog"
10:
11: const Command = React.forwardRef<
12:   React.ElementRef<typeof CommandPrimitive>,
13:   React.ComponentPropsWithoutRef<typeof CommandPrimitive>
14: >(({ className, ...props }, ref) => (
15:   <CommandPrimitive
16:     ref={ref}
17:     className={cn(
18:       "flex h-full w-full flex-col overflow-hidden rounded-md bg-popover text-popover-foreground",
19:       className
20:     )}
21:     {...props}
22:   />
23: ))
24: Command.displayName = CommandPrimitive.displayName
25:
26: const CommandDialog = ({ children, ...props }: DialogProps) => {
27:   return (
28:     <Dialog {...props}>
29:       <DialogContent className="overflow-hidden p-0 shadow-lg">
30:         <Command className="[&_[:cmdk-group-heading]]:px-2 [&_[:cmdk-group-heading]]:font-medium [&_[:cmdk-group-heading]]:text-sm">
31:           {children}
32:         </Command>
33:       </DialogContent>
34:     </Dialog>
35:   )
36: }
37:
38: const CommandInput = React.forwardRef<
39:   React.ElementRef<typeof CommandPrimitive.Input>,
40:   React.ComponentPropsWithoutRef<typeof CommandPrimitive.Input>
41: >(({ className, ...props }, ref) => (
42:   <div className="flex items-center border-b px-3 cmdk-input-wrapper=">
43:     <Search className="mr-2 h-4 w-4 shrink-0 opacity-50" />
44:     <CommandPrimitive.Input
45:       ref={ref}
46:       className={cn(
47:         "flex h-11 w-full rounded-md bg-transparent py-3 text-sm outline-none placeholder:text-muted-foreground",
48:         className
49:       )}
50:       {...props}
51:     />
52:   </div>
53: ))
54:
55: CommandInput.displayName = CommandPrimitive.Input.displayName
56:
57: const CommandList = React.forwardRef<
58:   React.ElementRef<typeof CommandPrimitive.List>,
59:   React.ComponentPropsWithoutRef<typeof CommandPrimitive.List>
60: >(({ className, ...props }, ref) => (
61:   <CommandPrimitive.List
62:     ref={ref}

```

```

63:     className={cn("max-h-[300px] overflow-y-auto overflow-x-hidden", className)}
64:     {...props}
65:   />
66: ))
67:
68: CommandList.displayName = CommandPrimitive.List.displayName
69:
70: const CommandEmpty = React.forwardRef<
71:   React.ElementRef<typeof CommandPrimitive.Empty>,
72:   React.ComponentPropsWithoutRef<typeof CommandPrimitive.Empty>
73: >((props, ref) => (
74:   <CommandPrimitive.Empty
75:     ref={ref}
76:     className="py-6 text-center text-sm"
77:     {...props}
78:   />
79: ))
80:
81: CommandEmpty.displayName = CommandPrimitive.Empty.displayName
82:
83: const CommandGroup = React.forwardRef<
84:   React.ElementRef<typeof CommandPrimitive.Group>,
85:   React.ComponentPropsWithoutRef<typeof CommandPrimitive.Group>
86: >(({ className, ...props }, ref) => (
87:   <CommandPrimitive.Group
88:     ref={ref}
89:     className={cn(
90:       "overflow-hidden p-1 text-foreground [&_cmdk-group-heading]:px-2 [&_cmdk-group-heading]:py-1.5 [&
91:       className
92:     )}
93:     {...props}
94:   />
95: ))
96:
97: CommandGroup.displayName = CommandPrimitive.Group.displayName
98:
99: const CommandSeparator = React.forwardRef<
100:   React.ElementRef<typeof CommandPrimitive.Separator>,
101:   React.ComponentPropsWithoutRef<typeof CommandPrimitive.Separator>
102: >(({ className, ...props }, ref) => (
103:   <CommandPrimitive.Separator
104:     ref={ref}
105:     className={cn("-mx-1 h-px bg-border", className)}
106:     {...props}
107:   />
108: ))
109: CommandSeparator.displayName = CommandPrimitive.Separator.displayName
110:
111: const CommandItem = React.forwardRef<
112:   React.ElementRef<typeof CommandPrimitive.Item>,
113:   React.ComponentPropsWithoutRef<typeof CommandPrimitive.Item>
114: >(({ className, ...props }, ref) => (
115:   <CommandPrimitive.Item
116:     ref={ref}
117:     className={cn(
118:       "relative flex cursor-default gap-2 select-none items-center rounded-sm px-2 py-1.5 text-sm outline-n
119:     className
120:   )}
121:   {...props}
122: />
123: ))
124:
125: CommandItem.displayName = CommandPrimitive.Item.displayName
126:
127: const CommandShortcut = ({
128:   className,
129:   ...props
130: }: React.HTMLAttributes<HTMLSpanElement>) => {
131:   return (
132:     <span
133:       className={cn(
134:         "ml-auto text-xs tracking-widest text-muted-foreground",
135:         className

```

```

136:     })
137:     {...props}
138:   />
139: )
140: }
141: CommandShortcut.displayName = "CommandShortcut"
142:
143: export {
144:   Command,
145:   CommandDialog,
146:   CommandInput,
147:   CommandList,
148:   CommandEmpty,
149:   CommandGroup,
150:   CommandItem,
151:   CommandShortcut,
152:   CommandSeparator,
153: }

```

## ■ File: components/ui/context-menu.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import * as ContextMenuPrimitive from "@radix-ui/react-context-menu"
5: import { Check, ChevronRight, Circle } from "lucide-react"
6:
7: import { cn } from "@lib/Utils"
8:
9: const ContextMenu = ContextMenuPrimitive.Root
10:
11: const ContextMenuTrigger = ContextMenuPrimitive.Trigger
12:
13: const ContextMenuGroup = ContextMenuPrimitive.Group
14:
15: const ContextMenuPortal = ContextMenuPrimitive.Portal
16:
17: const ContextMenuSub = ContextMenuPrimitive.Sub
18:
19: const ContextMenuRadioGroup = ContextMenuPrimitive.RadioGroup
20:
21: const ContextMenuSubTrigger = React.forwardRef<
22:   React.ElementRef<typeof ContextMenuPrimitive.SubTrigger>,
23:   React.ComponentPropsWithoutRef<typeof ContextMenuPrimitive.SubTrigger> & {
24:     inset?: boolean
25:   }
26: >(({ className, inset, children, ...props }, ref) => (
27:   <ContextMenuPrimitive.SubTrigger
28:     ref={ref}
29:     className={cn(
30:       "flex cursor-default select-none items-center rounded-sm px-2 py-1.5 text-sm outline-none focus:bg-accent",
31:       inset && "pl-8",
32:       className
33:     )}
34:     {...props}
35:   >
36:     {children}
37:     <ChevronRight className="ml-auto h-4 w-4" />
38:   </ContextMenuPrimitive.SubTrigger>
39: ))
40: ContextMenuSubTrigger.displayName = ContextMenuPrimitive.SubTrigger.displayName
41:
42: const ContextMenuSubContent = React.forwardRef<
43:   React.ElementRef<typeof ContextMenuPrimitive.SubContent>,
44:   React.ComponentPropsWithoutRef<typeof ContextMenuPrimitive.SubContent>
45: >(({ className, ...props }, ref) => (
46:   <ContextMenuPrimitive.SubContent
47:     ref={ref}
48:     className={cn(
49:       "z-50 min-w-[8rem] overflow-hidden rounded-md border bg-popover p-1 text-popover-foreground shadow-md",
50:       className

```



```

51:     })
52:     {...props}
53:   />
54: ))
55: ContextMenuSubContent.displayName = ContextMenuPrimitive.SubContent.displayName
56:
57: const ContextMenuContent = React.forwardRef<
58:   React.ElementRef<typeof ContextMenuPrimitive.Content>,
59:   React.ComponentPropsWithoutRef<typeof ContextMenuPrimitive.Content>
60: >(({ className, ...props }, ref) => (
61:   <ContextMenuPrimitive.Portal>
62:     <ContextMenuPrimitive.Content
63:       ref={ref}
64:       className={cn(
65:         "z-50 min-w-[8rem] overflow-hidden rounded-md border bg-popover p-1 text-popover-foreground shadow-
66:         className
67:       )}
68:       {...props}
69:     />
70:   </ContextMenuPrimitive.Portal>
71: ))
72: ContextMenuContent.displayName = ContextMenuPrimitive.Content.displayName
73:
74: const ContextMenuItem = React.forwardRef<
75:   React.ElementRef<typeof ContextMenuPrimitive.Item>,
76:   React.ComponentPropsWithoutRef<typeof ContextMenuPrimitive.Item> & {
77:     inset?: boolean
78:   }
79: >(({ className, inset, ...props }, ref) => (
80:   <ContextMenuPrimitive.Item
81:     ref={ref}
82:     className={cn(
83:       "relative flex cursor-default select-none items-center rounded-sm px-2 py-1.5 text-sm outline-none fo
84:       inset && "pl-8",
85:       className
86:     )}
87:     {...props}
88:   />
89: ))
90: ContextMenuItem.displayName = ContextMenuPrimitive.Item.displayName
91:
92: const ContextMenuCheckboxItem = React.forwardRef<
93:   React.ElementRef<typeof ContextMenuPrimitive.CheckboxItem>,
94:   React.ComponentPropsWithoutRef<typeof ContextMenuPrimitive.CheckboxItem>
95: >(({ className, children, checked, ...props }, ref) => (
96:   <ContextMenuPrimitive.CheckboxItem
97:     ref={ref}
98:     className={cn(
99:       "relative flex cursor-default select-none items-center rounded-sm py-1.5 pl-8 pr-2 text-sm outline-no
100:     className
101:   )}
102:   checked={checked}
103:   {...props}
104: >
105:   <span className="absolute left-2 flex h-3.5 w-3.5 items-center justify-center">
106:     <ContextMenuPrimitive.ItemIndicator>
107:       <Check className="h-4 w-4" />
108:     </ContextMenuPrimitive.ItemIndicator>
109:   </span>
110:   {children}
111: </ContextMenuPrimitive.CheckboxItem>
112: ))
113: ContextMenuCheckboxItem.displayName =
114:   ContextMenuPrimitive.CheckboxItem.displayName
115:
116: const ContextMenuRadioItem = React.forwardRef<
117:   React.ElementRef<typeof ContextMenuPrimitive.RadioItem>,
118:   React.ComponentPropsWithoutRef<typeof ContextMenuPrimitive.RadioItem>
119: >(({ className, children, ...props }, ref) => (
120:   <ContextMenuPrimitive.RadioItem
121:     ref={ref}
122:     className={cn(
123:       "relative flex cursor-default select-none items-center rounded-sm py-1.5 pl-8 pr-2 text-sm outline-no

```

```

124:     className
125:   })
126:   {...props}
127: >
128:   <span className="absolute left-2 flex h-3.5 w-3.5 items-center justify-center">
129:     <ContextMenuPrimitive.ItemIndicator>
130:       <Circle className="h-2 w-2 fill-current" />
131:     </ContextMenuPrimitive.ItemIndicator>
132:   </span>
133:   {children}
134: </ContextMenuPrimitive.RadioItem>
135: ))
136: ContextMenuRadioItem.displayName = ContextMenuPrimitive.RadioItem.displayName
137:
138: const ContextMenuLabel = React.forwardRef<
139:   React.ElementRef<typeof ContextMenuPrimitive.Label>,
140:   React.ComponentPropsWithoutRef<typeof ContextMenuPrimitive.Label> & {
141:     inset?: boolean
142:   }
143: >(({ className, inset, ...props }, ref) => (
144:   <ContextMenuPrimitive.Label
145:     ref={ref}
146:     className={cn(
147:       "px-2 py-1.5 text-sm font-semibold text-foreground",
148:       inset && "pl-8",
149:       className
150:     )}
151:     {...props}
152:   />
153: ))
154: ContextMenuLabel.displayName = ContextMenuPrimitive.Label.displayName
155:
156: const ContextMenuSeparator = React.forwardRef<
157:   React.ElementRef<typeof ContextMenuPrimitive.Separator>,
158:   React.ComponentPropsWithoutRef<typeof ContextMenuPrimitive.Separator>
159: >(({ className, ...props }, ref) => (
160:   <ContextMenuPrimitive.Separator
161:     ref={ref}
162:     className={cn("-mx-1 my-1 h-px bg-border", className)}
163:     {...props}
164:   />
165: ))
166: ContextMenuSeparator.displayName = ContextMenuPrimitive.Separator.displayName
167:
168: const ContextMenuShortcut = ({
169:   className,
170:   ...props
171: }: React.HTMLAttributes<HTMLSpanElement>) => {
172:   return (
173:     <span
174:       className={cn(
175:         "ml-auto text-xs tracking-widest text-muted-foreground",
176:         className
177:       )}
178:       {...props}
179:     />
180:   )
181: }
182: ContextMenuShortcut.displayName = "ContextMenuShortcut"
183:
184: export {
185:   ContextMenu,
186:   ContextMenuTrigger,
187:   ContextMenuContent,
188:   ContextMenuItem,
189:   ContextMenuCheckboxItem,
190:   ContextMenuRadioItem,
191:   ContextMenuLabel,
192:   ContextMenuSeparator,
193:   ContextMenuShortcut,
194:   ContextMenuGroup,
195:   ContextMenuPortal,
196:   ContextMenuSub,

```

```

197:   ContextMenuSubContent,
198:   ContextMenuSubTrigger,
199:   ContextMenuRadioGroup,
200: }

```

## ■ File: components\ui\dialog.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import * as DialogPrimitive from "@radix-ui/react-dialog"
5: import { X } from "lucide-react"
6:
7: import { cn } from "@lib/utils"
8:
9: const Dialog = DialogPrimitive.Root
10:
11: const DialogTrigger = DialogPrimitive.Trigger
12:
13: const DialogPortal = DialogPrimitive.Portal
14:
15: const DialogClose = DialogPrimitive.Close
16:
17: const DialogOverlay = React.forwardRef<
18:   React.ElementRef<typeof DialogPrimitive.Overlay>,
19:   React.ComponentPropsWithoutRef<typeof DialogPrimitive.Overlay>
20: >(({ className, ...props }, ref) => (
21:   <DialogPrimitive.Overlay
22:     ref={ref}
23:     className={cn(
24:       "fixed inset-0 z-50 bg-black/80 data-[state=open]:animate-in data-[state=closed]:animate-out data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0",
25:       className
26:     )}
27:     {...props}
28:   />
29: ))
30: DialogOverlay.displayName = DialogPrimitive.Overlay.displayName
31:
32: const DialogContent = React.forwardRef<
33:   React.ElementRef<typeof DialogPrimitive.Content>,
34:   React.ComponentPropsWithoutRef<typeof DialogPrimitive.Content>
35: >(({ className, children, ...props }, ref) => (
36:   <DialogPortal>
37:     <DialogOverlay />
38:     <DialogPrimitive.Content
39:       ref={ref}
40:       className={cn(
41:         "fixed left-[50%] top-[50%] z-50 grid w-full max-w-lg translate-x-[-50%] translate-y-[-50%] gap-4 bg-white p-4 shadow-lg",
42:         className
43:       )}
44:       {...props}
45:     >
46:       {children}
47:       <DialogPrimitive.Close className="absolute right-4 top-4 rounded-sm opacity-70 ring-offset-background transition-opacity hover:opacity-100 focus:outline-none focus:ring-2 focus:ring-gray-400 focus:ring-offset-2 disabled:cursor-not-allowed disabled:opacity-50" />
48:       <span className="sr-only">Close</span>
49:     </DialogPrimitive.Close>
50:   </DialogPrimitive.Content>
51: </DialogPortal>
52: ))
53: DialogContent.displayName = DialogPrimitive.Content.displayName
54:
55:
56: const DialogHeader = ({
57:   className,
58:   ...props
59: }: React.HTMLAttributes<HTMLDivElement>) => (
60:   <div
61:     className={cn(
62:       "flex flex-col space-y-1.5 text-center sm:text-left",
63:       className
64:     )}

```

```

65:     {...props}
66:   />
67: )
68: DialogHeader.displayName = "DialogHeader"
69:
70: const DialogFooter = ({
71:   className,
72:   ...props
73: }: React.HTMLAttributes<HTMLDivElement>) => (
74:   <div
75:     className={cn(
76:       "flex flex-col-reverse sm:flex-row sm:justify-end sm:space-x-2",
77:       className
78:     )}
79:     {...props}
80:   />
81: )
82: DialogFooter.displayName = "DialogFooter"
83:
84: const DialogTitle = React.forwardRef<
85:   React.ElementRef<typeof DialogPrimitive.Title>,
86:   React.ComponentPropsWithoutRef<typeof DialogPrimitive.Title>
87: >(({ className, ...props }, ref) => (
88:   <DialogPrimitive.Title
89:     ref={ref}
90:     className={cn(
91:       "text-lg font-semibold leading-none tracking-tight",
92:       className
93:     )}
94:     {...props}
95:   />
96: ))
97: DialogTitle.displayName = DialogPrimitive.Title.displayName
98:
99: const DialogDescription = React.forwardRef<
100:   React.ElementRef<typeof DialogPrimitive.Description>,
101:   React.ComponentPropsWithoutRef<typeof DialogPrimitive.Description>
102: >(({ className, ...props }, ref) => (
103:   <DialogPrimitive.Description
104:     ref={ref}
105:     className={cn("text-sm text-muted-foreground", className)}
106:     {...props}
107:   />
108: ))
109: DialogDescription.displayName = DialogPrimitive.Description.displayName
110:
111: export {
112:   Dialog,
113:   DialogPortal,
114:   DialogOverlay,
115:   DialogClose,
116:   DialogTrigger,
117:   DialogContent,
118:   DialogHeader,
119:   DialogFooter,
120:   DialogTitle,
121:   DialogDescription,
122: }

```

## ■ File: components/ui/drawer.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import { Drawer as DrawerPrimitive } from "vaul"
5:
6: import { cn } from "@/lib/utils"
7:
8: const Drawer = ({
9:   shouldScaleBackground = true,
10:   ...props

```

```

11: } : React.ComponentProps<typeof DrawerPrimitive.Root>) => (
12:   <DrawerPrimitive.Root
13:     shouldScaleBackground={shouldScaleBackground}
14:     {...props}
15:   />
16: )
17: Drawer.displayName = "Drawer"
18:
19: const DrawerTrigger = DrawerPrimitive.Trigger
20:
21: const DrawerPortal = DrawerPrimitive.Portal
22:
23: const DrawerClose = DrawerPrimitive.Close
24:
25: const DrawerOverlay = React.forwardRef<
26:   React.ElementRef<typeof DrawerPrimitive.Overlay>,
27:   React.ComponentPropsWithoutRef<typeof DrawerPrimitive.Overlay>
28: >(({ className, ...props }, ref) => (
29:   <DrawerPrimitive.Overlay
30:     ref={ref}
31:     className={cn("fixed inset-0 z-50 bg-black/80", className)}
32:     {...props}
33:   />
34: ))
35: DrawerOverlay.displayName = DrawerPrimitive.Overlay.displayName
36:
37: const DrawerContent = React.forwardRef<
38:   React.ElementRef<typeof DrawerPrimitive.Content>,
39:   React.ComponentPropsWithoutRef<typeof DrawerPrimitive.Content>
40: >(({ className, children, ...props }, ref) => (
41:   <DrawerPortal>
42:     <DrawerOverlay />
43:     <DrawerPrimitive.Content
44:       ref={ref}
45:       className={cn(
46:         "fixed inset-x-0 bottom-0 z-50 mt-24 flex h-auto flex-col rounded-t-[10px] border bg-background",
47:         className
48:       )}
49:       {...props}
50:     >
51:       <div className="mx-auto mt-4 h-2 w-[100px] rounded-full bg-muted" />
52:       {children}
53:     </DrawerPrimitive.Content>
54:   </DrawerPortal>
55: ))
56: DrawerContent.displayName = "DrawerContent"
57:
58: const DrawerHeader = ({
59:   className,
60:   ...props
61: }: React.HTMLAttributes<HTMLDivElement>) => (
62:   <div
63:     className={cn("grid gap-1.5 p-4 text-center sm:text-left", className)}
64:     {...props}
65:   />
66: )
67: DrawerHeader.displayName = "DrawerHeader"
68:
69: const DrawerFooter = ({
70:   className,
71:   ...props
72: }: React.HTMLAttributes<HTMLDivElement>) => (
73:   <div
74:     className={cn("mt-auto flex flex-col gap-2 p-4", className)}
75:     {...props}
76:   />
77: )
78: DrawerFooter.displayName = "DrawerFooter"
79:
80: const DrawerTitle = React.forwardRef<
81:   React.ElementRef<typeof DrawerPrimitive.Title>,
82:   React.ComponentPropsWithoutRef<typeof DrawerPrimitive.Title>
83: >(({ className, ...props }, ref) => (

```

```

84:   <DrawerPrimitive.Title
85:     ref={ref}
86:     className={cn(
87:       "text-lg font-semibold leading-none tracking-tight",
88:       className
89:     )}
90:     {...props}
91:   />
92: ))
93: DrawerTitle.displayName = DrawerPrimitive.Title.displayName
94:
95: const DrawerDescription = React.forwardRef<
96:   React.ElementRef<typeof DrawerPrimitive.Description>,
97:   React.ComponentPropsWithoutRef<typeof DrawerPrimitive.Description>
98: >(({ className, ...props }, ref) => (
99:   <DrawerPrimitive.Description
100:     ref={ref}
101:     className={cn("text-sm text-muted-foreground", className)}
102:     {...props}
103:   />
104: ))
105: DrawerDescription.displayName = DrawerPrimitive.Description.displayName
106:
107: export {
108:   Drawer,
109:   DrawerPortal,
110:   DrawerOverlay,
111:   DrawerTrigger,
112:   DrawerClose,
113:   DrawerContent,
114:   DrawerHeader,
115:   DrawerFooter,
116:   DrawerTitle,
117:   DrawerDescription,
118: }

```

## ■ File: components/ui/dropdown-menu.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import * as DropdownMenuPrimitive from "@radix-ui/react-dropdown-menu"
5: import { Check, ChevronRight, Circle } from "lucide-react"
6:
7: import { cn } from "@/lib/utils"
8:
9: const DropdownMenu = DropdownMenuPrimitive.Root
10:
11: const DropdownMenuTrigger = DropdownMenuPrimitive.Trigger
12:
13: const DropdownMenuGroup = DropdownMenuPrimitive.Group
14:
15: const DropdownMenuPortal = DropdownMenuPrimitive.Portal
16:
17: const DropdownMenuSub = DropdownMenuPrimitive.Sub
18:
19: const DropdownMenuRadioGroup = DropdownMenuPrimitive.RadioGroup
20:
21: const DropdownMenuSubTrigger = React.forwardRef<
22:   React.ElementRef<typeof DropdownMenuPrimitive.SubTrigger>,
23:   React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.SubTrigger> & {
24:     inset?: boolean
25:   }
26: >(({ className, inset, children, ...props }, ref) => (
27:   <DropdownMenuPrimitive.SubTrigger
28:     ref={ref}
29:     className={cn(
30:       "flex cursor-default gap-2 select-none items-center rounded-sm px-2 py-1.5 text-sm outline-none focus
31:       inset && "pl-8",
32:       className
33:     )}

```

```

34:     {...props}
35:   >
36:     {children}
37:     <ChevronRight className="ml-auto" />
38:   </DropdownMenuPrimitive.SubTrigger>
39: ))
40: DropdownMenuSubTrigger.displayName =
41:   DropdownMenuPrimitive.SubTrigger.displayName
42:
43: const DropdownMenuSubContent = React.forwardRef<
44:   React.ElementRef<typeof DropdownMenuPrimitive.SubContent>,
45:   React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.SubContent>
46: >(({ className, ...props }, ref) => (
47:   <DropdownMenuPrimitive.SubContent
48:     ref={ref}
49:     className={cn(
50:       "z-50 min-w-[8rem] overflow-hidden rounded-md border bg-popover p-1 text-popover-foreground shadow-lg
51:       className
52:     )}
53:     {...props}
54:   />
55: ))
56: DropdownMenuSubContent.displayName =
57:   DropdownMenuPrimitive.SubContent.displayName
58:
59: const DropdownMenuContent = React.forwardRef<
60:   React.ElementRef<typeof DropdownMenuPrimitive.Content>,
61:   React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.Content>
62: >(({ className, sideOffset = 4, ...props }, ref) => (
63:   <DropdownMenuPrimitive.Portal>
64:     <DropdownMenuPrimitive.Content
65:       ref={ref}
66:       sideOffset={sideOffset}
67:       className={cn(
68:         "z-50 min-w-[8rem] overflow-hidden rounded-md border bg-popover p-1 text-popover-foreground shadow-
69:         className
70:       )}
71:       {...props}
72:     />
73:   </DropdownMenuPrimitive.Portal>
74: ))
75: DropdownMenuContent.displayName = DropdownMenuPrimitive.Content.displayName
76:
77: const DropdownMenuItem = React.forwardRef<
78:   React.ElementRef<typeof DropdownMenuPrimitive.Item>,
79:   React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.Item> & {
80:     inset?: boolean
81:   }
82: >(({ className, inset, ...props }, ref) => (
83:   <DropdownMenuPrimitive.Item
84:     ref={ref}
85:     className={cn(
86:       "relative flex cursor-default select-none items-center gap-2 rounded-sm px-2 py-1.5 text-sm outline-n
87:       inset && "pl-8",
88:       className
89:     )}
90:     {...props}
91:   />
92: ))
93: DropdownMenuItem.displayName = DropdownMenuPrimitive.Item.displayName
94:
95: const DropdownMenuCheckboxItem = React.forwardRef<
96:   React.ElementRef<typeof DropdownMenuPrimitive.CheckboxItem>,
97:   React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.CheckboxItem>
98: >(({ className, children, checked, ...props }, ref) => (
99:   <DropdownMenuPrimitive.CheckboxItem
100:     ref={ref}
101:     className={cn(
102:       "relative flex cursor-default select-none items-center rounded-sm py-1.5 pl-8 pr-2 text-sm outline-no
103:       className
104:     )}
105:     checked={checked}
106:     {...props}

```

```

107:   >
108:   <span className="absolute left-2 flex h-3.5 w-3.5 items-center justify-center">
109:     <DropdownMenuPrimitive.ItemIndicator>
110:       <Check className="h-4 w-4" />
111:     </DropdownMenuPrimitive.ItemIndicator>
112:   </span>
113:   {children}
114: </DropdownMenuPrimitive.CheckboxItem>
115: ))
116: DropdownMenuCheckboxItem.displayName =
117:   DropdownMenuPrimitive.CheckboxItem.displayName
118:
119: const DropdownMenuRadioItem = React.forwardRef<
120:   React.ElementRef<typeof DropdownMenuPrimitive.RadioItem>,
121:   React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.RadioItem>
122: >(({ className, children, ...props }, ref) => (
123:   <DropdownMenuPrimitive.RadioItem
124:     ref={ref}
125:     className={cn(
126:       "relative flex cursor-default select-none items-center rounded-sm py-1.5 pl-8 pr-2 text-sm outline-none",
127:       className
128:     )}
129:     {...props}
130:   >
131:     <span className="absolute left-2 flex h-3.5 w-3.5 items-center justify-center">
132:       <DropdownMenuPrimitive.ItemIndicator>
133:         <Circle className="h-2 w-2 fill-current" />
134:       </DropdownMenuPrimitive.ItemIndicator>
135:     </span>
136:     {children}
137:   </DropdownMenuPrimitive.RadioItem>
138: ))
139: DropdownMenuRadioItem.displayName = DropdownMenuPrimitive.RadioItem.displayName
140:
141: const DropdownMenuLabel = React.forwardRef<
142:   React.ElementRef<typeof DropdownMenuPrimitive.Label>,
143:   React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.Label> & {
144:     inset?: boolean
145:   }
146: >(({ className, inset, ...props }, ref) => (
147:   <DropdownMenuPrimitive.Label
148:     ref={ref}
149:     className={cn(
150:       "px-2 py-1.5 text-sm font-semibold",
151:       inset && "pl-8",
152:       className
153:     )}
154:     {...props}
155:   />
156: ))
157: DropdownMenuLabel.displayName = DropdownMenuPrimitive.Label.displayName
158:
159: const DropdownMenuSeparator = React.forwardRef<
160:   React.ElementRef<typeof DropdownMenuPrimitive.Separator>,
161:   React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.Separator>
162: >(({ className, ...props }, ref) => (
163:   <DropdownMenuPrimitive.Separator
164:     ref={ref}
165:     className={cn("-mx-1 my-1 h-px bg-muted", className)}
166:     {...props}
167:   />
168: ))
169: DropdownMenuSeparator.displayName = DropdownMenuPrimitive.Separator.displayName
170:
171: const DropdownMenuShortcut = ({
172:   className,
173:   ...props
174: }: React.HTMLAttributes<HTMLSpanElement>) => {
175:   return (
176:     <span
177:       className={cn("ml-auto text-xs tracking-widest opacity-60", className)}
178:       {...props}
179:     />

```



```

180:   )
181: }
182: DropdownMenuShortcut.displayName = "DropdownMenuShortcut"
183:
184: export {
185:   DropdownMenu,
186:   DropdownMenuTrigger,
187:   DropdownMenuContent,
188:   DropdownMenuItem,
189:   DropdownMenuCheckboxItem,
190:   DropdownMenuRadioItem,
191:   DropdownMenuLabel,
192:   DropdownMenuSeparator,
193:   DropdownMenuShortcut,
194:   DropdownMenuGroup,
195:   DropdownMenuPortal,
196:   DropdownMenuSub,
197:   DropdownMenuSubContent,
198:   DropdownMenuSubTrigger,
199:   DropdownMenuRadioGroup,
200: }

```

## File: components\ui\form.tsx

```

1: "use client"
2:
3: import * as React from "react"
4: import * as LabelPrimitive from "@radix-ui/react-label"
5: import { Slot } from "@radix-ui/react-slot"
6: import {
7:   Controller,
8:   ControllerProps,
9:   FieldPath,
10:  FieldValues,
11:  FormProvider,
12:  useFormContext,
13: } from "react-hook-form"
14:
15: import { cn } from "@/lib/Utils"
16: import { Label } from "@/components/ui/label"
17:
18: const Form = FormProvider
19:
20: type FormFieldContextValue<
21:   TFieldValues extends FieldValues = FieldValues,
22:   TName extends FieldPath<TFieldValues> = FieldPath<TFieldValues>
23: > = {
24:   name: TName
25: }
26:
27: const FormFieldContext = React.createContext<FormFieldContextValue>(
28:   {} as FormFieldContextValue
29: )
30:
31: const FormField = <
32:   TFieldValues extends FieldValues = FieldValues,
33:   TName extends FieldPath<TFieldValues> = FieldPath<TFieldValues>
34: >({
35:   ...props
36: }): ControllerProps<TFieldValues, TName> => {
37:   return (
38:     <FormFieldContext.Provider value={{ name: props.name }}>
39:       <Controller {...props} />
40:     </FormFieldContext.Provider>
41:   )
42: }
43:
44: const useFormField = () => {
45:   const fieldContext = React.useContext(FormFieldContext)
46:   const itemContext = React.useContext(FormItemContext)
47:   const { getFieldState, formState } = useFormContext()

```

```

48:
49:   const fieldState = getFieldState(fieldContext.name, formState)
50:
51:   if (!fieldContext) {
52:     throw new Error("useFormField should be used within <FormField>")
53:   }
54:
55:   const { id } = itemContext
56:
57:   return {
58:     id,
59:     name: fieldContext.name,
60:     formItemId: `${id}-form-item`,
61:     formDescriptionId: `${id}-form-item-description`,
62:     formMessageId: `${id}-form-item-message`,
63:     ...fieldState,
64:   }
65: }
66:
67: type FormItemContextValue = {
68:   id: string
69: }
70:
71: const FormItemContext = React.createContext<FormItemContextValue>(
72:   {} as FormItemContextValue
73: )
74:
75: const FormItem = React.forwardRef<
76:   HTMLDivElement,
77:   React.HTMLAttributes<HTMLDivElement>
78: >(({ className, ...props }, ref) => {
79:   const id = React.useId()
80:
81:   return (
82:     <FormItemContext.Provider value={{ id }}>
83:       <div ref={ref} className={cn("space-y-2", className)} {...props} />
84:     </FormItemContext.Provider>
85:   )
86: })
87: FormItem.displayName = "FormItem"
88:
89: const FormLabel = React.forwardRef<
90:   React.ElementRef<typeof LabelPrimitive.Root>,
91:   React.ComponentPropsWithoutRef<typeof LabelPrimitive.Root>
92: >(({ className, ...props }, ref) => {
93:   const { error, formItemId } = useFormField()
94:
95:   return (
96:     <Label
97:       ref={ref}
98:       className={cn(error && "text-destructive", className)}
99:       htmlFor={formItemId}
100:       {...props}
101:     />
102:   )
103: })
104: FormLabel.displayName = "FormLabel"
105:
106: const FormControl = React.forwardRef<
107:   React.ElementRef<typeof Slot>,
108:   React.ComponentPropsWithoutRef<typeof Slot>
109: >(({ ...props }, ref) => {
110:   const { error, formItemId, formDescriptionId, formMessageId } = useFormField()
111:
112:   return (
113:     <Slot
114:       ref={ref}
115:       id={formItemId}
116:       aria-describedby={
117:         !error
118:           ? `${formDescriptionId}`
119:           : `${formDescriptionId} ${formMessageId}`
120:       }

```

```

121:         aria-invalid={!error}
122:         {...props}
123:     />
124: )
125: })
126: FormControl.displayName = "FormControl"
127:
128: const FormDescription = React.forwardRef<
129:   HTMLParagraphElement,
130:   React.HTMLAttributes<HTMLParagraphElement>
131: >(({ className, ...props }, ref) => {
132:   const { formDescriptionId } = useFormField()
133:
134:   return (
135:     <p
136:       ref={ref}
137:       id={formDescriptionId}
138:       className={cn("text-sm text-muted-foreground", className)}
139:       {...props}
140:     />
141:   )
142: })
143: FormDescription.displayName = "FormDescription"
144:
145: const FormMessage = React.forwardRef<
146:   HTMLParagraphElement,
147:   React.HTMLAttributes<HTMLParagraphElement>
148: >(({ className, children, ...props }, ref) => {
149:   const { error, formMessageId } = useFormField()
150:   const body = error ? String(error?.message) : children
151:
152:   if (!body) {
153:     return null
154:   }
155:
156:   return (
157:     <p
158:       ref={ref}
159:       id={formMessageId}
160:       className={cn("text-sm font-medium text-destructive", className)}
161:       {...props}
162:     >
163:       {body}
164:     </p>
165:   )
166: })
167: FormMessage.displayName = "FormMessage"
168:
169: export {
170:   useFormField,
171:   Form,
172:   FormItem,
173:   FormLabel,
174:   FormControl,
175:   FormDescription,
176:   FormMessage,
177:   FormField,
178: }

```

## ■ File: components/ui/hover-card.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import * as HoverCardPrimitive from "@radix-ui/react-hover-card"
5:
6: import { cn } from "@/lib/utils"
7:
8: const HoverCard = HoverCardPrimitive.Root
9:
10: const HoverCardTrigger = HoverCardPrimitive.Trigger

```

```

11:
12: const HoverCardContent = React.forwardRef<
13:   React.ElementRef<typeof HoverCardPrimitive.Content>,
14:   React.ComponentPropsWithoutRef<typeof HoverCardPrimitive.Content>
15: >(({ className, align = "center", sideOffset = 4, ...props }, ref) => (
16:   <HoverCardPrimitive.Content
17:     ref={ref}
18:     align={align}
19:     sideOffset={sideOffset}
20:     className={cn(
21:       "z-50 w-64 rounded-md border bg-popover p-4 text-popover-foreground shadow-md outline-none data-[state=open]:animate-in data-[state=closed]:animate-out data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0",
22:       className
23:     )}
24:     {...props}
25:   />
26: ))
27: HoverCardContent.displayName = HoverCardPrimitive.Content.displayName
28:
29: export { HoverCard, HoverCardTrigger, HoverCardContent }

```

## ■ File: components\ui\input-otp.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import { OTPInput, OTPInputContext } from "input-otp"
5: import { Dot } from "lucide-react"
6:
7: import { cn } from "@/lib/utils"
8:
9: const InputOTP = React.forwardRef<
10:   React.ElementRef<typeof OTPInput>,
11:   React.ComponentPropsWithoutRef<typeof OTPInput>
12: >(({ className, containerClassName, ...props }, ref) => (
13:   <OTPInput
14:     ref={ref}
15:     containerClassName={cn(
16:       "flex items-center gap-2 has-[:disabled]:opacity-50",
17:       containerClassName
18:     )}
19:     className={cn("disabled:cursor-not-allowed", className)}
20:     {...props}
21:   />
22: ))
23: InputOTP.displayName = "InputOTP"
24:
25: const InputOTPGroup = React.forwardRef<
26:   React.ElementRef<"div">,
27:   React.ComponentPropsWithoutRef<"div">
28: >(({ className, ...props }, ref) => (
29:   <div ref={ref} className={cn("flex items-center", className)} {...props} />
30: ))
31: InputOTPGroup.displayName = "InputOTPGroup"
32:
33: const InputOTPSlot = React.forwardRef<
34:   React.ElementRef<"div">,
35:   React.ComponentPropsWithoutRef<"div"> & { index: number }
36: >(({ index, className, ...props }, ref) => {
37:   const inputOTPContext = React.useContext(OTPInputContext)
38:   const { char, hasFakeCaret, isActive } = inputOTPContext.slots[index]
39:
40:   return (
41:     <div
42:       ref={ref}
43:       className={cn(
44:         "relative flex h-10 w-10 items-center justify-center border-y border-r border-input text-sm transition-colors",
45:         isActive && "z-10 ring-2 ring-ring ring-offset-background",
46:         className
47:       )}
48:       {...props}
49:     >

```

```

50:         {char}
51:         {hasFakeCaret && (
52:             <div className="pointer-events-none absolute inset-0 flex items-center justify-center">
53:                 <div className="h-4 w-px animate-caret-blink bg-foreground duration-1000" />
54:             </div>
55:         )}
56:     </div>
57: )
58: })
59: InputOTPSlot.displayName = "InputOTPSlot"
60:
61: const InputOTPSeparator = React.forwardRef<
62:   React.ElementRef<"div">,
63:   React.ComponentPropsWithoutRef<"div">
64: >(({ ...props }, ref) => (
65:   <div ref={ref} role="separator" {...props}>
66:     <Dot />
67:   </div>
68: ))
69: InputOTPSeparator.displayName = "InputOTPSeparator"
70:
71: export { InputOTP, InputOTPGroup, InputOTPSlot, InputOTPSeparator }

```

## ■ File: components\ui\input.tsx

```

1: import * as React from "react"
2:
3: import { cn } from "@/lib/utils"
4:
5: const Input = React.forwardRef<HTMLInputElement, React.ComponentProps<"input">>(
6:   ({ className, type, ...props }, ref) => {
7:     return (
8:       <input
9:         type={type}
10:        className={cn(
11:          "flex h-10 w-full rounded-md border border-input bg-background px-3 py-2 text-base ring-offset-background",
12:          className
13:        )}
14:        ref={ref}
15:        {...props}
16:      />
17:    )
18:   }
19: )
20: Input.displayName = "Input"
21:
22: export { Input }

```

## ■ File: components\ui\label.tsx

```

1: "use client"
2:
3: import * as React from "react"
4: import * as LabelPrimitive from "@radix-ui/react-label"
5: import { cva, type VariantProps } from "class-variance-authority"
6:
7: import { cn } from "@/lib/utils"
8:
9: const labelVariants = cva(
10:   "text-sm font-medium leading-none peer-disabled:cursor-not-allowed peer-disabled:opacity-70"
11: )
12:
13: const Label = React.forwardRef<
14:   React.ElementRef<typeof LabelPrimitive.Root>,
15:   React.ComponentPropsWithoutRef<typeof LabelPrimitive.Root> &
16:   VariantProps<typeof labelVariants>
17: >(({ className, ...props }, ref) => (
18:   <LabelPrimitive.Root
19:     ref={ref}

```

```

20:     className={cn(labelVariants(), className)}
21:     {...props}
22:   />
23: ))
24: Label.displayName = LabelPrimitive.Root.displayName
25:
26: export { Label }

```

## ■ File: components\ui\menubar.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import * as MenubarPrimitive from "@radix-ui/react-menubar"
5: import { Check, ChevronRight, Circle } from "lucide-react"
6:
7: import { cn } from "@/lib/utils"
8:
9: const MenubarMenu = MenubarPrimitive.Menu
10:
11: const MenubarGroup = MenubarPrimitive.Group
12:
13: const MenubarPortal = MenubarPrimitive.Portal
14:
15: const MenubarSub = MenubarPrimitive.Sub
16:
17: const MenubarRadioGroup = MenubarPrimitive.RadioGroup
18:
19: const Menubar = React.forwardRef<
20:   React.ElementRef<typeof MenubarPrimitive.Root>,
21:   React.ComponentPropsWithoutRef<typeof MenubarPrimitive.Root>
22: >(({ className, ...props }, ref) => (
23:   <MenubarPrimitive.Root
24:     ref={ref}
25:     className={cn(
26:       "flex h-10 items-center space-x-1 rounded-md border bg-background p-1",
27:       className
28:     )}
29:     {...props}
30:   />
31: ))
32: Menubar.displayName = MenubarPrimitive.Root.displayName
33:
34: const MenubarTrigger = React.forwardRef<
35:   React.ElementRef<typeof MenubarPrimitive.Trigger>,
36:   React.ComponentPropsWithoutRef<typeof MenubarPrimitive.Trigger>
37: >(({ className, ...props }, ref) => (
38:   <MenubarPrimitive.Trigger
39:     ref={ref}
40:     className={cn(
41:       "flex cursor-default select-none items-center rounded-sm px-3 py-1.5 text-sm font-medium outline-none",
42:       className
43:     )}
44:     {...props}
45:   />
46: ))
47: MenubarTrigger.displayName = MenubarPrimitive.Trigger.displayName
48:
49: const MenubarSubTrigger = React.forwardRef<
50:   React.ElementRef<typeof MenubarPrimitive.SubTrigger>,
51:   React.ComponentPropsWithoutRef<typeof MenubarPrimitive.SubTrigger> & {
52:     inset?: boolean
53:   }
54: >(({ className, inset, children, ...props }, ref) => (
55:   <MenubarPrimitive.SubTrigger
56:     ref={ref}
57:     className={cn(
58:       "flex cursor-default select-none items-center rounded-sm px-2 py-1.5 text-sm outline-none focus:bg-accent",
59:       inset && "pl-8",
60:       className
61:     )}

```

```

62:     {...props}
63:   >
64:     {children}
65:     <ChevronRight className="ml-auto h-4 w-4" />
66:   </MenubarPrimitive.SubTrigger>
67: ))
68: MenubarSubTrigger.displayName = MenubarPrimitive.SubTrigger.displayName
69:
70: const MenubarSubContent = React.forwardRef<
71:   React.ElementRef<typeof MenubarPrimitive.SubContent>,
72:   React.ComponentPropsWithoutRef<typeof MenubarPrimitive.SubContent>
73: >(({ className, ...props }, ref) => (
74:   <MenubarPrimitive.SubContent
75:     ref={ref}
76:     className={cn(
77:       "z-50 min-w-[8rem] overflow-hidden rounded-md border bg-popover p-1 text-popover-foreground data-[state=open]:animate-in data-[state=closed]:animate-out data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0",
78:       className
79:     )}
80:     {...props}
81:   />
82: ))
83: MenubarSubContent.displayName = MenubarPrimitive.SubContent.displayName
84:
85: const MenubarContent = React.forwardRef<
86:   React.ElementRef<typeof MenubarPrimitive.Content>,
87:   React.ComponentPropsWithoutRef<typeof MenubarPrimitive.Content>
88: >((
89:   (
90:     { className, align = "start", alignOffset = -4, sideOffset = 8, ...props },
91:     ref
92:   ) => (
93:     <MenubarPrimitive.Portal>
94:       <MenubarPrimitive.Content
95:         ref={ref}
96:         align={align}
97:         alignOffset={alignOffset}
98:         sideOffset={sideOffset}
99:         className={cn(
100:           "z-50 min-w-[12rem] overflow-hidden rounded-md border bg-popover p-1 text-popover-foreground shadow-lg",
101:           className
102:         )}
103:         {...props}
104:       />
105:     </MenubarPrimitive.Portal>
106:   )
107: ))
108: MenubarContent.displayName = MenubarPrimitive.Content.displayName
109:
110: const MenubarItem = React.forwardRef<
111:   React.ElementRef<typeof MenubarPrimitive.Item>,
112:   React.ComponentPropsWithoutRef<typeof MenubarPrimitive.Item> & {
113:     inset?: boolean
114:   }
115: >(({ className, inset, ...props }, ref) => (
116:   <MenubarPrimitive.Item
117:     ref={ref}
118:     className={cn(
119:       "relative flex cursor-default select-none items-center rounded-sm px-2 py-1.5 text-sm outline-none focus:bg-accent focus:text-accent-foreground",
120:       inset && "pl-8",
121:       className
122:     )}
123:     {...props}
124:   />
125: ))
126: MenubarItem.displayName = MenubarPrimitive.Item.displayName
127:
128: const MenubarCheckboxItem = React.forwardRef<
129:   React.ElementRef<typeof MenubarPrimitive.CheckboxItem>,
130:   React.ComponentPropsWithoutRef<typeof MenubarPrimitive.CheckboxItem>
131: >(({ className, children, checked, ...props }, ref) => (
132:   <MenubarPrimitive.CheckboxItem
133:     ref={ref}
134:     className={cn(

```

```

135:     "relative flex cursor-default select-none items-center rounded-sm py-1.5 pl-8 pr-2 text-sm outline-no
136:     className
137:   )}
138:   checked={checked}
139:   {...props}
140: >
141:   <span className="absolute left-2 flex h-3.5 w-3.5 items-center justify-center">
142:     <MenubarPrimitive.ItemIndicator>
143:       <Check className="h-4 w-4" />
144:     </MenubarPrimitive.ItemIndicator>
145:   </span>
146:   {children}
147: </MenubarPrimitive.CheckboxItem>
148: ))
149: MenubarCheckboxItem.displayName = MenubarPrimitive.CheckboxItem.displayName
150:
151: const MenubarRadioItem = React.forwardRef<
152:   React.ElementRef<typeof MenubarPrimitive.RadioItem>,
153:   React.ComponentPropsWithoutRef<typeof MenubarPrimitive.RadioItem>
154: >(({ className, children, ...props }, ref) => (
155:   <MenubarPrimitive.RadioItem
156:     ref={ref}
157:     className={cn(
158:       "relative flex cursor-default select-none items-center rounded-sm py-1.5 pl-8 pr-2 text-sm outline-no
159:       className
160:     )}
161:     {...props}
162:   >
163:     <span className="absolute left-2 flex h-3.5 w-3.5 items-center justify-center">
164:       <MenubarPrimitive.ItemIndicator>
165:         <Circle className="h-2 w-2 fill-current" />
166:       </MenubarPrimitive.ItemIndicator>
167:     </span>
168:     {children}
169:   </MenubarPrimitive.RadioItem>
170: ))
171: MenubarRadioItem.displayName = MenubarPrimitive.RadioItem.displayName
172:
173: const MenubarLabel = React.forwardRef<
174:   React.ElementRef<typeof MenubarPrimitive.Label>,
175:   React.ComponentPropsWithoutRef<typeof MenubarPrimitive.Label> & {
176:     inset?: boolean
177:   }
178: >(({ className, inset, ...props }, ref) => (
179:   <MenubarPrimitive.Label
180:     ref={ref}
181:     className={cn(
182:       "px-2 py-1.5 text-sm font-semibold",
183:       inset && "pl-8",
184:       className
185:     )}
186:     {...props}
187:   />
188: ))
189: MenubarLabel.displayName = MenubarPrimitive.Label.displayName
190:
191: const MenubarSeparator = React.forwardRef<
192:   React.ElementRef<typeof MenubarPrimitive.Separator>,
193:   React.ComponentPropsWithoutRef<typeof MenubarPrimitive.Separator>
194: >(({ className, ...props }, ref) => (
195:   <MenubarPrimitive.Separator
196:     ref={ref}
197:     className={cn("-mx-1 my-1 h-px bg-muted", className)}
198:     {...props}
199:   />
200: ))
201: MenubarSeparator.displayName = MenubarPrimitive.Separator.displayName
202:
203: const MenubarShortcut = ({
204:   className,
205:   ...props
206: }: React.HTMLAttributes<HTMLSpanElement>) => {
207:   return (

```



```

208:     <span
209:       className={cn(
210:         "ml-auto text-xs tracking-widest text-muted-foreground",
211:         className
212:       )}
213:       {...props}
214:     />
215:   )
216: }
217: MenubarShortcut.displayName = "MenubarShortcut"
218:
219: export {
220:   Menubar,
221:   MenubarMenu,
222:   MenubarTrigger,
223:   MenubarContent,
224:   MenubarItem,
225:   MenubarSeparator,
226:   MenubarLabel,
227:   MenubarCheckboxItem,
228:   MenubarRadioGroup,
229:   MenubarRadioItem,
230:   MenubarPortal,
231:   MenubarSubContent,
232:   MenubarSubTrigger,
233:   MenubarGroup,
234:   MenubarSub,
235:   MenubarShortcut,
236: }

```

## ■ File: components/ui/navigation-menu.tsx

```

=====
1: import * as React from "react"
2: import * as NavigationMenuPrimitive from "@radix-ui/react-navigation-menu"
3: import { cva } from "class-variance-authority"
4: import { ChevronDown } from "lucide-react"
5:
6: import { cn } from "@lib/utils"
7:
8: const NavigationMenu = React.forwardRef<
9:   React.ElementRef<typeof NavigationMenuPrimitive.Root>,
10:   React.ComponentPropsWithoutRef<typeof NavigationMenuPrimitive.Root>
11: >(({ className, children, ...props }, ref) => (
12:   <NavigationMenuPrimitive.Root
13:     ref={ref}
14:     className={cn(
15:       "relative z-10 flex max-w-max flex-1 items-center justify-center",
16:       className
17:     )}
18:     {...props}
19:   >
20:     {children}
21:     <NavigationMenuViewport />
22:   </NavigationMenuPrimitive.Root>
23: ))
24: NavigationMenu.displayName = NavigationMenuPrimitive.Root.displayName
25:
26: const NavigationMenuList = React.forwardRef<
27:   React.ElementRef<typeof NavigationMenuPrimitive.List>,
28:   React.ComponentPropsWithoutRef<typeof NavigationMenuPrimitive.List>
29: >(({ className, ...props }, ref) => (
30:   <NavigationMenuPrimitive.List
31:     ref={ref}
32:     className={cn(
33:       "group flex flex-1 list-none items-center justify-center space-x-1",
34:       className
35:     )}
36:     {...props}
37:   />
38: ))
39: NavigationMenuList.displayName = NavigationMenuPrimitive.List.displayName

```

```

40:
41: const NavigationMenuItem = NavigationMenuPrimitive.Item
42:
43: const navigationMenuTriggerStyle = cva(
44:   "group inline-flex h-10 w-max items-center justify-center rounded-md bg-background px-4 py-2 text-sm font
45: )
46:
47: const NavigationMenuTrigger = React.forwardRef<
48:   React.ElementRef<typeof NavigationMenuPrimitive.Trigger>,
49:   React.ComponentPropsWithoutRef<typeof NavigationMenuPrimitive.Trigger>
50: >(({ className, children, ...props }, ref) => (
51:   <NavigationMenuPrimitive.Trigger
52:     ref={ref}
53:     className={cn(navigationMenuTriggerStyle(), "group", className)}
54:     {...props}
55:   >
56:     {children}{" "}
57:     <ChevronDown
58:       className="relative top-[1px] ml-1 h-3 w-3 transition duration-200 group-data-[state=open]:rotate-180
59:       aria-hidden="true"
60:     />
61:   </NavigationMenuPrimitive.Trigger>
62: ))
63: NavigationMenuTrigger.displayName = NavigationMenuPrimitive.Trigger.displayName
64:
65: const NavigationMenuContent = React.forwardRef<
66:   React.ElementRef<typeof NavigationMenuPrimitive.Content>,
67:   React.ComponentPropsWithoutRef<typeof NavigationMenuPrimitive.Content>
68: >(({ className, ...props }, ref) => (
69:   <NavigationMenuPrimitive.Content
70:     ref={ref}
71:     className={cn(
72:       "left-0 top-0 w-full data-[motion^=from-]:animate-in data-[motion^=to-]:animate-out data-[motion^=from-]
73:       className
74:     )}
75:     {...props}
76:   />
77: ))
78: NavigationMenuContent.displayName = NavigationMenuPrimitive.Content.displayName
79:
80: const NavigationMenuLink = NavigationMenuPrimitive.Link
81:
82: const NavigationMenuViewport = React.forwardRef<
83:   React.ElementRef<typeof NavigationMenuPrimitive.Viewport>,
84:   React.ComponentPropsWithoutRef<typeof NavigationMenuPrimitive.Viewport>
85: >(({ className, ...props }, ref) => (
86:   <div className={cn("absolute left-0 top-full flex justify-center")}>
87:     <NavigationMenuPrimitive.Viewport
88:       className={cn(
89:         "origin-top-center relative mt-1.5 h-[var(--radix-navigation-menu-viewport-height)] w-full overflow
90:         className
91:       )}
92:       ref={ref}
93:       {...props}
94:     />
95:   </div>
96: ))
97: NavigationMenuViewport.displayName =
98:   NavigationMenuPrimitive.Viewport.displayName
99:
100: const NavigationMenuIndicator = React.forwardRef<
101:   React.ElementRef<typeof NavigationMenuPrimitive.Indicator>,
102:   React.ComponentPropsWithoutRef<typeof NavigationMenuPrimitive.Indicator>
103: >(({ className, ...props }, ref) => (
104:   <NavigationMenuPrimitive.Indicator
105:     ref={ref}
106:     className={cn(
107:       "top-full z-[1] flex h-1.5 items-end justify-center overflow-hidden data-[state=visible]:animate-in d
108:       className
109:     )}
110:     {...props}
111:   >
112:     <div className="relative top-[60%] h-2 w-2 rotate-45 rounded-tl-sm bg-border shadow-md" />

```

```

113:   </NavigationMenuPrimitive.Indicator>
114: ))
115: NavigationMenuIndicator.displayName =
116:   NavigationMenuPrimitive.Indicator.displayName
117:
118: export {
119:   navigationMenuTriggerStyle,
120:   NavigationMenu,
121:   NavigationMenuList,
122:   NavigationMenuItem,
123:   NavigationMenuContent,
124:   NavigationMenuTrigger,
125:   NavigationMenuLink,
126:   NavigationMenuIndicator,
127:   NavigationMenuViewport,
128: }

```

## ■ File: components\ui\pagination.tsx

```

=====
1: import * as React from "react"
2: import { ChevronLeft, ChevronRight, MoreHorizontal } from "lucide-react"
3:
4: import { cn } from "@lib/utils"
5: import { ButtonProps, buttonVariants } from "@components/ui/button"
6:
7: const Pagination = ({ className, ...props }: React.ComponentProps<"nav">) => (
8:   <nav
9:     role="navigation"
10:    aria-label="pagination"
11:    className={cn("mx-auto flex w-full justify-center", className)}
12:    {...props}
13:  />
14: )
15: Pagination.displayName = "Pagination"
16:
17: const PaginationContent = React.forwardRef<
18:   HTMLUListElement,
19:   React.ComponentProps<"ul">
20: >(({ className, ...props }, ref) => (
21:   <ul
22:     ref={ref}
23:     className={cn("flex flex-row items-center gap-1", className)}
24:     {...props}
25:   />
26: ))
27: PaginationContent.displayName = "PaginationContent"
28:
29: const PaginationItem = React.forwardRef<
30:   HTMLLIElement,
31:   React.ComponentProps<"li">
32: >(({ className, ...props }, ref) => (
33:   <li ref={ref} className={cn("", className)} {...props} />
34: ))
35: PaginationItem.displayName = "PaginationItem"
36:
37: type PaginationLinkProps = {
38:   isActive?: boolean
39: } & Pick<ButtonProps, "size"> &
40:   React.ComponentProps<"a">
41:
42: const PaginationLink = ({
43:   className,
44:   isActive,
45:   size = "icon",
46:   ...props
47: }: PaginationLinkProps) => (
48:   <a
49:     aria-current={isActive ? "page" : undefined}
50:     className={cn(
51:       buttonVariants({
52:         variant: isActive ? "outline" : "ghost",

```

```

53:         size,
54:     }},
55:     className
56:   )}
57:   {...props}
58: />
59: )
60: PaginationLink.displayName = "PaginationLink"
61:
62: const PaginationPrevious = ({
63:   className,
64:   ...props
65: }: React.ComponentProps<typeof PaginationLink>) => (
66:   <PaginationLink
67:     aria-label="Go to previous page"
68:     size="default"
69:     className={cn("gap-1 pl-2.5", className)}
70:     {...props}
71:   >
72:     <ChevronLeft className="h-4 w-4" />
73:     <span>Previous</span>
74:   </PaginationLink>
75: )
76: PaginationPrevious.displayName = "PaginationPrevious"
77:
78: const PaginationNext = ({
79:   className,
80:   ...props
81: }: React.ComponentProps<typeof PaginationLink>) => (
82:   <PaginationLink
83:     aria-label="Go to next page"
84:     size="default"
85:     className={cn("gap-1 pr-2.5", className)}
86:     {...props}
87:   >
88:     <span>Next</span>
89:     <ChevronRight className="h-4 w-4" />
90:   </PaginationLink>
91: )
92: PaginationNext.displayName = "PaginationNext"
93:
94: const PaginationEllipsis = ({
95:   className,
96:   ...props
97: }: React.ComponentProps<"span">) => (
98:   <span
99:     aria-hidden
100:     className={cn("flex h-9 w-9 items-center justify-center", className)}
101:     {...props}
102:   >
103:     <MoreHorizontal className="h-4 w-4" />
104:     <span className="sr-only">More pages</span>
105:   </span>
106: )
107: PaginationEllipsis.displayName = "PaginationEllipsis"
108:
109: export {
110:   Pagination,
111:   PaginationContent,
112:   PaginationEllipsis,
113:   PaginationItem,
114:   PaginationLink,
115:   PaginationNext,
116:   PaginationPrevious,
117: }

```

## ■ File: components/ui/popover.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"

```

```

4: import * as PopoverPrimitive from "@radix-ui/react-popover"
5:
6: import { cn } from "@lib/utils"
7:
8: const Popover = PopoverPrimitive.Root
9:
10: const PopoverTrigger = PopoverPrimitive.Trigger
11:
12: const PopoverContent = React.forwardRef<
13:   React.ElementRef<typeof PopoverPrimitive.Content>,
14:   React.ComponentPropsWithoutRef<typeof PopoverPrimitive.Content>
15: >(({ className, align = "center", sideOffset = 4, ...props }, ref) => (
16:   <PopoverPrimitive.Portal>
17:     <PopoverPrimitive.Content
18:       ref={ref}
19:       align={align}
20:       sideOffset={sideOffset}
21:       className={cn(
22:         "z-50 w-72 rounded-md border bg-popover p-4 text-popover-foreground shadow-md outline-none data-[state=open]:animate-in data-[state=closed]:animate-out data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0",
23:         className
24:       )}
25:       {...props}
26:     />
27:   </PopoverPrimitive.Portal>
28: ))
29: PopoverContent.displayName = PopoverPrimitive.Content.displayName
30:
31: export { Popover, PopoverTrigger, PopoverContent }

```

## ■ File: components/ui/progress.tsx

```

1: "use client"
2:
3: import * as React from "react"
4: import * as ProgressPrimitive from "@radix-ui/react-progress"
5:
6: import { cn } from "@lib/utils"
7:
8: const Progress = React.forwardRef<
9:   React.ElementRef<typeof ProgressPrimitive.Root>,
10:   React.ComponentPropsWithoutRef<typeof ProgressPrimitive.Root>
11: >(({ className, value, ...props }, ref) => (
12:   <ProgressPrimitive.Root
13:     ref={ref}
14:     className={cn(
15:       "relative h-4 w-full overflow-hidden rounded-full bg-secondary",
16:       className
17:     )}
18:     {...props}
19:   >
20:     <ProgressPrimitive.Indicator
21:       className="h-full w-full flex-1 bg-primary transition-all"
22:       style={{ transform: `translateX(-${100 - (value || 0)}%)` }}
23:     />
24:   </ProgressPrimitive.Root>
25: ))
26: Progress.displayName = ProgressPrimitive.Root.displayName
27:
28: export { Progress }

```

## ■ File: components/ui/radio-group.tsx

```

1: "use client"
2:
3: import * as React from "react"
4: import * as RadioGroupPrimitive from "@radix-ui/react-radio-group"
5: import { Circle } from "lucide-react"
6:
7: import { cn } from "@lib/utils"

```

```

8:
9: const RadioGroup = React.forwardRef<
10:   React.ElementRef<typeof RadioGroupPrimitive.Root>,
11:   React.ComponentPropsWithoutRef<typeof RadioGroupPrimitive.Root>
12: >(({ className, ...props }, ref) => {
13:   return (
14:     <RadioGroupPrimitive.Root
15:       className={cn("grid gap-2", className)}
16:       {...props}
17:       ref={ref}
18:     />
19:   )
20: })
21: RadioGroup.displayName = RadioGroupPrimitive.Root.displayName
22:
23: const RadioGroupItem = React.forwardRef<
24:   React.ElementRef<typeof RadioGroupPrimitive.Item>,
25:   React.ComponentPropsWithoutRef<typeof RadioGroupPrimitive.Item>
26: >(({ className, ...props }, ref) => {
27:   return (
28:     <RadioGroupPrimitive.Item
29:       ref={ref}
30:       className={cn(
31:         "aspect-square h-4 w-4 rounded-full border border-primary text-primary ring-offset-background focus
32:         className
33:       )}
34:       {...props}
35:     >
36:       <RadioGroupPrimitive.Indicator className="flex items-center justify-center">
37:         <Circle className="h-2.5 w-2.5 fill-current text-current" />
38:       </RadioGroupPrimitive.Indicator>
39:     </RadioGroupPrimitive.Item>
40:   )
41: })
42: RadioGroupItem.displayName = RadioGroupPrimitive.Item.displayName
43:
44: export { RadioGroup, RadioGroupItem }

```

## ■ File: components\ui\resizable.tsx

```

1: "use client"
2:
3: import { GripVertical } from "lucide-react"
4: import * as ResizablePrimitive from "react-resizable-panels"
5:
6: import { cn } from "@lib/Utils"
7:
8: const ResizablePanelGroup = ({
9:   className,
10:   ...props
11: }: React.ComponentProps<typeof ResizablePrimitive.PanelGroup>) => (
12:   <ResizablePrimitive.PanelGroup
13:     className={cn(
14:       "flex h-full w-full data-[panel-group-direction=vertical]:flex-col",
15:       className
16:     )}
17:     {...props}
18:   />
19: )
20:
21: const ResizablePanel = ResizablePrimitive.Panel
22:
23: const ResizableHandle = ({
24:   withHandle,
25:   className,
26:   ...props
27: }: React.ComponentProps<typeof ResizablePrimitive.PanelResizeHandle> & {
28:   withHandle?: boolean
29: }) => (
30:   <ResizablePrimitive.PanelResizeHandle
31:     className={cn(

```

```

32:         "relative flex w-px items-center justify-center bg-border after:absolute after:inset-y-0 after:left-1
33:         className
34:     )}
35:     {...props}
36: >
37:     {withHandle && (
38:         <div className="z-10 flex h-4 w-3 items-center justify-center rounded-sm border bg-border">
39:             <GripVertical className="h-2.5 w-2.5" />
40:         </div>
41:     )}
42: </ResizablePrimitive.PanelResizeHandle>
43: )
44:
45: export { ResizablePanelGroup, ResizablePanel, ResizableHandle }

```

## ■ File: components\ui\scroll-area.tsx

```

1: "use client"
2:
3: import * as React from "react"
4: import * as ScrollAreaPrimitive from "@radix-ui/react-scroll-area"
5:
6: import { cn } from "@lib/Utils"
7:
8: const ScrollArea = React.forwardRef<
9:   React.ElementRef<typeof ScrollAreaPrimitive.Root>,
10:   React.ComponentPropsWithoutRef<typeof ScrollAreaPrimitive.Root>
11: >(({ className, children, ...props }, ref) => (
12:   <ScrollAreaPrimitive.Root
13:     ref={ref}
14:     className={cn("relative overflow-hidden", className)}
15:     {...props}
16:   >
17:     <ScrollAreaPrimitive.Viewport className="h-full w-full rounded-[inherit]">
18:       {children}
19:     </ScrollAreaPrimitive.Viewport>
20:     <ScrollBar />
21:     <ScrollAreaPrimitive.Corner />
22:   </ScrollAreaPrimitive.Root>
23: ))
24: ScrollArea.displayName = ScrollAreaPrimitive.Root.displayName
25:
26: const ScrollBar = React.forwardRef<
27:   React.ElementRef<typeof ScrollAreaPrimitive.ScrollAreaScrollbar>,
28:   React.ComponentPropsWithoutRef<typeof ScrollAreaPrimitive.ScrollAreaScrollbar>
29: >(({ className, orientation = "vertical", ...props }, ref) => (
30:   <ScrollAreaPrimitive.ScrollAreaScrollbar
31:     ref={ref}
32:     orientation={orientation}
33:     className={cn(
34:       "flex touch-none select-none transition-colors",
35:       orientation === "vertical" &&
36:         "h-full w-2.5 border-l border-l-transparent p-[1px]",
37:       orientation === "horizontal" &&
38:         "h-2.5 flex-col border-t border-t-transparent p-[1px]",
39:       className
40:     )}
41:     {...props}
42:   >
43:     <ScrollAreaPrimitive.ScrollAreaThumb className="relative flex-1 rounded-full bg-border" />
44:   </ScrollAreaPrimitive.ScrollAreaScrollbar>
45: ))
46: ScrollBar.displayName = ScrollAreaPrimitive.ScrollAreaScrollbar.displayName
47:
48: export { ScrollArea, ScrollBar }

```

## ■ File: components/ui/select.tsx

```
=====
1: "use client"
2:
3: import * as React from "react"
4: import * as SelectPrimitive from "@radix-ui/react-select"
5: import { Check, ChevronDown, ChevronUp } from "lucide-react"
6:
7: import { cn } from "@/lib/utils"
8:
9: const Select = SelectPrimitive.Root
10:
11: const SelectGroup = SelectPrimitive.Group
12:
13: const SelectValue = SelectPrimitive.Value
14:
15: const SelectTrigger = React.forwardRef<
16:   React.ElementRef<typeof SelectPrimitive.Trigger>,
17:   React.ComponentPropsWithoutRef<typeof SelectPrimitive.Trigger>
18: >(({ className, children, ...props }, ref) => (
19:   <SelectPrimitive.Trigger
20:     ref={ref}
21:     className={cn(
22:       "flex h-10 w-full items-center justify-between rounded-md border border-input bg-background px-3 py-2",
23:       className
24:     )}
25:     {...props}
26:   >
27:     {children}
28:     <SelectPrimitive.Icon asChild>
29:       <ChevronDown className="h-4 w-4 opacity-50" />
30:     </SelectPrimitive.Icon>
31:   </SelectPrimitive.Trigger>
32: ))
33: SelectTrigger.displayName = SelectPrimitive.Trigger.displayName
34:
35: const SelectScrollUpButton = React.forwardRef<
36:   React.ElementRef<typeof SelectPrimitive.ScrollUpButton>,
37:   React.ComponentPropsWithoutRef<typeof SelectPrimitive.ScrollUpButton>
38: >(({ className, ...props }, ref) => (
39:   <SelectPrimitive.ScrollUpButton
40:     ref={ref}
41:     className={cn(
42:       "flex cursor-default items-center justify-center py-1",
43:       className
44:     )}
45:     {...props}
46:   >
47:     <ChevronUp className="h-4 w-4" />
48:   </SelectPrimitive.ScrollUpButton>
49: ))
50: SelectScrollUpButton.displayName = SelectPrimitive.ScrollUpButton.displayName
51:
52: const SelectScrollDownButton = React.forwardRef<
53:   React.ElementRef<typeof SelectPrimitive.ScrollDownButton>,
54:   React.ComponentPropsWithoutRef<typeof SelectPrimitive.ScrollDownButton>
55: >(({ className, ...props }, ref) => (
56:   <SelectPrimitive.ScrollDownButton
57:     ref={ref}
58:     className={cn(
59:       "flex cursor-default items-center justify-center py-1",
60:       className
61:     )}
62:     {...props}
63:   >
64:     <ChevronDown className="h-4 w-4" />
65:   </SelectPrimitive.ScrollDownButton>
66: ))
67: SelectScrollDownButton.displayName =
68:   SelectPrimitive.ScrollDownButton.displayName
69:
70: const SelectContent = React.forwardRef<
71:   React.ElementRef<typeof SelectPrimitive.Content>,>
```



```

72:   React.ComponentPropsWithoutRef<typeof SelectPrimitive.Content>
73: >(({ className, children, position = "popper", ...props }, ref) => (
74:   <SelectPrimitive.Portal>
75:     <SelectPrimitive.Content
76:       ref={ref}
77:       className={cn(
78:         "relative z-50 max-h-96 min-w-[8rem] overflow-hidden rounded-md border bg-popover text-popover-forever
79:         position === "popper" &&
80:         "data-[side=bottom]:translate-y-1 data-[side=left]:-translate-x-1 data-[side=right]:translate-x-1
81:         className
82:       )}
83:       position={position}
84:       {...props}
85:     >
86:       <SelectScrollUpButton />
87:       <SelectPrimitive.Viewport
88:         className={cn(
89:           "p-1",
90:           position === "popper" &&
91:           "h-[var(--radix-select-trigger-height)] w-full min-w-[var(--radix-select-trigger-width)]"
92:         )}
93:       >
94:         {children}
95:       </SelectPrimitive.Viewport>
96:       <SelectScrollDownButton />
97:     </SelectPrimitive.Content>
98:   </SelectPrimitive.Portal>
99: ))
100: SelectContent.displayName = SelectPrimitive.Content.displayName
101:
102: const SelectLabel = React.forwardRef<
103:   React.ElementRef<typeof SelectPrimitive.Label>,
104:   React.ComponentPropsWithoutRef<typeof SelectPrimitive.Label>
105: >(({ className, ...props }, ref) => (
106:   <SelectPrimitive.Label
107:     ref={ref}
108:     className={cn("py-1.5 pl-8 pr-2 text-sm font-semibold", className)}
109:     {...props}
110:   />
111: ))
112: SelectLabel.displayName = SelectPrimitive.Label.displayName
113:
114: const SelectItem = React.forwardRef<
115:   React.ElementRef<typeof SelectPrimitive.Item>,
116:   React.ComponentPropsWithoutRef<typeof SelectPrimitive.Item>
117: >(({ className, children, ...props }, ref) => (
118:   <SelectPrimitive.Item
119:     ref={ref}
120:     className={cn(
121:       "relative flex w-full cursor-default select-none items-center rounded-sm py-1.5 pl-8 pr-2 text-sm outline
122:       className
123:     )}
124:     {...props}
125:   >
126:     <span className="absolute left-2 flex h-3.5 w-3.5 items-center justify-center">
127:       <SelectPrimitive.ItemIndicator>
128:         <Check className="h-4 w-4" />
129:       </SelectPrimitive.ItemIndicator>
130:     </span>
131:
132:     <SelectPrimitive.ItemText>{children}</SelectPrimitive.ItemText>
133:   </SelectPrimitive.Item>
134: ))
135: SelectItem.displayName = SelectPrimitive.Item.displayName
136:
137: const SelectSeparator = React.forwardRef<
138:   React.ElementRef<typeof SelectPrimitive.Separator>,
139:   React.ComponentPropsWithoutRef<typeof SelectPrimitive.Separator>
140: >(({ className, ...props }, ref) => (
141:   <SelectPrimitive.Separator
142:     ref={ref}
143:     className={cn("-mx-1 my-1 h-px bg-muted", className)}
144:     {...props}

```

```

145:   />
146: ))
147: SelectSeparator.displayName = SelectPrimitive.Separator.displayName
148:
149: export {
150:   Select,
151:   SelectGroup,
152:   SelectValue,
153:   SelectTrigger,
154:   SelectContent,
155:   SelectLabel,
156:   SelectItem,
157:   SelectSeparator,
158:   SelectScrollUpButton,
159:   SelectScrollDownButton,
160: }

```

## ■ File: components\ui\separator.tsx

```

1: "use client"
2:
3: import * as React from "react"
4: import * as SeparatorPrimitive from "@radix-ui/react-separator"
5:
6: import { cn } from "@lib/utils"
7:
8: const Separator = React.forwardRef<
9:   React.ElementRef<typeof SeparatorPrimitive.Root>,
10:   React.ComponentPropsWithoutRef<typeof SeparatorPrimitive.Root>
11: > (
12:   (
13:     { className, orientation = "horizontal", decorative = true, ...props },
14:     ref
15:   ) => (
16:     <SeparatorPrimitive.Root
17:       ref={ref}
18:       decorative={decorative}
19:       orientation={orientation}
20:       className={cn(
21:         "shrink-0 bg-border",
22:         orientation === "horizontal" ? "h-[1px] w-full" : "h-full w-[1px]",
23:         className
24:       )}
25:       {...props}
26:     />
27:   )
28: )
29: Separator.displayName = SeparatorPrimitive.Root.displayName
30:
31: export { Separator }

```

## ■ File: components\ui\sheet.tsx

```

1: "use client"
2:
3: import * as React from "react"
4: import * as SheetPrimitive from "@radix-ui/react-dialog"
5: import { cva, type VariantProps } from "class-variance-authority"
6: import { X } from "lucide-react"
7:
8: import { cn } from "@lib/utils"
9:
10: const Sheet = SheetPrimitive.Root
11:
12: const SheetTrigger = SheetPrimitive.Trigger
13:
14: const SheetClose = SheetPrimitive.Close
15:
16: const SheetPortal = SheetPrimitive.Portal

```

```

17:
18: const SheetOverlay = React.forwardRef<
19:   React.ElementRef<typeof SheetPrimitive.Overlay>,
20:   React.ComponentPropsWithoutRef<typeof SheetPrimitive.Overlay>
21: >(({ className, ...props }, ref) => (
22:   <SheetPrimitive.Overlay
23:     className={cn(
24:       "fixed inset-0 z-50 bg-black/80 data-[state=open]:animate-in data-[state=closed]:animate-out data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0",
25:       className
26:     )}
27:     {...props}
28:     ref={ref}
29:   />
30: ))
31: SheetOverlay.displayName = SheetPrimitive.Overlay.displayName
32:
33: const sheetVariants = cva(
34:   "fixed z-50 gap-4 bg-background p-6 shadow-lg transition ease-in-out data-[state=open]:animate-in data-[state=closed]:animate-out data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0",
35:   {
36:     variants: {
37:       side: {
38:         top: "inset-x-0 top-0 border-b data-[state=closed]:slide-out-to-top data-[state=open]:slide-in-from-top",
39:         bottom:
40:           "inset-x-0 bottom-0 border-t data-[state=closed]:slide-out-to-bottom data-[state=open]:slide-in-from-bottom",
41:         left: "inset-y-0 left-0 h-full w-3/4 border-r data-[state=closed]:slide-out-to-left data-[state=open]:slide-in-from-left",
42:         right:
43:           "inset-y-0 right-0 h-full w-3/4 border-l data-[state=closed]:slide-out-to-right data-[state=open]:slide-in-from-right",
44:       },
45:     },
46:     defaultVariants: {
47:       side: "right",
48:     },
49:   }
50: )
51:
52: interface SheetContentProps
53:   extends React.ComponentPropsWithoutRef<typeof SheetPrimitive.Content>,
54:   VariantProps<typeof sheetVariants> {}
55:
56: const SheetContent = React.forwardRef<
57:   React.ElementRef<typeof SheetPrimitive.Content>,
58:   SheetContentProps
59: >(({ side = "right", className, children, ...props }, ref) => (
60:   <SheetPortal>
61:     <SheetOverlay />
62:     <SheetPrimitive.Content
63:       ref={ref}
64:       className={cn(sheetVariants({ side }), className)}
65:       {...props}
66:     >
67:       {children}
68:       <SheetPrimitive.Close className="absolute right-4 top-4 rounded-sm opacity-70 ring-offset-background transition-opacity hover:opacity-100 focus:outline-none focus:ring-2 focus:ring-ring focus:ring-offset-2" />
69:       <X className="h-4 w-4" />
70:       <span className="sr-only">Close</span>
71:     </SheetPrimitive.Close>
72:   </SheetPrimitive.Content>
73: </SheetPortal>
74: ))
75: SheetContent.displayName = SheetPrimitive.Content.displayName
76:
77: const SheetHeader = ({
78:   className,
79:   ...props
80: }: React.HTMLAttributes<HTMLDivElement>) => (
81:   <div
82:     className={cn(
83:       "flex flex-col space-y-2 text-center sm:text-left",
84:       className
85:     )}
86:     {...props}
87:   />
88: )
89: SheetHeader.displayName = "SheetHeader"

```

```

90:
91: const SheetFooter = ({
92:   className,
93:   ...props
94: }: React.HTMLAttributes<HTMLDivElement>) => (
95:   <div
96:     className={cn(
97:       "flex flex-col-reverse sm:flex-row sm:justify-end sm:space-x-2",
98:       className
99:     )}
100:     {...props}
101:   />
102: )
103: SheetFooter.displayName = "SheetFooter"
104:
105: const SheetTitle = React.forwardRef<
106:   React.ElementRef<typeof SheetPrimitive.Title>,
107:   React.ComponentPropsWithoutRef<typeof SheetPrimitive.Title>
108: >(({ className, ...props }, ref) => (
109:   <SheetPrimitive.Title
110:     ref={ref}
111:     className={cn("text-lg font-semibold text-foreground", className)}
112:     {...props}
113:   />
114: ))
115: SheetTitle.displayName = SheetPrimitive.Title.displayName
116:
117: const SheetDescription = React.forwardRef<
118:   React.ElementRef<typeof SheetPrimitive.Description>,
119:   React.ComponentPropsWithoutRef<typeof SheetPrimitive.Description>
120: >(({ className, ...props }, ref) => (
121:   <SheetPrimitive.Description
122:     ref={ref}
123:     className={cn("text-sm text-muted-foreground", className)}
124:     {...props}
125:   />
126: ))
127: SheetDescription.displayName = SheetPrimitive.Description.displayName
128:
129: export {
130:   Sheet,
131:   SheetPortal,
132:   SheetOverlay,
133:   SheetTrigger,
134:   SheetClose,
135:   SheetContent,
136:   SheetHeader,
137:   SheetFooter,
138:   SheetTitle,
139:   SheetDescription,
140: }

```

## ■ File: components/ui/sidebar.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import { Slot } from "@radix-ui/react-slot"
5: import { VariantProps, cva } from "class-variance-authority"
6: import { PanelLeft } from "lucide-react"
7:
8: import { useIsMobile } from "@/hooks/use-mobile"
9: import { cn } from "@/lib/utlis"
10: import { Button } from "@/components/ui/button"
11: import { Input } from "@/components/ui/input"
12: import { Separator } from "@/components/ui/separator"
13: import { Sheet, SheetContent } from "@/components/ui/sheet"
14: import { Skeleton } from "@/components/ui/skeleton"
15: import {
16:   Tooltip,
17:   TooltipContent,

```

```

18:   TooltipProvider,
19:   TooltipTrigger,
20: } from "@components/ui/tooltip"
21:
22: const SIDEBAR_COOKIE_NAME = "sidebar:state"
23: const SIDEBAR_COOKIE_MAX_AGE = 60 * 60 * 24 * 7
24: const SIDEBAR_WIDTH = "16rem"
25: const SIDEBAR_WIDTH_MOBILE = "18rem"
26: const SIDEBAR_WIDTH_ICON = "3rem"
27: const SIDEBAR_KEYBOARD_SHORTCUT = "b"
28:
29: type SidebarContext = {
30:   state: "expanded" | "collapsed"
31:   open: boolean
32:   setOpen: (open: boolean) => void
33:   openMobile: boolean
34:   setOpenMobile: (open: boolean) => void
35:   isMobile: boolean
36:   toggleSidebar: () => void
37: }
38:
39: const SidebarContext = React.createContext<SidebarContext | null>(null)
40:
41: function useSidebar() {
42:   const context = React.useContext(SidebarContext)
43:   if (!context) {
44:     throw new Error("useSidebar must be used within a SidebarProvider.")
45:   }
46:
47:   return context
48: }
49:
50: const SidebarProvider = React.forwardRef<
51:   HTMLDivElement,
52:   React.ComponentProps<"div"> & {
53:     defaultOpen?: boolean
54:     open?: boolean
55:     onOpenChange?: (open: boolean) => void
56:   }
57: >((
58:   {
59:     {
60:       defaultOpen = true,
61:       open: openProp,
62:       onOpenChange: setOpenProp,
63:       className,
64:       style,
65:       children,
66:       ...props
67:     },
68:     ref
69:   ) => {
70:     const isMobile = useIsMobile()
71:     const [openMobile, setOpenMobile] = React.useState(false)
72:
73:     // This is the internal state of the sidebar.
74:     // We use openProp and setOpenProp for control from outside the component.
75:     const [_open, _setOpen] = React.useState(defaultOpen)
76:     const open = openProp ?? _open
77:     const setOpen = React.useCallback(
78:       (value: boolean | ((value: boolean) => boolean)) => {
79:         const openState = typeof value === "function" ? value(open) : value
80:         if (setOpenProp) {
81:           setOpenProp(openState)
82:         } else {
83:           _setOpen(openState)
84:         }
85:
86:         // This sets the cookie to keep the sidebar state.
87:         document.cookie = `${SIDEBAR_COOKIE_NAME}=${openState}; path=/; max-age=${SIDEBAR_COOKIE_MAX_AGE}`
88:       },
89:       [setOpenProp, open]
90:     )

```

```

91:
92: // Helper to toggle the sidebar.
93: const toggleSidebar = React.useCallback(() => {
94:   return isMobile
95:     ? setOpenMobile((open) => !open)
96:     : setOpen((open) => !open)
97: }, [isMobile, setOpen, setOpenMobile])
98:
99: // Adds a keyboard shortcut to toggle the sidebar.
100: React.useEffect(() => {
101:   const handleKeyDown = (event: KeyboardEvent) => {
102:     if (
103:       event.key === SIDEBAR_KEYBOARD_SHORTCUT &&
104:       (event.metaKey || event.ctrlKey)
105:     ) {
106:       event.preventDefault()
107:       toggleSidebar()
108:     }
109:   }
110:
111:   window.addEventListener("keydown", handleKeyDown)
112:   return () => window.removeEventListener("keydown", handleKeyDown)
113: }, [toggleSidebar])
114:
115: // We add a state so that we can do data-state="expanded" or "collapsed".
116: // This makes it easier to style the sidebar with Tailwind classes.
117: const state = open ? "expanded" : "collapsed"
118:
119: const contextValue = React.useMemo<SidebarContext>(
120:   () => ({
121:     state,
122:     open,
123:     setOpen,
124:     isMobile,
125:     openMobile,
126:     setOpenMobile,
127:     toggleSidebar,
128:   }),
129:   [state, open, setOpen, isMobile, openMobile, setOpenMobile, toggleSidebar]
130: )
131:
132: return (
133:   <SidebarContext.Provider value={contextValue}>
134:     <TooltipProvider delayDuration={0}>
135:       <div
136:         style={
137:           {
138:             "--sidebar-width": SIDEBAR_WIDTH,
139:             "--sidebar-width-icon": SIDEBAR_WIDTH_ICON,
140:             ...style,
141:           } as React.CSSProperties
142:         }
143:         className={cn(
144:           "group/sidebar-wrapper flex min-h-svh w-full has-[[data-variant=inset]]:bg-sidebar",
145:           className
146:         )}
147:         ref={ref}
148:         {...props}
149:       >
150:         {children}
151:       </div>
152:     </TooltipProvider>
153:   </SidebarContext.Provider>
154: )
155: }
156: )
157: SidebarProvider.displayName = "SidebarProvider"
158:
159: const Sidebar = React.forwardRef<
160:   HTMLDivElement,
161:   React.ComponentProps<"div"> & {
162:     side?: "left" | "right"
163:     variant?: "sidebar" | "floating" | "inset"

```

```

164:     collapsible?: "offcanvas" | "icon" | "none"
165:   }
166: >(
167:   (
168:     {
169:       side = "left",
170:       variant = "sidebar",
171:       collapsible = "offcanvas",
172:       className,
173:       children,
174:       ...props
175:     },
176:     ref
177:   ) => {
178:     const { isMobile, state, openMobile, setOpenMobile } = useSidebar()
179:
180:     if (collapsible === "none") {
181:       return (
182:         <div
183:           className={cn(
184:             "flex h-full w-[--sidebar-width] flex-col bg-sidebar text-sidebar-foreground",
185:             className
186:           )}
187:           ref={ref}
188:           {...props}
189:         >
190:           {children}
191:         </div>
192:       )
193:     }
194:
195:     if (isMobile) {
196:       return (
197:         <Sheet open={openMobile} onOpenChange={setOpenMobile} {...props}>
198:           <SheetContent
199:             data-sidebar="sidebar"
200:             data-mobile="true"
201:             className="w-[--sidebar-width] bg-sidebar p-0 text-sidebar-foreground [&button]:hidden"
202:             style={
203:               {
204:                 "--sidebar-width": SIDEBAR_WIDTH_MOBILE,
205:               } as React.CSSProperties
206:             }
207:             side={side}
208:           >
209:             <div className="flex h-full w-full flex-col">{children}</div>
210:           </SheetContent>
211:         </Sheet>
212:       )
213:     }
214:
215:     return (
216:       <div
217:         ref={ref}
218:         className="group peer hidden md:block text-sidebar-foreground"
219:         data-state={state}
220:         data-collapsible={state === "collapsed" ? collapsible : ""}
221:         data-variant={variant}
222:         data-side={side}
223:       >
224:         { /* This is what handles the sidebar gap on desktop */ }
225:         <div
226:           className={cn(
227:             "duration-200 relative h-svh w-[--sidebar-width] bg-transparent transition-[width] ease-linear"
228:             "group-data-[collapsible=offcanvas]:w-0",
229:             "group-data-[side=right]:rotate-180",
230:             variant === "floating" || variant === "inset"
231:               ? "group-data-[collapsible=icon]:w-[calc(var(--sidebar-width-icon)_+_theme(spacing.4))]"
232:               : "group-data-[collapsible=icon]:w-[--sidebar-width-icon]"
233:           )}
234:         />
235:         <div
236:           className={cn(

```

```

237:         "duration-200 fixed inset-y-0 z-10 hidden h-svh w-[--sidebar-width] transition-[left,right,width]
238:         side === "left"
239:         ? "left-0 group-data-[collapsible=offcanvas]:left-[calc(var(--sidebar-width)*-1)]"
240:         : "right-0 group-data-[collapsible=offcanvas]:right-[calc(var(--sidebar-width)*-1)]",
241:         // Adjust the padding for floating and inset variants.
242:         variant === "floating" || variant === "inset"
243:         ? "p-2 group-data-[collapsible=icon]:w-[calc(var(--sidebar-width-icon)+_theme(spacing.4)+2px)]"
244:         : "group-data-[collapsible=icon]:w-[--sidebar-width-icon] group-data-[side=left]:border-r group-data-[side=right]:border-l",
245:         className
246:       )}
247:       {...props}
248:     >
249:       <div
250:         data-sidebar="sidebar"
251:         className="flex h-full w-full flex-col bg-sidebar group-data-[variant=floating]:rounded-lg group-data-[variant=floating]:border group-data-[variant=floating]:border-sidebar"
252:       >
253:         {children}
254:       </div>
255:     </div>
256:   </div>
257: )
258: }
259: )
260: Sidebar.displayName = "Sidebar"
261:
262: const SidebarTrigger = React.forwardRef<
263:   React.ElementRef<typeof Button>,
264:   React.ComponentProps<typeof Button>
265: >(({ className, onClick, ...props }, ref) => {
266:   const { toggleSidebar } = useSidebar()
267:
268:   return (
269:     <Button
270:       ref={ref}
271:       data-sidebar="trigger"
272:       variant="ghost"
273:       size="icon"
274:       className={cn("h-7 w-7", className)}
275:       onClick={(event) => {
276:         onClick?.(event)
277:         toggleSidebar()
278:       }}
279:       {...props}
280:     >
281:       <PanelLeft />
282:       <span className="sr-only">Toggle Sidebar</span>
283:     </Button>
284:   )
285: })
286: SidebarTrigger.displayName = "SidebarTrigger"
287:
288: const SidebarRail = React.forwardRef<
289:   HTMLButtonElement,
290:   React.ComponentProps<"button">
291: >(({ className, ...props }, ref) => {
292:   const { toggleSidebar } = useSidebar()
293:
294:   return (
295:     <button
296:       ref={ref}
297:       data-sidebar="rail"
298:       aria-label="Toggle Sidebar"
299:       tabIndex={-1}
300:       onClick={toggleSidebar}
301:       title="Toggle Sidebar"
302:       className={cn(
303:         "absolute inset-y-0 z-20 hidden w-4 -translate-x-1/2 transition-all ease-linear after:absolute after:inset-y-0 after:left-0 after:right-0 after:w-full",
304:         "[data-side=left]&:cursor-w-resize [[data-side=right]&]:cursor-e-resize",
305:         "[data-side=left][data-state=collapsed]&:cursor-e-resize [[data-side=right][data-state=collapsed]&]:cursor-w-resize",
306:         "group-data-[collapsible=offcanvas]:translate-x-0 group-data-[collapsible=offcanvas]:after:left-full group-data-[collapsible=offcanvas]:after:right-full",
307:         "[data-side=left][data-collapsible=offcanvas]&:-right-2",
308:         "[data-side=right][data-collapsible=offcanvas]&:-left-2",
309:         className

```



```

310:     })
311:     {...props}
312:   />
313: )
314: })
315: SidebarRail.displayName = "SidebarRail"
316:
317: const SidebarInset = React.forwardRef<
318:   HTMLDivElement,
319:   React.ComponentProps<"main">
320: >(({ className, ...props }, ref) => {
321:   return (
322:     <main
323:       ref={ref}
324:       className={cn(
325:         "relative flex min-h-svh flex-1 flex-col bg-background",
326:         "peer-data-[variant=inset]:min-h-[calc(100svh-theme(spacing.4))]" md:peer-data-[variant=inset]:m-2 m
327:         className
328:       )}
329:       {...props}
330:     />
331:   )
332: })
333: SidebarInset.displayName = "SidebarInset"
334:
335: const SidebarInput = React.forwardRef<
336:   React.ElementRef<typeof Input>,
337:   React.ComponentProps<typeof Input>
338: >(({ className, ...props }, ref) => {
339:   return (
340:     <Input
341:       ref={ref}
342:       data-sidebar="input"
343:       className={cn(
344:         "h-8 w-full bg-background shadow-none focus-visible:ring-2 focus-visible:ring-sidebar-ring",
345:         className
346:       )}
347:       {...props}
348:     />
349:   )
350: })
351: SidebarInput.displayName = "SidebarInput"
352:
353: const SidebarHeader = React.forwardRef<
354:   HTMLDivElement,
355:   React.ComponentProps<"div">
356: >(({ className, ...props }, ref) => {
357:   return (
358:     <div
359:       ref={ref}
360:       data-sidebar="header"
361:       className={cn("flex flex-col gap-2 p-2", className)}
362:       {...props}
363:     />
364:   )
365: })
366: SidebarHeader.displayName = "SidebarHeader"
367:
368: const SidebarFooter = React.forwardRef<
369:   HTMLDivElement,
370:   React.ComponentProps<"div">
371: >(({ className, ...props }, ref) => {
372:   return (
373:     <div
374:       ref={ref}
375:       data-sidebar="footer"
376:       className={cn("flex flex-col gap-2 p-2", className)}
377:       {...props}
378:     />
379:   )
380: })
381: SidebarFooter.displayName = "SidebarFooter"
382:

```

```

383: const SidebarSeparator = React.forwardRef<
384:   React.ElementRef<typeof Separator>,
385:   React.ComponentProps<typeof Separator>
386: >(({ className, ...props }, ref) => {
387:   return (
388:     <Separator
389:       ref={ref}
390:       data-sidebar="separator"
391:       className={cn("mx-2 w-auto bg-sidebar-border", className)}
392:       {...props}
393:     />
394:   )
395: })
396: SidebarSeparator.displayName = "SidebarSeparator"
397:
398: const SidebarContent = React.forwardRef<
399:   HTMLDivElement,
400:   React.ComponentProps<"div">
401: >(({ className, ...props }, ref) => {
402:   return (
403:     <div
404:       ref={ref}
405:       data-sidebar="content"
406:       className={cn(
407:         "flex min-h-0 flex-1 flex-col gap-2 overflow-auto group-data-[collapsible=icon]:overflow-hidden",
408:         className
409:       )}
410:       {...props}
411:     />
412:   )
413: })
414: SidebarContent.displayName = "SidebarContent"
415:
416: const SidebarGroup = React.forwardRef<
417:   HTMLDivElement,
418:   React.ComponentProps<"div">
419: >(({ className, ...props }, ref) => {
420:   return (
421:     <div
422:       ref={ref}
423:       data-sidebar="group"
424:       className={cn("relative flex w-full min-w-0 flex-col p-2", className)}
425:       {...props}
426:     />
427:   )
428: })
429: SidebarGroup.displayName = "SidebarGroup"
430:
431: const SidebarGroupLabel = React.forwardRef<
432:   HTMLDivElement,
433:   React.ComponentProps<"div"> & { asChild?: boolean }
434: >(({ className, asChild = false, ...props }, ref) => {
435:   const Comp = asChild ? Slot : "div"
436:
437:   return (
438:     <Comp
439:       ref={ref}
440:       data-sidebar="group-label"
441:       className={cn(
442:         "duration-200 flex h-8 shrink-0 items-center rounded-md px-2 text-xs font-medium text-sidebar-foreground",
443:         "group-data-[collapsible=icon]:-mt-8 group-data-[collapsible=icon]:opacity-0",
444:         className
445:       )}
446:       {...props}
447:     />
448:   )
449: })
450: SidebarGroupLabel.displayName = "SidebarGroupLabel"
451:
452: const SidebarGroupAction = React.forwardRef<
453:   HTMLButtonElement,
454:   React.ComponentProps<"button"> & { asChild?: boolean }
455: >(({ className, asChild = false, ...props }, ref) => {

```

```

456:   const Comp = asChild ? Slot : "button"
457:
458:   return (
459:     <Comp
460:       ref={ref}
461:       data-sidebar="group-action"
462:       className={cn(
463:         "absolute right-3 top-3.5 flex aspect-square w-5 items-center justify-center rounded-md p-0 text-si
464:         // Increases the hit area of the button on mobile.
465:         "after:absolute after:-inset-2 after:md:hidden",
466:         "group-data-[collapsible=icon]:hidden",
467:         className
468:       )}
469:       {...props}
470:     />
471:   )
472: })
473: SidebarGroupAction.displayName = "SidebarGroupAction"
474:
475: const SidebarGroupContent = React.forwardRef<
476:   HTMLDivElement,
477:   React.ComponentProps<"div">
478: >(({ className, ...props }, ref) => (
479:   <div
480:     ref={ref}
481:     data-sidebar="group-content"
482:     className={cn("w-full text-sm", className)}
483:     {...props}
484:   />
485: ))
486: SidebarGroupContent.displayName = "SidebarGroupContent"
487:
488: const SidebarMenu = React.forwardRef<
489:   HTMLUListElement,
490:   React.ComponentProps<"ul">
491: >(({ className, ...props }, ref) => (
492:   <ul
493:     ref={ref}
494:     data-sidebar="menu"
495:     className={cn("flex w-full min-w-0 flex-col gap-1", className)}
496:     {...props}
497:   />
498: ))
499: SidebarMenu.displayName = "SidebarMenu"
500:
501: const SidebarMenuItem = React.forwardRef<
502:   HTMLLIElement,
503:   React.ComponentProps<"li">
504: >(({ className, ...props }, ref) => (
505:   <li
506:     ref={ref}
507:     data-sidebar="menu-item"
508:     className={cn("group/menu-item relative", className)}
509:     {...props}
510:   />
511: ))
512: SidebarMenuItem.displayName = "SidebarMenuItem"
513:
514: const sidebarMenuButtonVariants = cva(
515:   "peer/menu-button flex w-full items-center gap-2 overflow-hidden rounded-md p-2 text-left text-sm outline
516:   {
517:     variants: {
518:       variant: {
519:         default: "hover:bg-sidebar-accent hover:text-sidebar-accent-foreground",
520:         outline:
521:           "bg-background shadow-[0_0_1px_hsl(var(--sidebar-border))] hover:bg-sidebar-accent hover:text-s
522:       },
523:       size: {
524:         default: "h-8 text-sm",
525:         sm: "h-7 text-xs",
526:         lg: "h-12 text-sm group-data-[collapsible=icon]:!p-0",
527:       },
528:     },

```

```

529:     defaultVariants: {
530:       variant: "default",
531:       size: "default",
532:     },
533:   }
534: )
535:
536: const SidebarMenuButton = React.forwardRef<
537:   HTMLButtonElement,
538:   React.ComponentProps<"button"> & {
539:     asChild?: boolean
540:     isActive?: boolean
541:     tooltip?: string | React.ComponentProps<typeof TooltipContent>
542:   } & VariantProps<typeof sidebarMenuButtonVariants>
543: > (
544:   (
545:     {
546:       asChild = false,
547:       isActive = false,
548:       variant = "default",
549:       size = "default",
550:       tooltip,
551:       className,
552:       ...props
553:     },
554:     ref
555:   ) => {
556:     const Comp = asChild ? Slot : "button"
557:     const { isMobile, state } = useSidebar()
558:
559:     const button = (
560:       <Comp
561:         ref={ref}
562:         data-sidebar="menu-button"
563:         data-size={size}
564:         data-active={isActive}
565:         className={cn(sidebarMenuButtonVariants({ variant, size })), className}
566:         {...props}
567:       />
568:     )
569:
570:     if (!tooltip) {
571:       return button
572:     }
573:
574:     if (typeof tooltip === "string") {
575:       tooltip = {
576:         children: tooltip,
577:       }
578:     }
579:
580:     return (
581:       <Tooltip>
582:         <TooltipTrigger asChild>{button}</TooltipTrigger>
583:         <TooltipContent
584:           side="right"
585:           align="center"
586:           hidden={state !== "collapsed" || isMobile}
587:           {...tooltip}
588:         />
589:       </Tooltip>
590:     )
591:   }
592: )
593: SidebarMenuButton.displayName = "SidebarMenuButton"
594:
595: const SidebarMenuAction = React.forwardRef<
596:   HTMLButtonElement,
597:   React.ComponentProps<"button"> & {
598:     asChild?: boolean
599:     showOnHover?: boolean
600:   }
601: > (({ className, asChild = false, showOnHover = false, ...props }, ref) => {

```

```

602:   const Comp = asChild ? Slot : "button"
603:
604:   return (
605:     <Comp
606:       ref={ref}
607:       data-sidebar="menu-action"
608:       className={cn(
609:         "absolute right-1 top-1.5 flex aspect-square w-5 items-center justify-center rounded-md p-0 text-si
610:         // Increases the hit area of the button on mobile.
611:         "after:absolute after:-inset-2 after:md:hidden",
612:         "peer-data-[size=sm]/menu-button:top-1",
613:         "peer-data-[size=default]/menu-button:top-1.5",
614:         "peer-data-[size=lg]/menu-button:top-2.5",
615:         "group-data-[collapsible=icon]:hidden",
616:         showOnHover &&
617:         "group-focus-within/menu-item:opacity-100 group-hover/menu-item:opacity-100 data-[state=open]:opa
618:         className
619:       )}
620:       {...props}
621:     />
622:   )
623: })
624: SidebarMenuAction.displayName = "SidebarMenuAction"
625:
626: const SidebarMenuBadge = React.forwardRef<
627:   HTMLDivElement,
628:   React.ComponentProps<"div">
629: >(({ className, ...props }, ref) => (
630:   <div
631:     ref={ref}
632:     data-sidebar="menu-badge"
633:     className={cn(
634:       "absolute right-1 flex h-5 min-w-5 items-center justify-center rounded-md px-1 text-xs font-medium tal
635:       "peer-hover/menu-button:text-sidebar-accent-foreground peer-data-[active=true]/menu-button:text-sideb
636:       "peer-data-[size=sm]/menu-button:top-1",
637:       "peer-data-[size=default]/menu-button:top-1.5",
638:       "peer-data-[size=lg]/menu-button:top-2.5",
639:       "group-data-[collapsible=icon]:hidden",
640:       className
641:     )}
642:     {...props}
643:   />
644: ))
645: SidebarMenuBadge.displayName = "SidebarMenuBadge"
646:
647: const SidebarMenuSkeleton = React.forwardRef<
648:   HTMLDivElement,
649:   React.ComponentProps<"div"> & {
650:     showIcon?: boolean
651:   }
652: >(({ className, showIcon = false, ...props }, ref) => {
653:   // Random width between 50 to 90%.
654:   const width = React.useMemo(() => {
655:     return `${Math.floor(Math.random() * 40) + 50}%`
656:   }, [])
657:
658:   return (
659:     <div
660:       ref={ref}
661:       data-sidebar="menu-skeleton"
662:       className={cn("rounded-md h-8 flex gap-2 px-2 items-center", className)}
663:       {...props}
664:     >
665:       {showIcon && (
666:         <Skeleton
667:           className="size-4 rounded-md"
668:           data-sidebar="menu-skeleton-icon"
669:         />
670:       )}
671:       <Skeleton
672:         className="h-4 flex-1 max-w-[--skeleton-width]"
673:         data-sidebar="menu-skeleton-text"
674:         style={

```

```

675:         {
676:             "--skeleton-width": width,
677:         } as React.CSSProperties
678:     }
679:     />
680: </div>
681: )
682: })
683: SidebarMenuSkeleton.displayName = "SidebarMenuSkeleton"
684:
685: const SidebarMenuSub = React.forwardRef<
686:   HTMLUListElement,
687:   React.ComponentProps<"ul">
688: >(({ className, ...props }, ref) => (
689:   <ul
690:     ref={ref}
691:     data-sidebar="menu-sub"
692:     className={cn(
693:       "mx-3.5 flex min-w-0 translate-x-px flex-col gap-1 border-l border-sidebar-border px-2.5 py-0.5",
694:       "group-data-[collapsible=icon]:hidden",
695:       className
696:     )}
697:     {...props}
698:   />
699: ))
700: SidebarMenuSub.displayName = "SidebarMenuSub"
701:
702: const SidebarMenuSubItem = React.forwardRef<
703:   HTMLLIElement,
704:   React.ComponentProps<"li">
705: >(({ ...props }, ref) => <li ref={ref} {...props} />)
706: SidebarMenuSubItem.displayName = "SidebarMenuSubItem"
707:
708: const SidebarMenuSubButton = React.forwardRef<
709:   HTMLAnchorElement,
710:   React.ComponentProps<"a"> & {
711:     asChild?: boolean
712:     size?: "sm" | "md"
713:     isActive?: boolean
714:   }
715: >(({ asChild = false, size = "md", isActive, className, ...props }, ref) => {
716:   const Comp = asChild ? Slot : "a"
717:
718:   return (
719:     <Comp
720:       ref={ref}
721:       data-sidebar="menu-sub-button"
722:       data-size={size}
723:       data-active={isActive}
724:       className={cn(
725:         "flex h-7 min-w-0 -translate-x-px items-center gap-2 overflow-hidden rounded-md px-2 text-sidebar-f",
726:         "data-[active=true]:bg-sidebar-accent data-[active=true]:text-sidebar-accent-foreground",
727:         size === "sm" && "text-xs",
728:         size === "md" && "text-sm",
729:         "group-data-[collapsible=icon]:hidden",
730:         className
731:       )}
732:       {...props}
733:     />
734:   )
735: })
736: SidebarMenuSubButton.displayName = "SidebarMenuSubButton"
737:
738: export {
739:   Sidebar,
740:   SidebarContent,
741:   SidebarFooter,
742:   SidebarGroup,
743:   SidebarGroupAction,
744:   SidebarGroupContent,
745:   SidebarGroupLabel,
746:   SidebarHeader,
747:   SidebarInput,

```

```
748:   SidebarInset,
749:   SidebarMenu,
750:   SidebarMenuAction,
751:   SidebarMenuBadge,
752:   SidebarMenuButton,
753:   SidebarMenuItem,
754:   SidebarMenuSkeleton,
755:   SidebarMenuSub,
756:   SidebarMenuSubButton,
757:   SidebarMenuSubItem,
758:   SidebarProvider,
759:   SidebarRail,
760:   SidebarSeparator,
761:   SidebarTrigger,
762:   useSidebar,
763: }
```

---

## ■ File: components\ui\skeleton.tsx

```
=====
1: import { cn } from "@/lib/utils"
2:
3: function Skeleton({
4:   className,
5:   ...props
6: }: React.HTMLAttributes<HTMLDivElement>) {
7:   return (
8:     <div
9:       className={cn("animate-pulse rounded-md bg-muted", className)}
10:      {...props}
11:    />
12:   )
13: }
14:
15: export { Skeleton }
```

---

## ■ File: components\ui\slider.tsx

```
=====
1: "use client"
2:
3: import * as React from "react"
4: import * as SliderPrimitive from "@radix-ui/react-slider"
5:
6: import { cn } from "@/lib/utils"
7:
8: const Slider = React.forwardRef<
9:   React.ElementRef<typeof SliderPrimitive.Root>,
10:   React.ComponentPropsWithoutRef<typeof SliderPrimitive.Root>
11: >(({ className, ...props }, ref) => (
12:   <SliderPrimitive.Root
13:     ref={ref}
14:     className={cn(
15:       "relative flex w-full touch-none select-none items-center",
16:       className
17:     )}
18:     {...props}
19:   >
20:     <SliderPrimitive.Track className="relative h-2 w-full grow overflow-hidden rounded-full bg-secondary">
21:       <SliderPrimitive.Range className="absolute h-full bg-primary" />
22:     </SliderPrimitive.Track>
23:     <SliderPrimitive.Thumb className="block h-5 w-5 rounded-full border-2 border-primary bg-background ring"
24:   </SliderPrimitive.Root>
25: ))
26: Slider.displayName = SliderPrimitive.Root.displayName
27:
28: export { Slider }
```

---

## ■ File: components/ui/sonner.tsx

```
=====
1: "use client"
2:
3: import { useTheme } from "next-themes"
4: import { Toaster as Sonner } from "sonner"
5:
6: type ToasterProps = React.ComponentProps<typeof Sonner>
7:
8: const Toaster = ({ ...props }: ToasterProps) => {
9:   const { theme = "system" } = useTheme()
10:
11:   return (
12:     <Sonner
13:       theme={theme as ToasterProps["theme"]}
14:       className="toaster group"
15:       toastOptions={{
16:         classNames: {
17:           toast:
18:             "group toast group-[.toaster]:bg-background group-[.toaster]:text-foreground group-[.toaster]:b
19:           description: "group-[.toast]:text-muted-foreground",
20:           actionButton:
21:             "group-[.toast]:bg-primary group-[.toast]:text-primary-foreground",
22:           cancelButton:
23:             "group-[.toast]:bg-muted group-[.toast]:text-muted-foreground",
24:         },
25:       }}
26:       {...props}
27:     />
28:   )
29: }
30:
31: export { Toaster }
```

## ■ File: components/ui/switch.tsx

```
=====
1: "use client"
2:
3: import * as React from "react"
4: import * as SwitchPrimitives from "@radix-ui/react-switch"
5:
6: import { cn } from "@/lib/utils"
7:
8: const Switch = React.forwardRef<
9:   React.ElementRef<typeof SwitchPrimitives.Root>,
10:   React.ComponentPropsWithoutRef<typeof SwitchPrimitives.Root>
11: >(({ className, ...props }, ref) => {
12:   <SwitchPrimitives.Root
13:     className={cn(
14:       "peer inline-flex h-6 w-11 shrink-0 cursor-pointer items-center rounded-full border-2 border-transparent
15:       className
16:     )}
17:     {...props}
18:     ref={ref}
19:   >
20:     <SwitchPrimitives.Thumb
21:       className={cn(
22:         "pointer-events-none block h-5 w-5 rounded-full bg-background shadow-lg ring-0 transition-transform
23:       )}
24:     />
25:   </SwitchPrimitives.Root>
26: ))
27: Switch.displayName = SwitchPrimitives.Root.displayName
28:
29: export { Switch }
```



## ■ File: components\ui\table.tsx

```
=====
1: import * as React from "react"
2:
3: import { cn } from "@/lib/utils"
4:
5: const Table = React.forwardRef<
6:   HTMLTableElement,
7:   React.HTMLAttributes<HTMLTableElement>
8: >(({ className, ...props }, ref) => (
9:   <div className="relative w-full overflow-auto">
10:     <table
11:       ref={ref}
12:       className={cn("w-full caption-bottom text-sm", className)}
13:       {...props}
14:     />
15:   </div>
16: ))
17: Table.displayName = "Table"
18:
19: const TableHeader = React.forwardRef<
20:   HTMLTableSectionElement,
21:   React.HTMLAttributes<HTMLTableSectionElement>
22: >(({ className, ...props }, ref) => (
23:   <thead ref={ref} className={cn("[&_tr]:border-b", className)} {...props} />
24: ))
25: TableHeader.displayName = "TableHeader"
26:
27: const TableBody = React.forwardRef<
28:   HTMLTableSectionElement,
29:   React.HTMLAttributes<HTMLTableSectionElement>
30: >(({ className, ...props }, ref) => (
31:   <tbody
32:     ref={ref}
33:     className={cn("[&_tr]:last-child:border-0", className)}
34:     {...props}
35:   />
36: ))
37: TableBody.displayName = "TableBody"
38:
39: const TableFooter = React.forwardRef<
40:   HTMLTableSectionElement,
41:   React.HTMLAttributes<HTMLTableSectionElement>
42: >(({ className, ...props }, ref) => (
43:   <tfoot
44:     ref={ref}
45:     className={cn(
46:       "border-t bg-muted/50 font-medium [&tr]:last:border-b-0",
47:       className
48:     )}
49:     {...props}
50:   />
51: ))
52: TableFooter.displayName = "TableFooter"
53:
54: const TableRow = React.forwardRef<
55:   HTMLTableRowElement,
56:   React.HTMLAttributes<HTMLTableRowElement>
57: >(({ className, ...props }, ref) => (
58:   <tr
59:     ref={ref}
60:     className={cn(
61:       "border-b transition-colors hover:bg-muted/50 data-[state=selected]:bg-muted",
62:       className
63:     )}
64:     {...props}
65:   />
66: ))
67: TableRow.displayName = "TableRow"
68:
69: const TableHead = React.forwardRef<
70:   HTMLTableCellElement,
71:   React.ThHTMLAttributes<HTMLTableCellElement>
```

```

72: >(({ className, ...props }, ref) => (
73:   <th
74:     ref={ref}
75:     className={cn(
76:       "h-12 px-4 text-left align-middle font-medium text-muted-foreground [&:has([role=checkbox])]:pr-0",
77:       className
78:     )}
79:     {...props}
80:   />
81: ))
82: TableHead.displayName = "TableHead"
83:
84: const TableCell = React.forwardRef<
85:   HTMLTableCellElement,
86:   React.TdHTMLAttributes<HTMLTableCellElement>
87: >(({ className, ...props }, ref) => (
88:   <td
89:     ref={ref}
90:     className={cn("p-4 align-middle [&:has([role=checkbox])]:pr-0", className)}
91:     {...props}
92:   />
93: ))
94: TableCell.displayName = "TableCell"
95:
96: const TableCaption = React.forwardRef<
97:   HTMLTableCaptionElement,
98:   React.HTMLAttributes<HTMLTableCaptionElement>
99: >(({ className, ...props }, ref) => (
100:   <caption
101:     ref={ref}
102:     className={cn("mt-4 text-sm text-muted-foreground", className)}
103:     {...props}
104:   />
105: ))
106: TableCaption.displayName = "TableCaption"
107:
108: export {
109:   Table,
110:   TableHeader,
111:   TableBody,
112:   TableFooter,
113:   TableHead,
114:   TableRow,
115:   TableCell,
116:   TableCaption,
117: }

```

## ■ File: components\ui\tabs.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import * as TabsPrimitive from "@radix-ui/react-tabs"
5:
6: import { cn } from "@lib/utils"
7:
8: const Tabs = TabsPrimitive.Root
9:
10: const TabsList = React.forwardRef<
11:   React.ElementRef<typeof TabsPrimitive.List>,
12:   React.ComponentPropsWithoutRef<typeof TabsPrimitive.List>
13: >(({ className, ...props }, ref) => (
14:   <TabsPrimitive.List
15:     ref={ref}
16:     className={cn(
17:       "inline-flex h-10 items-center justify-center rounded-md bg-muted p-1 text-muted-foreground",
18:       className
19:     )}
20:     {...props}
21:   />
22: ))

```

```

23: TabsList.displayName = TabsPrimitive.List.displayName
24:
25: const TabsTrigger = React.forwardRef<
26:   React.ElementRef<typeof TabsPrimitive.Trigger>,
27:   React.ComponentPropsWithoutRef<typeof TabsPrimitive.Trigger>
28: >(({ className, ...props }, ref) => (
29:   <TabsPrimitive.Trigger
30:     ref={ref}
31:     className={cn(
32:       "inline-flex items-center justify-center whitespace-nowrap rounded-sm px-3 py-1.5 text-sm font-medium
33:       className
34:     )}
35:     {...props}
36:   />
37: ))
38: TabsTrigger.displayName = TabsPrimitive.Trigger.displayName
39:
40: const TabsContent = React.forwardRef<
41:   React.ElementRef<typeof TabsPrimitive.Content>,
42:   React.ComponentPropsWithoutRef<typeof TabsPrimitive.Content>
43: >(({ className, ...props }, ref) => (
44:   <TabsPrimitive.Content
45:     ref={ref}
46:     className={cn(
47:       "mt-2 ring-offset-background focus-visible:outline-none focus-visible:ring-2 focus-visible:ring-ring
48:       className
49:     )}
50:     {...props}
51:   />
52: ))
53: TabsContent.displayName = TabsPrimitive.Content.displayName
54:
55: export { Tabs, TabsList, TabsTrigger, TabsContent }

```

## ■ File: components/ui/textarea.tsx

```

1: import * as React from "react";
2:
3: import { cn } from "@/lib/utils";
4:
5: const Textarea = React.forwardRef<
6:   HTMLTextAreaElement,
7:   React.ComponentProps<"textarea">
8: >(({ className, ...props }, ref) => {
9:   return (
10:     <textarea
11:       className={cn(
12:         "flex min-h-[80px] w-full rounded-md border border-input bg-white text-black px-3 py-2 text-base ri
13:         className
14:       )}
15:       ref={ref}
16:       {...props}
17:     />
18:   );
19: });
20:
21: Textarea.displayName = "Textarea";
22:
23: export { Textarea };

```

## ■ File: components/ui/toast.tsx

```

1: "use client";
2:
3: import * as React from "react";
4: import * as ToastPrimitives from "@radix-ui/react-toast";
5: import { cva, type VariantProps } from "class-variance-authority";
6: import { CheckCircle, X, AlertTriangle, Loader2 } from "lucide-react";
7: import { cn } from "@/lib/utils";

```

```

8:
9: const ToastProvider = ToastPrimitives.Provider;
10:
11: const ToastViewport = React.forwardRef<
12:   React.ElementRef<typeof ToastPrimitives.Viewport>,
13:   React.ComponentPropsWithoutRef<typeof ToastPrimitives.Viewport>
14: >(({ className, ...props }, ref) => (
15:   <ToastPrimitives.Viewport
16:     ref={ref}
17:     className={cn(
18:       "fixed z-[100] flex w-full max-w-sm flex-col space-y-2",
19:       // mobile: top center
20:       "top-2 left-1/2 -translate-x-1/2",
21:       // sm and up: top right
22:       "sm:top-2 sm:right-2 sm:left-auto sm:translate-x-0",
23:       className
24:     )}
25:     {...props}
26:   />
27: ));
28:
29: ToastViewport.displayName = ToastPrimitives.Viewport.displayName;
30:
31: const toastVariants = cva(
32:   "group pointer-events-auto relative flex w-full items-start justify-between space-x-4 overflow-hidden rounded-md border p-4",
33:   {
34:     variants: {
35:       variant: {
36:         default: "bg-white text-black border-gray-200",
37:         success: "bg-green-500 text-white border-green-500",
38:         warning: "bg-yellow-400 text-black border-yellow-500",
39:         destructive: "bg-red-500 text-white border-red-500",
40:         loading: "bg-gray-100 text-gray-800 border-gray-300",
41:       },
42:     },
43:     defaultVariants: {
44:       variant: "default",
45:     },
46:   },
47: );
48:
49: const variantIcons = {
50:   warning: <AlertTriangle className="h-5 w-5 text-black shrink-0 mt-1" />,
51:   destructive: <X className="h-5 w-5 text-white shrink-0 mt-1" />,
52:   loading: (
53:     <Loader2 className="h-5 w-5 animate-spin text-gray-600 shrink-0 mt-1" />
54:   ),
55: };
56:
57: type CustomToastProps = {
58:   hideClose?: boolean;
59: } & React.ComponentPropsWithoutRef<typeof ToastPrimitives.Root> &
60:   VariantProps<typeof toastVariants>;
61:
62: const Toast = React.forwardRef<
63:   React.ElementRef<typeof ToastPrimitives.Root>,
64:   CustomToastProps
65: >(({ className, variant, hideClose, children, ...props }, ref) => {
66:   return (
67:     <ToastPrimitives.Root
68:       ref={ref}
69:       className={cn(toastVariants({ variant }), className)}
70:       duration={3000}
71:       {...props}
72:     >
73:       {children}
74:       {!hideClose && (
75:         <ToastPrimitives.Close className="absolute right-2 top-2 rounded-md p-1 text-foreground/50 opacity-100 hover:opacity-80 focus:outline-none focus:ring-2 focus:ring-foreground focus:ring-offset-2" />
76:         <X className="h-4 w-4" />
77:       )}
78:     </ToastPrimitives.Root>
79:   );
80: });

```

```

81: });
82: Toast.displayName = ToastPrimitives.Root.displayName;
83:
84: const ToastAction = React.forwardRef<
85:   React.ElementRef<typeof ToastPrimitives.Action>,
86:   React.ComponentPropsWithoutRef<typeof ToastPrimitives.Action>
87: >(({ className, ...props }, ref) => (
88:   <ToastPrimitives.Action
89:     ref={ref}
90:     className={cn(
91:       "inline-flex h-8 shrink-0 items-center justify-center rounded-md border bg-transparent px-3 text-sm f
92:       className
93:     )}
94:     {...props}
95:   />
96: ));
97: ToastAction.displayName = ToastPrimitives.Action.displayName;
98:
99: const ToastClose = React.forwardRef<
100:   React.ElementRef<typeof ToastPrimitives.Close>,
101:   React.ComponentPropsWithoutRef<typeof ToastPrimitives.Close>
102: >(({ className, ...props }, ref) => (
103:   <ToastPrimitives.Close
104:     ref={ref}
105:     className={cn(
106:       "absolute right-2 top-2 rounded-md p-1 text-foreground/50 opacity-0 transition-opacity hover:text-for
107:       className
108:     )}
109:     {...props}
110:   >
111:     <X className="h-4 w-4" />
112:   </ToastPrimitives.Close>
113: ));
114: ToastClose.displayName = ToastPrimitives.Close.displayName;
115:
116: const ToastTitle = React.forwardRef<
117:   React.ElementRef<typeof ToastPrimitives.Title>,
118:   React.ComponentPropsWithoutRef<typeof ToastPrimitives.Title>
119: >(({ className, ...props }, ref) => (
120:   <ToastPrimitives.Title
121:     ref={ref}
122:     className={cn("text-sm font-semibold", className)}
123:     {...props}
124:   />
125: ));
126: ToastTitle.displayName = ToastPrimitives.Title.displayName;
127:
128: const ToastDescription = React.forwardRef<
129:   React.ElementRef<typeof ToastPrimitives.Description>,
130:   React.ComponentPropsWithoutRef<typeof ToastPrimitives.Description>
131: >(({ className, ...props }, ref) => (
132:   <ToastPrimitives.Description
133:     ref={ref}
134:     className={cn("text-sm opacity-90", className)}
135:     {...props}
136:   />
137: ));
138: ToastDescription.displayName = ToastPrimitives.Description.displayName;
139:
140: type ToastProps = React.ComponentPropsWithoutRef<typeof Toast>;
141: type ToastActionElement = React.ReactElement<typeof ToastAction>;
142:
143: export {
144:   ToastProvider,
145:   ToastViewport,
146:   Toast,
147:   ToastTitle,
148:   ToastDescription,
149:   ToastAction,
150:   ToastClose,
151:   type ToastProps,
152:   type ToastActionElement,
153:   variantIcons,

```

```
154: };
```

---

## ■ File: components\ui\toaster.tsx

---

```
1: "use client";
2:
3: import {
4:   Toast,
5:   ToastDescription,
6:   ToastProvider,
7:   ToastTitle,
8:   ToastViewport,
9:   variantIcons,
10: } from "@components/ui/toast";
11: import { useToast } from "@components/ui/use-toast";
12:
13: export function Toaster() {
14:   const { toasts } = useToast();
15:
16:   return (
17:     <ToastProvider swipeDirection="up">
18:       {toasts.map(
19:         ({ id, title, description, action, variant = "default", ...props }) => (
20:           <Toast key={id} variant={variant} {...props} hideClose>
21:             <div className="flex items-start gap-3">
22:               {variant !== "default" && variantIcons[variant]}
23:               <div className="grid gap-1">
24:                 {title && <ToastTitle>{title}</ToastTitle>}
25:                 {description && (
26:                   <ToastDescription>{description}</ToastDescription>
27:                 )}
28:               </div>
29:             </div>
30:             {action}
31:           </Toast>
32:         )
33:       )}
34:     <ToastViewport />
35:   </ToastProvider>
36: );
37: }
```

---

## ■ File: components\ui\toggle-group.tsx

---

```
1: "use client"
2:
3: import * as React from "react"
4: import * as ToggleGroupPrimitive from "@radix-ui/react-toggle-group"
5: import { type VariantProps } from "class-variance-authority"
6:
7: import { cn } from "@lib/utils"
8: import { toggleVariants } from "@components/ui/toggle"
9:
10: const ToggleGroupContext = React.createContext<
11:   VariantProps<typeof toggleVariants>
12: >({
13:   size: "default",
14:   variant: "default",
15: })
16:
17: const ToggleGroup = React.forwardRef<
18:   React.ElementRef<typeof ToggleGroupPrimitive.Root>,
19:   React.ComponentPropsWithoutRef<typeof ToggleGroupPrimitive.Root> &
20:   VariantProps<typeof toggleVariants>
21: >(({ className, variant, size, children, ...props }, ref) => (
22:   <ToggleGroupPrimitive.Root
23:     ref={ref}
24:     className={cn("flex items-center justify-center gap-1", className)}
25:     {...props}
```

```

26:   >
27:   <ToggleGroupContext.Provider value={{ variant, size }}>
28:     {children}
29:   </ToggleGroupContext.Provider>
30: </ToggleGroupPrimitive.Root>
31: ))
32:
33: ToggleGroup.displayName = ToggleGroupPrimitive.Root.displayName
34:
35: const ToggleGroupItem = React.forwardRef<
36:   React.ElementRef<typeof ToggleGroupPrimitive.Item>,
37:   React.ComponentPropsWithoutRef<typeof ToggleGroupPrimitive.Item> &
38:     VariantProps<typeof toggleVariants>
39: >(({ className, children, variant, size, ...props }, ref) => {
40:   const context = React.useContext(ToggleGroupContext)
41:
42:   return (
43:     <ToggleGroupPrimitive.Item
44:       ref={ref}
45:       className={cn(
46:         toggleVariants({
47:           variant: context.variant || variant,
48:           size: context.size || size,
49:         }),
50:         className
51:       )}
52:       {...props}
53:     >
54:       {children}
55:     </ToggleGroupPrimitive.Item>
56:   )
57: })
58:
59: ToggleGroupItem.displayName = ToggleGroupPrimitive.Item.displayName
60:
61: export { ToggleGroup, ToggleGroupItem }

```

## ■ File: components/ui/toggle.tsx

```

1: "use client"
2:
3: import * as React from "react"
4: import * as TogglePrimitive from "@radix-ui/react-toggle"
5: import { cva, type VariantProps } from "class-variance-authority"
6:
7: import { cn } from "@lib/Utils"
8:
9: const toggleVariants = cva(
10:   "inline-flex items-center justify-center rounded-md text-sm font-medium ring-offset-background transition-colors",
11:   {
12:     variants: {
13:       variant: {
14:         default: "bg-transparent",
15:         outline:
16:           "border border-input bg-transparent hover:bg-accent hover:text-accent-foreground",
17:       },
18:       size: {
19:         default: "h-10 px-3 min-w-10",
20:         sm: "h-9 px-2.5 min-w-9",
21:         lg: "h-11 px-5 min-w-11",
22:       },
23:     },
24:     defaultVariants: {
25:       variant: "default",
26:       size: "default",
27:     },
28:   }
29: )
30:
31: const Toggle = React.forwardRef<
32:   React.ElementRef<typeof TogglePrimitive.Root>,

```

```

33:   React.ComponentPropsWithoutRef<typeof TogglePrimitive.Root> &
34:     VariantProps<typeof toggleVariants>
35: >(({ className, variant, size, ...props }, ref) => (
36:   <TogglePrimitive.Root
37:     ref={ref}
38:     className={cn(toggleVariants({ variant, size, className })))}
39:     {...props}
40:   />
41: ))
42:
43: Toggle.displayName = TogglePrimitive.Root.displayName
44:
45: export { Toggle, toggleVariants }

```

## ■ File: components\ui\tooltip.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import * as TooltipPrimitive from "@radix-ui/react-tooltip"
5:
6: import { cn } from "@/lib/utils"
7:
8: const TooltipProvider = TooltipPrimitive.Provider
9:
10: const Tooltip = TooltipPrimitive.Root
11:
12: const TooltipTrigger = TooltipPrimitive.Trigger
13:
14: const TooltipContent = React.forwardRef<
15:   React.ElementRef<typeof TooltipPrimitive.Content>,
16:   React.ComponentPropsWithoutRef<typeof TooltipPrimitive.Content>
17: >(({ className, sideOffset = 4, ...props }, ref) => (
18:   <TooltipPrimitive.Content
19:     ref={ref}
20:     sideOffset={sideOffset}
21:     className={cn(
22:       "z-50 overflow-hidden rounded-md border bg-popover px-3 py-1.5 text-sm text-popover-foreground shadow
23:       className
24:     )}
25:     {...props}
26:   />
27: ))
28: TooltipContent.displayName = TooltipPrimitive.Content.displayName
29:
30: export { Tooltip, TooltipTrigger, TooltipContent, TooltipProvider }

```

## ■ File: components\ui\use-mobile.tsx

```

=====
1: import * as React from "react"
2:
3: const MOBILE_BREAKPOINT = 768
4:
5: export function useIsMobile() {
6:   const [isMobile, setIsMobile] = React.useState<boolean | undefined>(undefined)
7:
8:   React.useEffect(() => {
9:     const mql = window.matchMedia(`(max-width: ${MOBILE_BREAKPOINT - 1}px)`)
10:    const onChange = () => {
11:      setIsMobile(window.innerWidth < MOBILE_BREAKPOINT)
12:    }
13:    mql.addEventListener("change", onChange)
14:    setIsMobile(window.innerWidth < MOBILE_BREAKPOINT)
15:    return () => mql.removeEventListener("change", onChange)
16:  }, [])
17:
18:   return !!isMobile
19: }

```



## ■ File: components\ui\use-toast.ts

```
=====
1: "use client";
2:
3: import { useEffect } from "react";
4: import type React from "react";
5: import { create } from "zustand";
6: import type { ToastProps } from "../toast";
7:
8: type Toast = ToastProps & {
9:   id: string;
10:  title?: React.ReactNode;
11:  description?: React.ReactNode;
12:  action?: React.ReactNode;
13: };
14:
15: const TOAST_LIMIT = 1;
16: const TOAST_REMOVE_DELAY = 100;
17:
18: type State = {
19:   toasts: Toast[];
20: };
21:
22: type Actions = {
23:   addToast: (toast: Toast) => void;
24:   updateToast: (id: string, toast: Partial<Toast>) => void;
25:   dismissToast: (id?: string) => void;
26:   removeToast: (id?: string) => void;
27: };
28:
29: const toastStore = create<State & Actions>((set, get) => ({
30:   toasts: [],
31:   addToast: (toast) => {
32:     set((state) => ({
33:       toasts: [toast, ...state.toasts].slice(0, TOAST_LIMIT),
34:     }));
35:   },
36:   updateToast: (id, toast) => {
37:     set((state) => ({
38:       toasts: state.toasts.map((t) => (t.id === id ? { ...t, ...toast } : t)),
39:     }));
40:   },
41:   dismissToast: (id) => {
42:     set((state) => ({
43:       toasts: state.toasts.map((t) =>
44:         t.id === id || id === undefined ? { ...t, open: false } : t
45:       ),
46:     }));
47:   },
48:   removeToast: (id) => {
49:     set((state) => ({
50:       toasts: state.toasts.filter((t) => t.id !== id),
51:     }));
52:   },
53: }));
54:
55: let count = 0;
56:
57: function genId() {
58:   count = (count + 1) % Number.MAX_SAFE_INTEGER;
59:   return count.toString();
60: }
61:
62: type ToastOptions = Omit<Toast, "id">;
63:
64: function toast(opts: ToastOptions) {
65:   const id = genId();
66:
67:   const update = (newOpts: Partial<Toast>) =>
68:     toastStore.getState().updateToast(id, newOpts);
69:   const dismiss = () => toastStore.getState().dismissToast(id);
70:
71:   toastStore.getState().addToast({
```

```

72:     id,
73:     open: true,
74:     ...opts,
75:   });
76:
77:   if (opts.duration !== Number.POSITIVE_INFINITY) {
78:     setTimeout(() => dismiss(), opts.duration || TOAST_REMOVE_DELAY);
79:   }
80:
81:   return {
82:     id,
83:     update,
84:     dismiss,
85:   };
86: }
87:
88: export function useToast() {
89:   const { toasts, addToast, updateToast, dismissToast, removeToast } =
90:     toastStore();
91:
92:   useEffect(() => {
93:     const unsub = toastStore.subscribe((state) => {
94:       state.toasts.forEach((t) => {
95:         if (t.open === false) {
96:           setTimeout(() => removeToast(t.id), 400);
97:         }
98:       });
99:     });
100:     return () => unsub();
101:   }, [removeToast]);
102:
103:   return {
104:     toasts,
105:     toast,
106:     dismiss: dismissToast,
107:   };
108: }

```

## ■ File: hooks\use-mobile.tsx

```

1: import * as React from "react"
2:
3: const MOBILE_BREAKPOINT = 768
4:
5: export function useIsMobile() {
6:   const [isMobile, setIsMobile] = React.useState<boolean | undefined>(undefined)
7:
8:   React.useEffect(() => {
9:     const mql = window.matchMedia(`(max-width: ${MOBILE_BREAKPOINT - 1}px)`);
10:    const onChange = () => {
11:      setIsMobile(window.innerWidth < MOBILE_BREAKPOINT)
12:    }
13:    mql.addEventListener("change", onChange)
14:    setIsMobile(window.innerWidth < MOBILE_BREAKPOINT)
15:    return () => mql.removeEventListener("change", onChange)
16:  }, [])
17:
18:   return !!isMobile
19: }

```

## ■ File: hooks\use-toast.ts

```

1: "use client"
2:
3: // Inspired by react-hot-toast library
4: import * as React from "react"
5:
6: import type {
7:   ToastActionElement,

```

```

8:   ToastProps,
9: } from "@components/ui/toast"
10:
11: const TOAST_LIMIT = 1
12: const TOAST_REMOVE_DELAY = 1000000
13:
14: type ToasterToast = ToastProps & {
15:   id: string
16:   title?: React.ReactNode
17:   description?: React.ReactNode
18:   action?: ToastActionElement
19: }
20:
21: const actionTypes = {
22:   ADD_TOAST: "ADD_TOAST",
23:   UPDATE_TOAST: "UPDATE_TOAST",
24:   DISMISS_TOAST: "DISMISS_TOAST",
25:   REMOVE_TOAST: "REMOVE_TOAST",
26: } as const
27:
28: let count = 0
29:
30: function genId() {
31:   count = (count + 1) % Number.MAX_SAFE_INTEGER
32:   return count.toString()
33: }
34:
35: type ActionType = typeof actionTypes
36:
37: type Action =
38:   | {
39:     type: ActionType["ADD_TOAST"]
40:     toast: ToasterToast
41:   }
42:   | {
43:     type: ActionType["UPDATE_TOAST"]
44:     toast: Partial<ToasterToast>
45:   }
46:   | {
47:     type: ActionType["DISMISS_TOAST"]
48:     toastId?: ToasterToast["id"]
49:   }
50:   | {
51:     type: ActionType["REMOVE_TOAST"]
52:     toastId?: ToasterToast["id"]
53:   }
54:
55: interface State {
56:   toasts: ToasterToast[]
57: }
58:
59: const toastTimeouts = new Map<string, ReturnType<typeof setTimeout>>()
60:
61: const addToRemoveQueue = (toastId: string) => {
62:   if (toastTimeouts.has(toastId)) {
63:     return
64:   }
65:
66:   const timeout = setTimeout(() => {
67:     toastTimeouts.delete(toastId)
68:     dispatch({
69:       type: "REMOVE_TOAST",
70:       toastId: toastId,
71:     })
72:   }, TOAST_REMOVE_DELAY)
73:
74:   toastTimeouts.set(toastId, timeout)
75: }
76:
77: export const reducer = (state: State, action: Action): State => {
78:   switch (action.type) {
79:     case "ADD_TOAST":
80:       return {

```

```

81:         ...state,
82:         toasts: [action.toast, ...state.toasts].slice(0, TOAST_LIMIT),
83:     }
84:
85:     case "UPDATE_TOAST":
86:         return {
87:             ...state,
88:             toasts: state.toasts.map((t) =>
89:                 t.id === action.toast.id ? { ...t, ...action.toast } : t
90:             ),
91:         }
92:
93:     case "DISMISS_TOAST": {
94:         const { toastId } = action
95:
96:         // ! Side effects ! - This could be extracted into a dismissToast() action,
97:         // but I'll keep it here for simplicity
98:         if (toastId) {
99:             addToRemoveQueue(toastId)
100:         } else {
101:             state.toasts.forEach((toast) => {
102:                 addToRemoveQueue(toast.id)
103:             })
104:         }
105:
106:         return {
107:             ...state,
108:             toasts: state.toasts.map((t) =>
109:                 t.id === toastId || toastId === undefined
110:                 ? {
111:                     ...t,
112:                     open: false,
113:                 }
114:                 : t
115:             ),
116:         }
117:     }
118:     case "REMOVE_TOAST":
119:         if (action.toastId === undefined) {
120:             return {
121:                 ...state,
122:                 toasts: [],
123:             }
124:         }
125:         return {
126:             ...state,
127:             toasts: state.toasts.filter((t) => t.id !== action.toastId),
128:         }
129:     }
130: }
131:
132: const listeners: Array<(state: State) => void> = []
133:
134: let memoryState: State = { toasts: [] }
135:
136: function dispatch(action: Action) {
137:     memoryState = reducer(memoryState, action)
138:     listeners.forEach((listener) => {
139:         listener(memoryState)
140:     })
141: }
142:
143: type Toast = Omit<ToasterToast, "id">
144:
145: function toast({ ...props }: Toast) {
146:     const id = genId()
147:
148:     const update = (props: ToasterToast) =>
149:         dispatch({
150:             type: "UPDATE_TOAST",
151:             toast: { ...props, id },
152:         })
153:     const dismiss = () => dispatch({ type: "DISMISS_TOAST", toastId: id })

```

```

154:
155:   dispatch({
156:     type: "ADD_TOAST",
157:     toast: {
158:       ...props,
159:       id,
160:       open: true,
161:       onOpenChange: (open) => {
162:         if (!open) dismiss()
163:       },
164:     },
165:   })
166:
167:   return {
168:     id: id,
169:     dismiss,
170:     update,
171:   }
172: }
173:
174: function useToast() {
175:   const [state, setState] = React.useState<State>(memoryState)
176:
177:   React.useEffect(() => {
178:     listeners.push(setState)
179:     return () => {
180:       const index = listeners.indexOf(setState)
181:       if (index > -1) {
182:         listeners.splice(index, 1)
183:       }
184:     }
185:   }, [state])
186:
187:   return {
188:     ...state,
189:     toast,
190:     dismiss: (toastId?: string) => dispatch({ type: "DISMISS_TOAST", toastId }),
191:   }
192: }
193:
194: export { useToast, toast }

```

## ■ File: lib/encryption.ts

```

1: import crypto from "crypto";
2:
3: // Use environment variable for encryption key
4: const ENCRYPTION_KEY =
5:   process.env.ENCRYPTION_KEY || crypto.randomBytes(32).toString("hex");
6: const ALGORITHM = "aes-256-gcm";
7: const IV_LENGTH = 16;
8: const SALT_LENGTH = 64;
9: const TAG_LENGTH = 16;
10: const TAG_POSITION = SALT_LENGTH + IV_LENGTH;
11: const ENCRYPTED_POSITION = TAG_POSITION + TAG_LENGTH;
12:
13: interface EncryptedData {
14:   encrypted: string;
15:   iv: string;
16:   tag: string;
17:   salt: string;
18: }
19:
20: /**
21:  * Encrypt data using AES-256-GCM
22:  */
23: export function encrypt(text: string): string {
24:   const key = Buffer.from(ENCRYPTION_KEY, "hex");
25:   const iv = crypto.randomBytes(IV_LENGTH);
26:   const salt = crypto.randomBytes(SALT_LENGTH);
27:

```

```

28:   const cipher = crypto.createCipheriv(ALGORITHM, key, iv);
29:
30:   let encrypted = cipher.update(text, "utf8", "hex");
31:   encrypted += cipher.final("hex");
32:
33:   const tag = cipher.getAuthTag();
34:
35:   // Combine salt + iv + tag + encrypted
36:   const result = Buffer.concat([salt, iv, tag, Buffer.from(encrypted, "hex")]);
37:
38:   return result.toString("base64url"); // URL-safe base64
39: }
40:
41: /**
42:  * Decrypt data using AES-256-GCM
43:  */
44: export function decrypt(encryptedData: string): string {
45:   const key = Buffer.from(ENCRYPTION_KEY, "hex");
46:   const buffer = Buffer.from(encryptedData, "base64url");
47:
48:   const salt = buffer.subarray(0, SALT_LENGTH);
49:   const iv = buffer.subarray(SALT_LENGTH, TAG_POSITION);
50:   const tag = buffer.subarray(TAG_POSITION, ENCRYPTED_POSITION);
51:   const encrypted = buffer.subarray(ENCRYPTED_POSITION);
52:
53:   const decipher = crypto.createDecipheriv(ALGORITHM, key, iv);
54:   decipher.setAuthTag(tag);
55:
56:   let decrypted = decipher.update(encrypted);
57:   decrypted = Buffer.concat([decrypted, decipher.final()]);
58:
59:   return decrypted.toString("utf8");
60: }
61:
62: /**
63:  * Hash password using bcrypt-like approach
64:  */
65: export async function hashPassword(password: string): Promise<string> {
66:   const salt = crypto.randomBytes(16).toString("hex");
67:
68:   return new Promise((resolve, reject) => {
69:     crypto.pbkdf2(password, salt, 100000, 64, "sha512", (err, derivedKey) => {
70:       if (err) reject(err);
71:       resolve(salt + ":" + derivedKey.toString("hex"));
72:     });
73:   });
74: }
75:
76: /**
77:  * Verify password against hash
78:  */
79: export async function verifyPassword(
80:   password: string,
81:   hash: string
82: ): Promise<boolean> {
83:   const [salt, key] = hash.split(":");
84:
85:   return new Promise((resolve, reject) => {
86:     crypto.pbkdf2(password, salt, 100000, 64, "sha512", (err, derivedKey) => {
87:       if (err) reject(err);
88:       resolve(key === derivedKey.toString("hex"));
89:     });
90:   });
91: }
92:
93: /**
94:  * Generate a secure random token
95:  */
96: export function generateSecureToken(): string {
97:   return crypto.randomBytes(32).toString("base64url");
98: }
99:
100: /**

```

```

101: * Create share token payload
102: */
103: export function createSharePayload(
104:   reportId: string,
105:   patientId: string
106: ): string {
107:   const payload = JSON.stringify({
108:     reportId,
109:     patientId,
110:     timestamp: Date.now(),
111:     nonce: crypto.randomBytes(16).toString("hex"),
112:   });
113:
114:   return encrypt(payload);
115: }
116:
117: /**
118:  * Parse share token payload
119:  */
120: export function parseSharePayload(
121:   token: string
122: ): { reportId: string; patientId: string } | null {
123:   try {
124:     const decrypted = decrypt(token);
125:     const parsed = JSON.parse(decrypted);
126:
127:     return {
128:       reportId: parsed.reportId,
129:       patientId: parsed.patientId,
130:     };
131:   } catch (error) {
132:     console.error("Failed to parse share token:", error);
133:     return null;
134:   }
135: }

```

## ■ File: lib/prisma.ts

```

=====
1: // lib/prisma.ts
2: import { PrismaClient } from "@prisma/client";
3:
4: const globalForPrisma = globalThis as unknown as {
5:   prisma: PrismaClient | undefined;
6: };
7:
8: export const prisma = globalForPrisma.prisma ?? new PrismaClient();
9:
10: if (process.env.NODE_ENV !== "production") globalForPrisma.prisma = prisma;

```

## ■ File: lib/utls.ts

```

=====
1: import { type ClassValue, clsx } from "clsx"
2: import { twMerge } from "tailwind-merge"
3:
4: export function cn(...inputs: ClassValue[]) {
5:   return twMerge(clsx(inputs))
6: }

```

## ■ File: next-env.d.ts

```

=====
1: /// <reference types="next" />
2: /// <reference types="next/image-types/global" />
3:
4: // NOTE: This file should not be edited
5: // see https://nextjs.org/docs/app/api-reference/config/typescript for more information.

```

## ■ File: package.json

```
=====
1: {
2:   "name": "report",
3:   "version": "0.1.0",
4:   "private": true,
5:   "scripts": {
6:     "build": "next build",
7:     "dev": "next dev",
8:     "lint": "next lint",
9:     "start": "next start",
10:    "export:pdf": "ts-node scripts/generate-pdf.ts"
11:  },
12:  "dependencies": {
13:    "@headlessui/react": "^2.2.7",
14:    "@hello-pangea/dnd": "latest",
15:    "@heroicons/react": "^2.2.0",
16:    "@hookform/resolvers": "^3.9.1",
17:    "@prisma/client": "^6.13.0",
18:    "@radix-ui/react-accordion": "1.2.2",
19:    "@radix-ui/react-alert-dialog": "1.1.4",
20:    "@radix-ui/react-aspect-ratio": "1.1.1",
21:    "@radix-ui/react-avatar": "1.1.2",
22:    "@radix-ui/react-checkbox": "1.1.3",
23:    "@radix-ui/react-collapsible": "1.1.2",
24:    "@radix-ui/react-context-menu": "2.2.4",
25:    "@radix-ui/react-dialog": "1.1.4",
26:    "@radix-ui/react-dropdown-menu": "2.1.4",
27:    "@radix-ui/react-hover-card": "1.1.4",
28:    "@radix-ui/react-label": "2.1.1",
29:    "@radix-ui/react-menubar": "1.1.4",
30:    "@radix-ui/react-navigation-menu": "1.2.3",
31:    "@radix-ui/react-popover": "1.1.4",
32:    "@radix-ui/react-progress": "1.1.1",
33:    "@radix-ui/react-radio-group": "1.2.2",
34:    "@radix-ui/react-scroll-area": "1.2.2",
35:    "@radix-ui/react-select": "2.1.4",
36:    "@radix-ui/react-separator": "1.1.1",
37:    "@radix-ui/react-slider": "1.2.2",
38:    "@radix-ui/react-slot": "1.1.1",
39:    "@radix-ui/react-switch": "1.1.2",
40:    "@radix-ui/react-tabs": "1.1.2",
41:    "@radix-ui/react-toast": "latest",
42:    "@radix-ui/react-toggle": "1.1.1",
43:    "@radix-ui/react-toggle-group": "1.1.1",
44:    "@radix-ui/react-tooltip": "1.1.6",
45:    "@react-pdf/renderer": "^4.3.0",
46:    "autoprefixer": "^10.4.20",
47:    "class-variance-authority": "^0.7.1",
48:    "clsx": "^2.1.1",
49:    "cmdk": "1.0.4",
50:    "date-fns": "4.1.0",
51:    "embla-carousel-react": "8.5.1",
52:    "fs": "latest",
53:    "html2canvas": "^1.4.1",
54:    "html2pdf.js": "^0.10.3",
55:    "immer": "latest",
56:    "input-otp": "1.4.1",
57:    "jspdf": "^3.0.1",
58:    "lucide-react": "^0.454.0",
59:    "next": "15.2.4",
60:    "next-themes": "^0.4.6",
61:    "path": "latest",
62:    "puppeteer": "^24.16.2",
63:    "react": "^19",
64:    "react-day-picker": "8.10.1",
65:    "react-dom": "^19",
66:    "react-hook-form": "^7.54.1",
67:    "react-resizable-panels": "^2.1.7",
68:    "react-to-print": "^3.1.1",
69:    "recharts": "2.15.0",
70:    "sonner": "^1.7.1",
71:    "tailwind-merge": "^2.5.5",
=====
```



```

72:     "tailwindcss-animate": "^1.0.7",
73:     "use-sync-external-store": "latest",
74:     "uuid": "latest",
75:     "vaul": "^0.9.6",
76:     "zod": "^3.24.1",
77:     "zustand": "latest"
78:   },
79:   "devDependencies": {
80:     "@types/node": "^22",
81:     "@types/react": "^19",
82:     "@types/react-dom": "^19",
83:     "postcss": "^8.5",
84:     "prisma": "^6.13.0",
85:     "tailwindcss": "^3.4.17",
86:     "ts-node": "^10.9.2",
87:     "typescript": "^5.9.2"
88:   }
89: }

```

## ■ File: scripts\migrate-data.ts

```

=====
1: const { PrismaClient } = require("@prisma/client");
2: const fs = require("fs");
3: const path = require("path");
4:
5: const prisma = new PrismaClient();
6:
7: async function migrateJsonToDB() {
8:   try {
9:     console.log("? Starting migration from JSON to MySQL...\n");
10:
11:     // Try multiple possible locations for your data
12:     const possiblePaths = [
13:       path.join(process.cwd(), "data", "patients.json"),
14:       path.join(process.cwd(), "data.json"),
15:       path.join(process.cwd(), "patients.json"),
16:       path.join(process.cwd(), "src", "data", "patients.json"),
17:       path.join(process.cwd(), "public", "data", "patients.json"),
18:     ];
19:
20:     let jsonPath = null;
21:     let jsonData = null;
22:
23:     // Check each possible path
24:     for (const pathToCheck of possiblePaths) {
25:       if (fs.existsSync(pathToCheck)) {
26:         jsonPath = pathToCheck;
27:         console.log(`? Found data file at: ${jsonPath}`);
28:         break;
29:       }
30:     }
31:
32:     // If no file found, create sample data for testing
33:     if (!jsonPath) {
34:       console.log(
35:         "?? No existing data file found. Creating sample data for testing..."
36:       );
37:       jsonData = createSampleData();
38:     } else {
39:       try {
40:         const fileContent = fs.readFileSync(jsonPath, "utf8");
41:         jsonData = JSON.parse(fileContent);
42:       } catch (parseError) {
43:         console.error("? Error parsing JSON file:", parseError.message);
44:         console.log("Creating sample data instead...");
45:         jsonData = createSampleData();
46:       }
47:     }
48:
49:     if (!Array.isArray(jsonData)) {
50:       console.log("Data is not an array, wrapping in array...");

```

```

51:     jsonData = [jsonData];
52: }
53:
54: console.log(`? Found ${jsonData.length} patient(s) to migrate...\n`);
55:
56: let migratedCount = 0;
57:
58: for (let i = 0; i < jsonData.length; i++) {
59:     const patient = jsonData[i];
60:     console.log(
61:         `? Migrating patient ${i + 1}/${jsonData.length}: ${
62:             patient.info?.name || patient.info?.sampleCode || "Unknown"
63:         }`
64:     );
65:
66:     try {
67:         // Create patient with safe defaults
68:         const createdPatient = await prisma.patient.create({
69:             data: {
70:                 name: patient.info?.name || "",
71:                 gender: patient.info?.gender || "MALE",
72:                 birthDate: patient.info?.birthDate || "",
73:                 sampleCode: patient.info?.sampleCode || `SAMPLE_${Date.now()}_${i}`,
74:                 sampleDate: patient.info?.sampleDate || "",
75:                 reportDate: patient.info?.reportDate || "",
76:                 checkedBy: patient.info?.checkedBy || "",
77:                 scientificContent: patient.info?.scientificContent || "",
78:                 disclaimer: patient.info?.disclaimer || "",
79:                 signature1: patient.info?.signature1 || null,
80:                 signature2: patient.info?.signature2 || null,
81:             },
82:         });
83:
84:         // Create reports for this patient
85:         const reports = patient.reports || [{}]; // Default to one empty report if none exist
86:
87:         for (let j = 0; j < reports.length; j++) {
88:             const report = reports[j];
89:             console.log(`? Creating report ${j + 1}/${reports.length}...\n`);
90:
91:             const createdReport = await prisma.report.create({
92:                 data: {
93:                     patientId: createdPatient.id,
94:                     // Content
95:                     introduction: report.content?.introduction || "",
96:                     genomicsExplanation: report.content?.genomicsExplanation || "",
97:                     genesHealthImpact: report.content?.genesHealthImpact || "",
98:                     fundamentalsPRS: report.content?.fundamentalsPRS || "",
99:                     utilityDoctors: report.content?.utilityDoctors || "",
100:                     microarrayExplanation:
101:                         report.content?.microarrayExplanation || "",
102:                     microarrayData: report.content?.microarrayData || "",
103:                     // Settings
104:                     title: report.settings?.title || "",
105:                     subtitle: report.settings?.subtitle || "",
106:                     companyName: report.settings?.companyName || "",
107:                     headerColor: report.settings?.headerColor || "#000000",
108:                     accentColor: report.settings?.accentColor || "#000000",
109:                     primaryFont:
110:                         report.settings?.fonts?.primary || "Arial, sans-serif",
111:                     secondaryFont:
112:                         report.settings?.fonts?.secondary || "Georgia, serif",
113:                     monoFont:
114:                         report.settings?.fonts?.mono || "Courier New, monospace",
115:                     // Quotes and descriptions
116:                     nutritionQuote: report.nutritionData?.quote || null,
117:                     nutritionDescription: report.nutritionData?.description || null,
118:                     lifestyleQuote:
119:                         typeof report.lifestyleConditions?.quote === "string"
120:                             ? report.lifestyleConditions?.quote
121:                             : null,
122:                     lifestyleDescription:
123:                         typeof report.lifestyleConditions?.description === "string"

```

```

124:         ? report.lifestyleConditions.description
125:         : null,
126:     metabolicQuote:
127:         typeof report.metabolicCore?.quote === "string"
128:         ? report.metabolicCore.quote
129:         : null,
130:     metabolicDescription:
131:         typeof report.metabolicCore?.description === "string"
132:         ? report.metabolicCore.description
133:         : null,
134:     digestiveQuote: report.digestiveHealth?.quote || null,
135:     digestiveDescription: report.digestiveHealth?.description || null,
136:     addictionQuote: report.genesAndAddiction?.quote || null,
137:     addictionDescription:
138:         report.genesAndAddiction?.description || null,
139:     sleepQuote: report.sleepAndRest?.quote || null,
140:     sleepDescription: report.sleepAndRest?.description || null,
141:     allergyQuote: report.allergiesAndSensitivity?.quote || "",
142:     allergyDescription:
143:         report.allergiesAndSensitivity?.description || "",
144:     allergyGeneralAdvice:
145:         report.allergiesAndSensitivity?.generalAdvice || "",
146:     // Summaries
147:     nutrigenomicsSummary:
148:         report.summaries?.nutrigenomicsSummary || "",
149:     exerciseGenomicsSummary:
150:         report.summaries?.exerciseGenomicsSummary || "",
151:     // Metabolic summary
152:     metabolicStrengths: report.metabolicSummary?.strengths || [],
153:     metabolicWeaknesses: report.metabolicSummary?.weaknesses || [],
154:     // Diet categories
155:     dietFieldCategories: report.dietFieldCategories || [],
156: },
157: });
158:
159: // Migrate related data with progress indicators
160: await migrateReportData(createdReport.id, report);
161:
162: console.log(` ? Report ${j + 1} migrated successfully`);
163: }
164:
165: migratedCount++;
166: console.log(`? Patient ${i + 1} migrated successfully\n`);
167: } catch (patientError) {
168:     console.error(
169:         `? Error migrating patient ${i + 1}:`,
170:         patientError.message
171:     );
172:     if (patientError.code === "P2002") {
173:         console.log(
174:             " ? This appears to be a duplicate sample code. Skipping...\n"
175:         );
176:     }
177:     continue;
178: }
179: }
180:
181: console.log(`\n? Migration completed successfully!`);
182: console.log(
183:     `? Results: ${migratedCount}/${jsonData.length} patients migrated to the database.`
184: );
185: } catch (error) {
186:     console.error("? Migration failed:", error);
187: } finally {
188:     await prisma.$disconnect();
189: }
190: }
191:
192: async function migrateReportData(reportId, report) {
193:     try {
194:         // Migrate Dynamic Diet Fields
195:         if (
196:             report.dynamicDietFieldDefinitions &&

```

```

197:     Array.isArray(report.dynamicDietFieldDefinitions)
198:   ) {
199:     for (const field of report.dynamicDietFieldDefinitions) {
200:       await prisma.dynamicDietField.create({
201:         data: {
202:           reportId: reportId,
203:           uuid: field._uuid || `uuid_${Date.now()}_${Math.random()}`,
204:           fieldId: field.id || "",
205:           label: field.label || "",
206:           category: field.category || "",
207:           min: typeof field.min === "number" ? field.min : 0,
208:           max: typeof field.max === "number" ? field.max : 100,
209:           highRecommendation: field.highRecommendation || "",
210:           normalRecommendation: field.normalRecommendation || "",
211:           lowRecommendation: field.lowRecommendation || "",
212:           quote: field.quote || null,
213:           description: field.description || null,
214:         },
215:       });
216:     }
217:   }
218:
219:   // Migrate Patient Diet Analysis
220:   if (
221:     report.patientDietAnalysisResults &&
222:     Array.isArray(report.patientDietAnalysisResults)
223:   ) {
224:     for (const analysis of report.patientDietAnalysisResults) {
225:       await prisma.patientDietAnalysis.create({
226:         data: {
227:           reportId: reportId,
228:           fieldId: analysis.fieldId || "",
229:           score: typeof analysis.score === "number" ? analysis.score : 0,
230:           level: analysis.level || "NORMAL",
231:           recommendation: analysis.recommendation || "",
232:           lowRecommendation: analysis.recommendations?.LOW || "",
233:           normalRecommendation: analysis.recommendations?.NORMAL || "",
234:           highRecommendation: analysis.recommendations?.HIGH || "",
235:           selectedLevel: analysis.selectedLevel || "NORMAL",
236:         },
237:       });
238:     }
239:   }
240:
241:   // Migrate Nutrition Data
242:   if (
243:     report.nutritionData?.data &&
244:     typeof report.nutritionData.data === "object"
245:   ) {
246:     for (const [section, fields] of Object.entries(
247:       report.nutritionData.data
248:     )) {
249:       if (fields && typeof fields === "object") {
250:         for (const [field, data] of Object.entries(fields)) {
251:           if (data && typeof data === "object") {
252:             await prisma.nutritionData.create({
253:               data: {
254:                 reportId: reportId,
255:                 section,
256:                 field,
257:                 score: typeof data.score === "number" ? data.score : 0,
258:                 healthImpact: data.healthImpact || "",
259:                 intakeLevel: data.intakeLevel || "",
260:                 source: data.source || "",
261:               },
262:             });
263:           }
264:         }
265:       }
266:     }
267:   }
268:
269:   // Migrate Gene Test Results

```

```

270:     if (report.geneTestResults && Array.isArray(report.geneTestResults)) {
271:       for (const geneTest of report.geneTestResults) {
272:         await prisma.geneTestResult.create({
273:           data: {
274:             reportId: reportId,
275:             geneName: geneTest.geneName || "",
276:             variation: geneTest.variation || "",
277:             result: geneTest.result || "",
278:           },
279:         });
280:       }
281:     }
282:
283:     // Migrate Genetic Categories
284:     if (report.categories && Array.isArray(report.categories)) {
285:       for (const category of report.categories) {
286:         await prisma.geneticCategory.create({
287:           data: {
288:             reportId: reportId,
289:             categoryId: category.id || `cat_${Date.now()}_${Math.random()}`,
290:             category: category.category || "",
291:             imageUrl: category.imageUrl || "",
292:             description: category.description || "",
293:             parameters: Array.isArray(category.parameters)
294:               ? category.parameters
295:               : [],
296:             isActive:
297:               category.isActive !== undefined ? category.isActive : true,
298:             order: typeof category.order === "number" ? category.order : 0,
299:           },
300:         });
301:       }
302:     }
303:   } catch (error) {
304:     console.log(
305:       `?? Warning: Some report data could not be migrated:`,
306:       error.message
307:     );
308:   }
309: }
310:
311: function createSampleData() {
312:   return [
313:     {
314:       id: "sample-patient-1",
315:       info: {
316:         name: "John Doe",
317:         gender: "MALE",
318:         birthDate: "1990-01-15",
319:         sampleCode: "DNL0000000001",
320:         sampleDate: "2025-01-01",
321:         reportDate: "2025-01-15",
322:         checkedBy: "Lab Technician",
323:         scientificContent: "Dr. Johnson",
324:         disclaimer: "This is a sample disclaimer for genetic testing results.",
325:         signature1: null,
326:         signature2: null,
327:       },
328:       reports: [
329:         {
330:           id: "sample-report-1",
331:           content: {
332:             introduction: "Welcome to your genetic report.",
333:             genomicsExplanation: "Genomics is the study of your DNA.",
334:             genesHealthImpact: "Your genes affect your health in various ways.",
335:             fundamentalsPRS: "Polygenic Risk Scores explained.",
336:             utilityDoctors: "How doctors can use this information.",
337:             microarrayExplanation: "Microarray technology explanation.",
338:             microarrayData: "Your microarray results.",
339:           },
340:           settings: {
341:             title: "Genetic Health Report",
342:             subtitle: "Personalized Insights",

```

```

343:         companyName: "Sample Genetics Inc.",
344:         headerColor: "#2563eb",
345:         accentColor: "#dc2626",
346:         fonts: {
347:             primary: "Arial, sans-serif",
348:             secondary: "Georgia, serif",
349:             mono: "Courier New, monospace",
350:         },
351:     },
352:     dynamicDietFieldDefinitions: [],
353:     patientDietAnalysisResults: [],
354:     dietFieldCategories: [],
355:     nutritionData: { quote: "", description: "", data: {} },
356:     sportsAndFitness: {
357:         exerciseType: [],
358:         performance: [],
359:         customImages: {},
360:     },
361:     lifestyleConditions: {},
362:     lifestyleCategoryImages: {},
363:     metabolicCore: {},
364:     digestiveHealth: { quote: "", description: "", data: {} },
365:     genesAndAddiction: { quote: "", description: "", data: {} },
366:     sleepAndRest: { quote: "", description: "", data: {} },
367:     allergiesAndSensitivity: {
368:         quote: "",
369:         description: "",
370:         generalAdvice: "",
371:         data: {},
372:     },
373:     geneTestResults: [],
374:     categories: [],
375:     summaries: {
376:         nutrigenomicsSummary: "Your nutrition genetics summary.",
377:         exerciseGenomicsSummary: "Your exercise genetics summary.",
378:     },
379:     metabolicSummary: { strengths: [], weaknesses: [] },
380:     preventiveHealth: {
381:         diagnosticTests: { halfYearly: [], yearly: [] },
382:         nutritionalSupplements: [],
383:     },
384:     familyGeneticImpact: [],
385: },
386: ],
387: },
388: ];
389: }
390:
391: // Run the migration
392: console.log("? Genetic Reports Database Migration Tool");
393: console.log("=====\n");
394:
395: migrateJsonToDB();

```

## ■ File: tailwind.config.ts

```

=====
1: import type { Config } from "tailwindcss";
2:
3: // all in fixtures is set to tailwind v3 as interims solutions
4:
5: const config: Config = {
6:     darkMode: ["class"],
7:     content: [
8:         "./pages/**/*.{js,ts,jsx,tsx,mdx}",
9:         "./components/**/*.{js,ts,jsx,tsx,mdx}",
10:        "./app/**/*.{js,ts,jsx,tsx,mdx}",
11:        "**.{js,ts,jsx,tsx,mdx}"
12:     ],
13:     theme: {
14:         extend: {
15:             colors: {

```

```

16:     background: 'hsl(var(--background))',
17:     foreground: 'hsl(var(--foreground))',
18:     card: {
19:         DEFAULT: 'hsl(var(--card))',
20:         foreground: 'hsl(var(--card-foreground))'
21:     },
22:     popover: {
23:         DEFAULT: 'hsl(var(--popover))',
24:         foreground: 'hsl(var(--popover-foreground))'
25:     },
26:     primary: {
27:         DEFAULT: 'hsl(var(--primary))',
28:         foreground: 'hsl(var(--primary-foreground))'
29:     },
30:     secondary: {
31:         DEFAULT: 'hsl(var(--secondary))',
32:         foreground: 'hsl(var(--secondary-foreground))'
33:     },
34:     muted: {
35:         DEFAULT: 'hsl(var(--muted))',
36:         foreground: 'hsl(var(--muted-foreground))'
37:     },
38:     accent: {
39:         DEFAULT: 'hsl(var(--accent))',
40:         foreground: 'hsl(var(--accent-foreground))'
41:     },
42:     destructive: {
43:         DEFAULT: 'hsl(var(--destructive))',
44:         foreground: 'hsl(var(--destructive-foreground))'
45:     },
46:     border: 'hsl(var(--border))',
47:     input: 'hsl(var(--input))',
48:     ring: 'hsl(var(--ring))',
49:     chart: {
50:         '1': 'hsl(var(--chart-1))',
51:         '2': 'hsl(var(--chart-2))',
52:         '3': 'hsl(var(--chart-3))',
53:         '4': 'hsl(var(--chart-4))',
54:         '5': 'hsl(var(--chart-5))'
55:     },
56:     sidebar: {
57:         DEFAULT: 'hsl(var(--sidebar-background))',
58:         foreground: 'hsl(var(--sidebar-foreground))',
59:         primary: 'hsl(var(--sidebar-primary))',
60:         'primary-foreground': 'hsl(var(--sidebar-primary-foreground))',
61:         accent: 'hsl(var(--sidebar-accent))',
62:         'accent-foreground': 'hsl(var(--sidebar-accent-foreground))',
63:         border: 'hsl(var(--sidebar-border))',
64:         ring: 'hsl(var(--sidebar-ring))'
65:     },
66:     },
67:     borderRadius: {
68:         lg: 'var(--radius)',
69:         md: 'calc(var(--radius) - 2px)',
70:         sm: 'calc(var(--radius) - 4px)'
71:     },
72:     keyframes: {
73:         'accordion-down': {
74:             from: {
75:                 height: '0'
76:             },
77:             to: {
78:                 height: 'var(--radix-accordion-content-height)'
79:             }
80:         },
81:         'accordion-up': {
82:             from: {
83:                 height: 'var(--radix-accordion-content-height)'
84:             },
85:             to: {
86:                 height: '0'
87:             }
88:         }

```

```

89:   },
90:   animation: {
91:     'accordion-down': 'accordion-down 0.2s ease-out',
92:     'accordion-up': 'accordion-up 0.2s ease-out'
93:   }
94: },
95: },
96:   plugins: [require("tailwindcss-animate")],
97: };
98: export default config;

```

## ■ File: tsconfig.json

```

1: {
2:   "compilerOptions": {
3:     "lib": ["dom", "dom.iterable", "esnext"],
4:     "allowJs": true,
5:     "target": "ES6",
6:     "skipLibCheck": true,
7:     "strict": true,
8:     "noEmit": true,
9:     "esModuleInterop": true,
10:    "module": "esnext",
11:    "moduleResolution": "bundler",
12:    "resolveJsonModule": true,
13:    "isolatedModules": true,
14:    "jsx": "preserve",
15:    "incremental": true,
16:    "plugins": [
17:      {
18:        "name": "next"
19:      }
20:    ],
21:    "paths": {
22:      "@/*": ["./*"]
23:    },
24:    "typeRoots": [".types", ".node_modules/@types"],    // ? Add this
25:    "types": ["node"]                                     // ? optional, but helps
26:  },
27:  "include": ["next-env.d.ts", "**/*.ts", "**/*.tsx", ".next/types/**/*.ts", "types/**/*.d.ts"],
28:  "exclude": ["node_modules"]
29: }

```

## ■ File: types/html2pdf.d.ts

```

1: declare module "html2pdf.js" {
2:   import type { Options as Html2CanvasOptions } from "html2canvas";
3:
4:   export interface Html2PdfOptions {
5:     margin?: number | [number, number, number, number];
6:     filename?: string;
7:     image?: { type?: string; quality?: number };
8:     html2canvas?: Partial<Html2CanvasOptions>;
9:     jsPDF?: {
10:      unit?: "pt" | "mm" | "cm" | "in";
11:      format?: string | [number, number];
12:      orientation?: "portrait" | "landscape";
13:    };
14:     pagebreak?: { mode?: string[] | string };
15:   }
16:
17:   export interface Html2PdfInstance {
18:     set: (options: Html2PdfOptions) => Html2PdfInstance;
19:     from: (element: HTMLElement | string) => Html2PdfInstance;
20:     toPdf: () => Html2PdfInstance;
21:     save: (filename?: string) => Promise<void>;
22:     outputPdf: (type: string, options?: any) => any;
23:   }
24: }

```



```

25:   function html2pdf(): Html2PdfInstance;
26:
27:   export default html2pdf;
28: }

```

## ■ File: types\report-types.ts

```

=====
1: export interface Patient {
2:   id: string;
3:   info: PatientInfo;
4:   reports: Report[];
5: }
6:
7: export interface Report {
8:   id: string;
9:   name?: string;
10:  content: ReportContent;
11:  settings: ReportSettings;
12:  dynamicDietFieldDefinitions: DynamicDietFieldDefinition[];
13:  patientDietAnalysisResults: PatientDietAnalysisResult[];
14:  nutritionData: FullNutritionData;
15:  sportsAndFitness: SportsAndFitness;
16:  lifestyleConditions: LifestyleConditions;
17:  lifestyleCategoryImages?: LifestyleCategoryImages;
18:  metabolicCore: MetabolicCore;
19:  digestiveHealth: DigestiveHealth;
20:  genesAndAddiction: GenesAndAddiction;
21:  sleepAndRest: SleepAndRest;
22:  allergiesAndSensitivity: AllergiesAndSensitivity;
23:  geneTestResults: GeneTestResult[];
24:  categories: GeneticCategory[];
25:  metabolicSummary: MetabolicSummary;
26:  preventiveHealth: PreventiveHealth;
27:  familyGeneticImpactSection: FamilyGeneticImpactSection;
28:  summaries: ReportSummaries;
29:  dietFieldCategories: string[];
30:  genomicAnalysisTable?: GenomicAnalysisTable;
31:  healthSummary?: HealthSummary;
32: }
33:
34: export interface HealthSummary {
35:   description?: string;
36:   data: HealthSummaryEntry[];
37: }
38:
39: export interface HealthSummaryEntry {
40:   title: string;
41:   description: string;
42: }
43:
44: export interface GenomicAnalysisTable {
45:   description: string;
46:   categories: GenomicCategoryGroup[];
47: }
48:
49: export interface GenomicCategoryGroup {
50:   category: string;
51:   subcategories: {
52:     area: string;
53:     trait: string;
54:     genes: string[];
55:   }[];
56: }
57:
58: export interface GeneticCategory {
59:   id: string;
60:   category: string;
61:   imageUrl: string;
62:   description: string;
63:   parameters: string[];
64:   isActive: boolean;

```

```

65:   order: number;
66: }
67:
68: export interface PatientInfo {
69:   name: string;
70:   gender: string;
71:   birthDate: string;
72:   sampleCode: string;
73:   sampleDate: string;
74:   reportDate: string;
75:   checkedBy: string;
76:   scientificContent: string;
77:   disclaimer: string;
78:   signature1: string | null;
79:   signature2: string | null;
80: }
81:
82: export interface ReportContent {
83:   introduction: string;
84:   genomicsExplanation: string;
85:   genesHealthImpact: string;
86:   fundamentalsPRS: string;
87:   utilityDoctors: string;
88:   microarrayExplanation: string;
89:   microarrayData: string;
90: }
91:
92: export interface ReportSettings {
93:   title: string;
94:   subtitle: string;
95:   companyName: string;
96:   headerColor: string;
97:   accentColor: string;
98:   fonts: {
99:     primary: string;
100:    secondary: string;
101:    mono: string;
102:   };
103: }
104:
105: export interface DynamicDietFieldDefinition {
106:   meta: {
107:     quote: string;
108:     description: string;
109:   };
110:   fields: DynamicDiet[];
111: }
112:
113: export interface DynamicDiet {
114:   id: string;
115:   label: string;
116:   category: string;
117:   min: number;
118:   max: number;
119:   highRecommendation: string;
120:   normalRecommendation: string;
121:   lowRecommendation: string;
122: }
123:
124: export interface PatientDietAnalysisResult {
125:   fieldId: string;
126:   score: number;
127:   level: "LOW" | "NORMAL" | "HIGH";
128:   recommendation: string;
129:   recommendations: {
130:     LOW: string;
131:     NORMAL: string;
132:     HIGH: string;
133:   };
134:   selectedLevel: "LOW" | "NORMAL" | "HIGH";
135: }
136:
137: export type DietFieldResult = PatientDietAnalysisResult & {

```

```

138:   label: string;
139:   lowRecommendation?: string;
140:   normalRecommendation?: string;
141:   highRecommendation?: string;
142: };
143:
144: export interface NutrientData {
145:   score: number;
146:   healthImpact: string;
147:   intakeLevel: string;
148:   source: string;
149: }
150:
151: export interface FullNutritionData {
152:   quote?: string;
153:   description?: string;
154:   data: Record<string, Record<string, NutrientData>>;
155: }
156:
157: export interface SportsAndFitness {
158:   quote?: string;
159:   description?: string;
160:   customImages?: Record<string, string>;
161:
162:   [categoryId: string]:
163:     | {
164:         [fieldId: string]: {
165:           level: string;
166:           description: string;
167:         };
168:       }
169:     | string
170:     | Record<string, string>
171:     | undefined;
172: }
173:
174: export interface HealthConditionStatus {
175:   status: "strength" | "improvement";
176:   description?: string;
177:   sensitivity?: "low" | "medium" | "high";
178:   avoid?: string[];
179:   follow?: string[];
180:   consume?: string[];
181:   monitor?: string[];
182:   avoidLabel?: string;
183:   followLabel?: string;
184:   consumeLabel?: string;
185:   monitorLabel?: string;
186: }
187:
188: export interface LifestyleConditions {
189:   quote?: string;
190:   description?: string;
191:   [categoryId: string]:
192:     | {
193:         [conditionName: string]: HealthConditionStatus;
194:       }
195:     | string
196:     | undefined;
197: }
198:
199: export interface LifestyleCategoryImages {
200:   [categoryId: string]: string;
201: }
202:
203: export interface MetabolicGeneEntry {
204:   name: string;
205:   genotype: string;
206:   impact: string;
207: }
208:
209: export interface MetabolicGeneData {
210:   genes: MetabolicGeneEntry[];

```

```
211:   impact?: string;
212:   advice: string;
213: }
214:
215: export type MetabolicCore = {
216:   quote?: string;
217:   description?: string;
218: } & Record<string, MetabolicGeneData>;
219:
220: export interface DigestiveHealthEntry {
221:   title: string;
222:   icon: string;
223:   sensitivity: "Low" | "High" | "";
224:   good: string;
225:   bad: string;
226: }
227:
228: export interface DigestiveHealth {
229:   quote?: string;
230:   description?: string;
231:   data: Record<string, DigestiveHealthEntry>;
232: }
233:
234: export interface AddictionEntry {
235:   title: string;
236:   icon: string;
237:   sensitivityIcon: string;
238: }
239:
240: export interface GenesAndAddiction {
241:   quote?: string;
242:   description?: string;
243:   data: Record<string, AddictionEntry>;
244: }
245:
246: export interface SleepData {
247:   title?: string;
248:   intervention: string;
249:   image?: string;
250: }
251:
252: export interface SleepAndRest {
253:   quote?: string;
254:   description?: string;
255:   data: Record<string, SleepData>;
256: }
257:
258: export interface AllergyEntry {
259:   title: string;
260:   image: string;
261: }
262:
263: export interface AllergiesAndSensitivity {
264:   quote: string;
265:   description: string;
266:   generalAdvice: string;
267:   data: Record<string, AllergyEntry>;
268: }
269:
270: export interface GeneTestResult {
271:   genecode: string;
272:   geneName: string;
273:   variation: string;
274:   result: string;
275: }
276:
277: export interface MetabolicSummary {
278:   strengths: string[];
279:   weaknesses: string[];
280: }
281:
282: export interface PreventiveHealth {
283:   description?: string;
```

```
284:   diagnosticTests: {
285:     halfYearly: string[];
286:     yearly: string[];
287:   };
288:   nutritionalSupplements: Array<{
289:     supplement: string;
290:     needed: boolean;
291:   }>;
292: }
293:
294: export interface FamilyGeneticImpactSection {
295:   description: string;
296:   impacts: FamilyGeneticImpact[];
297: }
298:
299: export interface FamilyGeneticImpact {
300:   gene: string;
301:   normalAlleles: string;
302:   yourResult: string;
303:   healthImpact: string;
304: }
305:
306: export interface ReportSummaries {
307:   nutrigenomicsSummary: string;
308:   exerciseGenomicsSummary: string;
309: }
310:
311: export interface ComprehensiveReportData extends Report {
312:   patientInfo: PatientInfo;
313: }
```

-----