

■ 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\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] 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\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\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\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:
12: import {
13:   Select,
14:   SelectContent,
15:   SelectItem,
16:   SelectTrigger,
17:   SelectValue,
18: } from "@components/ui/select";
19:
20: import type {
21:   Patient,
22:   Report,
23:   PatientInfo,
24:   ReportContent,
25:   ReportSettings,
26:   NutrientData,
27:   GeneTestResult,
28:   ReportSummaries,
29:   SportsAndFitness,
30:   LifestyleConditions,
31:   HealthConditionStatus,
32:   MetabolicCore,
33:   DigestiveHealth,
34:   DigestiveHealthEntry,
35:   GenesAndAddiction,
36:   SleepAndRest,
37:   SleepData,
38:   AllergiesAndSensitivity,
39:   PreventiveHealth,
40:   FamilyGeneticImpact,
41:   DynamicDietFieldDefinition,
42:   PatientDietAnalysisResult,
43:   GenomicAnalysisTable,
44:   GenomicCategoryGroup,
45:   HealthSummary,
46:   HealthSummaryEntry,
47:   ComprehensiveReportData,
48: } from "@types/report-types";
49:
50: // Corrected imports for admin components
51: import PatientInfoAdmin from "@components/admin/patient-info-admin";
52: import ContentAdmin from "@components/admin/content-admin";
53: import SettingsAdmin from "@components/admin/settings-admin";
54: import HealthSummaryAdmin from "@components/admin/health-summary-admin";
55: import DynamicDietFieldAdmin from "@components/admin/dynamic-diet-field-admin";
56: import GeneResultsAdmin from "@components/admin/gene-results-admin";
57: import ImportExportAdmin from "@components/admin/import-export-admin";
58: import NutritionAdmin from "@components/admin/nutrition-admin";
59: import SportsFitnessAdmin from "@components/admin/sports-fitness-admin";
60: import LifestyleConditionsAdmin from "@components/admin/lifestyle-conditions-admin";
61: import DigestiveHealthAdmin from "@components/admin/digestive-health-admin";
62: import GenesAddictionAdmin from "@components/admin/genes-addiction-admin";
63: import SleepRestAdmin from "@components/admin/sleep-rest-admin";
64: import AllergiesSensitivityAdmin from "@components/admin/allergies-sensitivity-admin";
65: import PreventiveHealthAdmin from "@components/admin/preventive-health-admin";
66: import FamilyGeneticImpactAdmin from "@components/admin/family-genetic-impact-admin";
67: import PDFGenerator from "@components/admin/pdf-generator";
68: import MetabolicCoreAdmin from "@components/admin/metabolic-core-admin";
69: import GeneticParametersAdmin from "@components/admin/genetic-parameters-admin";
70: import GenomicAnalysisAdmin from "@components/admin/GenomicAnalysisAdmin";
71:
```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```



```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```



```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```

```

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

```



```

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

```

```

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

```

```

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

```

```

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

```

```

1897:         <TabsTrigger
1898:             value="genes-addiction"
1899:             className="text-xs sm:text-sm whitespace-nowrap"
1900:         >
1901:             Genes & Addiction
1902:         </TabsTrigger>
1903:         <TabsTrigger
1904:             value="sleep-rest"
1905:             className="text-xs sm:text-sm whitespace-nowrap"
1906:         >
1907:             Sleep & Rest
1908:         </TabsTrigger>
1909:         <TabsTrigger
1910:             value="allergies-sensitivity"
1911:             className="text-xs sm:text-sm whitespace-nowrap"
1912:         >
1913:             Allergies
1914:         </TabsTrigger>
1915:         <TabsTrigger
1916:             value="preventive-health"
1917:             className="text-xs sm:text-sm whitespace-nowrap"
1918:         >
1919:             Preventive Health
1920:         </TabsTrigger>
1921:         <TabsTrigger
1922:             value="family-genetic-impact"
1923:             className="text-xs sm:text-sm whitespace-nowrap"
1924:         >
1925:             Family Impact
1926:         </TabsTrigger>
1927:         <TabsTrigger
1928:             value="genomic-analysis"
1929:             className="text-xs sm:text-sm whitespace-nowrap"
1930:         >
1931:             Genomic Analysis
1932:         </TabsTrigger>
1933:         <TabsTrigger
1934:             value="genes"
1935:             className="text-xs sm:text-sm whitespace-nowrap"
1936:         >
1937:             Gene Results
1938:         </TabsTrigger>
1939:
1940:         <TabsTrigger
1941:             value="summaries"
1942:             className="text-xs sm:text-sm whitespace-nowrap"
1943:         >
1944:             Summaries
1945:         </TabsTrigger>
1946:
1947:         <TabsTrigger
1948:             value="pdf-export"
1949:             className="text-xs sm:text-sm whitespace-nowrap"
1950:         >
1951:             PDF Export
1952:         </TabsTrigger>
1953:         <TabsTrigger
1954:             value="import-export"
1955:             className="text-xs sm:text-sm whitespace-nowrap"
1956:         >
1957:             Import/Export
1958:         </TabsTrigger>
1959:
1960:         <TabsTrigger value="preview">Preview</TabsTrigger>
1961:     </TabsList>
1962: </div>
1963:
1964: <div className="w-full overflow-x-auto">
1965:     <TabsContent className="w-full min-w-0" value="patient-info">
1966:         <PatientInfoAdmin
1967:             patientInfo={selectedPatient?.info}
1968:             updatePatientInfo={updatePatientInfo}
1969:             onSave={saveReportData}

```

```

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

```

```

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

```

```

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

```



```

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

```

```

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

```

```

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

```

```

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

```

```

2481:         }}
2482:         setReportData=({updatedReport: ComprehensiveReportData} => {
2483:             setPatients((prev) => {
2484:                 const updated = [...prev];
2485:                 const patient = { ...updated[selectedPatientIndex] };
2486:                 const reports = [...patient.reports];
2487:
2488:                 // Destructure into patientInfo and the rest of the report
2489:                 const { patientInfo, ...reportRest } = updatedReport;
2490:
2491:                 reports[selectedReportIndex] = reportRest; // already typed as Report
2492:
2493:                 updated[selectedPatientIndex] = {
2494:                     ...patient,
2495:                     reports,
2496:                     info: patientInfo,
2497:                 };
2498:
2499:                 return updated;
2500:             });
2501:         }}
2502:     />
2503: </TabsContent>
2504:
2505: <TabsContent className="w-full min-w-0" value="preview">
2506:     {selectedReport && selectedPatient ? (
2507:         <ReportPreview
2508:             selectedReport={selectedReport}
2509:             selectedPatient={selectedPatient}
2510:         />
2511:     ) : (
2512:         <div className="text-center text-gray-500 italic py-10">
2513:             No patient or report selected for preview.
2514:         </div>
2515:     )}
2516: </TabsContent>
2517: </div>
2518: </Tabs>
2519: </div>
2520: );
2521: };
2522:
2523: 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(iconPath);
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 addition 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: }
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\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/${file.name}`;
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/${file.name}`;
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: 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-
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 className="space-x-2">
112:       <Button onClick={onSave}>Save Changes</Button>
113:     </div>
114:   </div>
115:   {/* Top description */}
116:   <Input
117:     value={genomicAnalysisTable.description}
118:     onChange={(e) => updateDescription(e.target.value)}
119:     placeholder="Enter overall description for genomic analysis..."
120:   />
121:
122:   {/* Categories */}
123:   {genomicAnalysisTable.categories.map((cat, catIndex) => (
124:     <div
125:       key={catIndex}
126:       className="border border-gray-200 rounded-lg p-4 space-y-4"
127:     >
128:       {/* Category title */}
129:       <div className="flex items-center gap-2">
130:         <Input
131:           value={cat.category}
132:           onChange={(e) => updateCategoryName(catIndex, e.target.value)}
133:           className="font-bold"
134:         />
135:         <Button
136:           variant="destructive"
137:           size="sm"
138:           onClick={() => removeCategory(catIndex)}
139:         >
140:           <Trash2 className="h-4 w-4" />
141:         </Button>
142:       </div>
143:
144:       {/* Add subcategory */}
145:       <Button
146:         variant="outline"
147:         size="sm"
148:         onClick={() => addSubcategory(catIndex)}
149:       >
150:         <Plus className="h-4 w-4 mr-2" />
151:         Add Row
152:       </Button>
153:
154:       {/* Subcategory table */}
155:       <div className="overflow-x-auto">
156:         <table className="w-full border border-gray-200 text-sm mt-2">
157:           <thead>
158:             <tr className="bg-gray-100">
159:               <th className="border p-2">Area</th>
160:               <th className="border p-2">Trait</th>
161:               <th className="border p-2">Genes (comma separated)</th>
162:
163:               <th className="border p-2">Actions</th>
164:             </tr>
165:           </thead>
166:           <tbody>
167:             {cat.subcategories.map((row, subIndex) => (
168:               <tr key={subIndex}>
169:                 <td className="border p-2">
170:                   <Input
171:                     value={row.area}

```

```

172:             onChange={(e) =>
173:                 updateSubcategory(
174:                     catIndex,
175:                     subIndex,
176:                     "area",
177:                     e.target.value
178:                 )
179:             }
180:         />
181:     </td>
182:     <td className="border p-2">
183:         <Input
184:             value={row.trait}
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 duration-300
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:     </div>
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:         </CardHeader>
249:     </Card>
250:
251:     <div className="text-end my-4">
252:         <Button onClick={handleSave}>
253:             <Save className="w-5 h-5 mr-2" />
254:             Save Changes
255:         </Button>
256:     </div>
257:
258:     { /* Quote & Description */ }
259:     <div className="grid grid-cols-1 lg:grid-cols-2 gap-6 p-4">
260:         <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"
324:                     Uploading...
325:                 </>
326:             ) : (
327:                 <>
328:                     <Upload className="w-4 h-4 mr-2" />
329:                     Upload Image
330:                 </>
331:             )}
332:         </Button>
333:
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:                     onChange={(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"
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-direction column">
589:             {/* Header */}
590:             <div className="flex justify-between items-center px-4 sm:px-6 py-4 border-b bg-gradient-to-r from-gray-100 to-gray-200">
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 border-gray-200 p-2"
611:                             >
612:                                 <div className="flex items-center gap-4 flex-1 overflow-hidden">

```

```

613:                 <Image
614:                     src={url}
615:                     alt={label}
616:                     width={50}
617:                     height={50}
618:                     className="rounded-md object-cover flex-shrink-0"
619:                 />
620:                 <span className="capitalize text-sm text-gray-700 truncate w-full">
621:                     {label.replace(/_/g, " ")}
622:                 </span>
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:         onChange= {(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 imageUrl = 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 iconUrl = 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:                     onValueChange={(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:                 onValueChange={(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:           </div>
694:           <div className="flex-shrink-0">
695:             {iconUrl ? (

```



```

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 justify-between items-start 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 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: { /* Categories Management Section */}
564: <div className="bg-gray-50 rounded-lg p-4 sm:p-6">
565:   <h4 className="font-semibold text-gray-700 mb-4 text-base sm:text-lg">
566:     Diet Field Categories
567:   </h4>
568:
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 border"
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 draggableId="fields">
633:       {(provided) => (
634:         <div
635:           {...provided.draggableProps}

```



```

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-between" >
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:                     onChange={(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: </Droppable>
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:                         </div>
823:                     </div>
824:                 )})
825:             </div>
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:             { /* Modal Header */}
845:             <div className="flex items-center justify-between p-3 sm:p-4 md:p-6 border-b">
846:                 <h3 className="text-base sm:text-lg md:text-xl font-bold">
847:                     Add New Diet Field Definition
848:                 </h3>
849:                 <Button
850:                     variant="ghost"
851:                     size="sm"
852:                     onClick={handleCreateModalClose}
853:                     className="flex-shrink-0"
854:                 >

```

```

855:         <X className="h-4 w-4" />
856:     </Button>
857: </div>
858:
859: { /* Modal Content */}
860: <div className="p-2 sm:p-4 md:p-6">
861:     <div className="space-y-4">
862:         { /* Basic Info */}
863:         <div className="grid grid-cols-1 sm:grid-cols-2 gap-3 sm:gap-4">
864:             <FieldInput
865:                 label="Field Label"
866:                 value={newFieldData.label}
867:                 onChange={ (val) =>
868:                     setNewFieldData({ ...newFieldData, label: String(val) })
869:                 }
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:                     onValueChange={ (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:       value={value ?? ""}
59:       onChange={(e) => onChange(e.target.value)}
60:     />
61:   </div>
62: )
63:
64: interface FieldSelectProps {
65:   label: string
66:   value: string | undefined
67:   options: string[]
68:   onChange: (value: string) => void
69: }
70:
71: export const FieldSelect: React.FC<FieldSelectProps> = ({ label, value, options, onChange }) => (
72:   <div className="flex flex-col">
73:     <label className="text-sm font-medium text-gray-700 mb-1">{label}</label>
74:     <select
75:       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"
76:       value={value ?? ""}
77:       onChange={(e) => onChange(e.target.value)}
78:     >
79:       <option value="">? Select Category ?</option>
80:       {options.map((opt, i) => (
81:         <option key={i} value={opt}>
82:           {opt}
83:         </option>
84:       ))}
85:     </select>
86:   </div>
87: )

```

■ 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 te"
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-fu"
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-indi"
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"
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

```

```

145:         <div className="w-8 h-8 bg-gradient-to-br from-blue-500 to-indigo-500 rounded-full
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
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-bl
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-center">
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 rounded">
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"
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-200"
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-200"
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:     { /* Addiction 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 addiction 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:                         Addiction 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-all"
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 justify-center">
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 addiction 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:             onChange={ (val) =>
497:                 handleFieldChange(key, "sensitivityIcon", val)
498:             }
499:         >
500:             <SelectTrigger className="text-sm border-gray-200 focus:border-red-300 focus:
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: />
333:
334: { /* <Textarea
335:     value={category.description || ""}
336:     onChange={(e) =>
337:         handleCategoryChange(
338:             actualIndex,
339:             "description",
340:             e.target.value
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-200">
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:                     </* Export Section */>
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 gap-4">
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:       <Button
455:         onClick={() => {
456:           const fieldName = newField[categoryId]?.trim();
457:           if (fieldName) {
458:             addField(categoryId, fieldName, "strength");
459:             setNewField((prev) => ({
460:               ...prev,
461:               [categoryId]: "",
462:             }));
463:           }
464:         }}
465:         className="bg-blue-600 hover:bg-blue-700"
466:       >
467:       <Plus className="w-4 h-4 mr-2" />
468:       Add Condition
469:     </Button>
470:   </div>
471: </div>
472:
473: {/* Conditions List */}
474: <div className="space-y-6">
475:   {Object.entries(fieldMap).map(([field, fieldData]) => (
476:     <div
477:       key={field}
478:       className="bg-white border-2 border-gray-100 rounded-lg p-4 sm:p-6"
479:     >
480:       {/* Header */}
481:       <div className="flex flex-col sm:flex-row sm:items-center justify-between gap-3 m-4">
482:         <h4 className="font-semibold text-gray-800 text-base sm:text-lg">
483:           {field}
484:         </h4>
485:         <div className="flex flex-wrap items-center gap-3">
486:           {/* Radio group */}
487:           <div className="flex flex-wrap gap-3">
488:             <label className="flex items-center gap-2 cursor-pointer">
489:               <input
490:                 type="radio"
491:                 name={`_${categoryId}-${field}`}
492:                 checked={fieldData.status === "strength"}
493:                 onChange={() =>
494:                   updateFieldStatus(categoryId, field, {
495:                     status: "strength",
496:                   })
497:               >
498:               Strength
499:             </label>
500:             <label className="flex items-center gap-2 cursor-pointer">
501:               <input
502:                 type="radio"
503:                 name={`_${categoryId}-${field}`}

```

[illegible]

```

583:         }
584:         alt={`${level} sensitivity`}
585:         className="w-14 h-14 sm:w-16 sm:h-16 object-contain"
586:     />
587:     <p className="text-xs sm:text-sm mt-1 capitalize">
588:         {level}
589:     </p>
590: </div>
591:     )
592:   )}
593: </div>
594: </div>
595:
596: { /* Recommendations */}
597: <div className="grid grid-cols-1 md:grid-cols-2 xl:grid-cols-4 gap-4">
598:   {
599:     ["avoid", "follow", "consume", "monitor"] as const
600:   ).map((key) => {
601:     const labelKey =
602:       `${key}Label` as EditableLabelKey;
603:     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 border-slate-200">
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:     className="text-sm"
276:   />
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:         onChange={(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:     </SelectedItem>
597:     <SelectedItem value="ENHANCED INTAKE (VEG SOURCES)">
598:         ENHANCED INTAKE (VEG SOURCES)
599:     </SelectedItem>
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:             <SelectedItem value="DIET">DIET</SelectedItem>
616:             <SelectedItem value="SUPPLEMENTS">SUPPLEMENTS</SelectedItem>
617:             <SelectedItem value="DIET & SUPPLEMENTS">
618:                 DIET & SUPPLEMENTS
619:             </SelectedItem>
620:             <SelectedItem value="DIET & SUN EXPOSURE">
621:                 DIET & SUN EXPOSURE
622:             </SelectedItem>
623:             <SelectedItem value="ENHANCED INTAKE (VEG/FISH OIL) & DIET">
624:                 ENHANCED INTAKE (VEG/FISH OIL) & DIET
625:             </SelectedItem>
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:   /** Single continuous page PDF generation with adaptive height */
42:   const handleSinglePagePDF = 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 - A4 width but adaptive height
69:     const a4Width = 210; // A4 width in mm
70:     const margin = 10; // 10mm margin
71:     const contentWidth = a4Width - 2 * margin;
72:
73:     // High quality scale for better resolution
74:     const scale = 3;
75:     const element = reportRef.current;
76:
77:     // Calculate dimensions
78:     const elementWidth = element.offsetWidth;
79:     const elementHeight = element.scrollHeight;
80:
81:     toast({
82:       title: "Generating PDF",
83:       description: "Capturing entire content...",
84:     });
85:
86:     // Capture the entire content in one go (no slicing)
87:     const canvas = await html2canvas(element, {
88:       scale: scale,
89:       useCORS: true,
90:       allowTaint: false,
91:       backgroundColor: "#FFFFFF",
92:       logging: false,
93:       width: elementWidth,
94:       height: elementHeight, // Capture full height
95:       x: 0,
96:       y: 0, // Start from top
97:       windowWidth: elementWidth,
98:       windowHeight: elementHeight, // Full height
99:       scrollX: 0,
100:      scrollY: 0, // No scroll offset
101:    });
102:
103:    // Calculate adaptive height based on content
104:    const canvasAspectRatio = canvas.width / canvas.height;
105:    const imgWidth = contentWidth;
106:    const imgHeight = contentWidth / canvasAspectRatio;
107:
108:    // Calculate total PDF height including margins
109:    const totalPDFHeight = imgHeight + 2 * margin;
110:
111:    toast({
112:      title: "Generating PDF",
113:      description: "Creating adaptive PDF...",
114:    });
115:
116:    // Create PDF with adaptive height (A4 width x content height)
117:    const pdf = new jsPDF({
118:      orientation: "portrait",
119:      unit: "mm",
120:      format: [a4Width, totalPDFHeight], // Custom format with adaptive height
121:    });
122:
123:    // Add the entire content as single image
124:    const imgData = canvas.toDataURL("image/jpeg", 0.95);
125:    pdf.addImage(imgData, "JPEG", margin, margin, imgWidth, imgHeight);
126:
127:    // Save the PDF
128:    const fileName = `${patientCode} || "medical-report"}-${`

```

```

129:         new Date().toISOString().split("T")[0]
130:     }.pdf`;
131:     pdf.save(fileName);
132:
133:     toast({
134:         title: "Success",
135:         description: `Single page PDF generated successfully (${Math.round(
136:             totalPDFHeight
137:         )}mm height)`,
138:         variant: "default",
139:     });
140: } catch (error) {
141:     console.error("PDF generation error:", error);
142:     toast({
143:         title: "Error",
144:         description: "Failed to generate PDF. Please try again.",
145:         variant: "destructive",
146:     });
147: } finally {
148:     setShowForCapture(false);
149:     setIsGenerating(false);
150: }
151: };
152:
153: return (
154:     <Card className="shadow-lg border-0 bg-white">
155:         <CardHeader className="bg-gradient-to-r from-red-500 to-pink-600 text-white rounded-lg p-4 sm:p-5">
156:             <CardTitle className="text-lg sm:text-xl md:text-2xl font-bold uppercase flex items-center justify-
157:                 PDF Generation
158:             </CardTitle>
159:         </CardHeader>
160:         <CardContent className="space-y-6 p-8">
161:             {/* Enhanced Offscreen Report for PDF Export */}
162:             {showForCapture && (
163:                 <div
164:                     style={{
165:                         position: "fixed",
166:                         top: "0",
167:                         left: "-9999px",
168:                         width: "210mm",
169:                         maxWidth: "210mm",
170:                         background: "white",
171:                         zIndex: 1000,
172:                         padding: "20px",
173:                         fontFamily: "Arial, sans-serif",
174:                         fontSize: "12px",
175:                         lineHeight: "1.4",
176:                         color: "#000000",
177:                         visibility: "visible",
178:                         opacity: 1,
179:                     }}
180:                 >
181:                     <div
182:                         ref={reportRef}
183:                         className="report-container"
184:                         style={{
185:                             minHeight: "auto",
186:                             height: "auto",
187:                             visibility: "visible",
188:                             opacity: 1,
189:                             display: "block",
190:                             width: "100%",
191:                         }}
192:                     >
193:                         <ComprehensiveReportViewer reportData={reportData} />
194:                     </div>
195:                 </div>
196:             )}
197:
198:             {/* Quick Actions */}
199:             <div className="grid md:grid-cols-3 gap-4">
200:                 <Button
201:                     onClick={() =>

```

```

202:         window.open(
203:             `/report/preview?patient=${encodeURIComponent(
204:                 patientCode
205:             )}&report=${reportIndex}` ,
206:             "_blank"
207:         )
208:     }
209:     variant="outline"
210:     className="h-20 flex flex-col items-center justify-center gap-2 bg-blue-200 hover:bg-blue-300 t
211:     disabled={isGenerating}
212: >
213:     <Eye className="h-6 w-6" />
214:     <span>Preview Report</span>
215: </Button>
216:
217: <Button
218:     onClick={handlePrint}
219:     disabled={isGenerating}
220:     className="h-20 flex flex-col items-center justify-center gap-2 bg-blue-600 hover:bg-blue-700 t
221: >
222:     <Printer className="h-6 w-6" />
223:     <span>Print Report</span>
224: </Button>
225:
226: <Button
227:     onClick={handleSinglePagePDF}
228:     disabled={isGenerating}
229:     className="h-20 flex flex-col items-center justify-center gap-2 bg-green-600 hover:bg-green-700
230: >
231:     <Download
232:         className={`h-6 w-6 ${isGenerating ? "animate-spin" : ""}`}
233:     />
234:     <span>{isGenerating ? "Generating..." : "Download PDF"}</span>
235: </Button>
236: </div>
237:
238: { /* Report Information */ }
239: <div className="bg-gray-50 border border-gray-200 rounded-lg p-4">
240:     <h4 className="font-semibold text-red-800 mb-3">
241:         Report Information
242:     </h4>
243:     <div className="grid md:grid-cols-1 gap-4 text-sm">
244:         <div>
245:             <span className="font-medium">Patient Name:</span>{ " " }
246:             {reportData.patientInfo.name || "Not specified"}
247:         </div>
248:         <div>
249:             <span className="font-medium">Sample Code:</span>{ " " }
250:             {reportData.patientInfo.sampleCode || "Not specified"}
251:         </div>
252:         <div>
253:             <span className="font-medium">Report Date:</span>{ " " }
254:             {reportData.patientInfo.reportDate || "Not specified"}
255:         </div>
256:     </div>
257: </div>
258:
259: { /* PDF Generation Tips */ }
260: <div className="bg-blue-50 border border-blue-200 rounded-lg p-4">
261:     <h4 className="font-semibold text-blue-800 mb-2">
262:         PDF Quality Features
263:     </h4>
264:     <ul className="text-sm text-blue-700 space-y-1">
265:         <li>? High-resolution output (3x scaling)</li>
266:         <li>? Single continuous page with adaptive height</li>
267:         <li>? A4 width with proper margins and formatting</li>
268:         <li>? No page breaks - content flows naturally</li>
269:         <li>? Optimized for long reports and comprehensive data</li>
270:     </ul>
271: </div>
272: </CardContent>
273: </Card>
274: );

```

275: }

■ 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-1
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-500"
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:                 >
406:                 <Checkbox
407:                   id={`needed-${index}`}
408:                   checked={supplement.needed}
409:                   onChange={(checked: boolean) =>
410:                     updateNutritionalSupplement(
411:                       index,
412:                       "needed",
413:                       checked
414:                     )
415:                 }
416:                 className="border-purple-300"
417:               />
418:               <Label
419:                 htmlFor={`needed-${index}`}
420:                 className="text-sm font-medium text-gray-700"
421:               >
422:                 Required
423:               </Label>
424:             </div>
425:             <Button
426:               onClick={() => removeNutritionalSupplement(index)}
427:               size="sm"
428:               variant="destructive"
429:               className="px-2 shadow-md hover:shadow-lg transition-all duration-200"
430:             >
431:               <Trash2 className="h-3 w-3" />
432:             </Button>
433:           </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-4">
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:                             <Plus 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:         { /* Action Buttons */ }
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\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>
309:
310:     { /* Entries Grid */}
311:     {Object.entries(entries).length > 0 && (
312:         <div className="grid grid-cols-1 lg:grid-cols-2 xl:grid-cols-3 gap-6">
313:             {Object.entries(entries).map(([key, entry], idx) => (
314:                 <Card key={key} className="border-black">
315:                     <CardHeader className="bg-gradient-to-r from-blue-50 to-cyan-50 ">
316:                         <div className="flex justify-between items-start">
317:                             <div className="flex-1">
318:                                 <CardTitle className="text-lg text-gray-700 line-clamp-2">
319:                                     {entry.title || `Sleep Entry ${idx + 1}`}
320:                                 </CardTitle>
321:                                 <p className="text-sm text-gray-500 mt-1">
322:                                     Entry #{idx + 1}
323:                                 </p>
324:                             </div>
325:                             <Button
326:                                 size="icon"
327:                                 variant="ghost"
328:                                 className="text-red-500 hover:text-red-700 hover:bg-red-50 ml-2"
329:                                 onClick={() => handleDeleteField(key)}
330:                             >
331:                                 <Trash2 className="w-4 h-4" />
332:                             </Button>
333:                         </div>
334:                     </CardHeader>
335:                     <CardContent className="p-6 space-y-6">
336:                         <div className="space-y-2">
337:                             <Label
338:                                 htmlFor={`title-${key}`}
339:                                 className="text-sm font-medium"
340:                             >
341:                                 Title
342:                             </Label>
343:                             <Input
344:                                 id={`title-${key}`}
345:                                 value={entry.title}
346:                                 onChange={(e) =>
347:                                     handleFieldChange(key, "title", e.target.value)
348:                                 }
349:                                 placeholder="e.g. Sleep Cycle Disruption"
350:                                 className="w-full"
351:                             />
352:                         </div>
353:
354:                         <div className="space-y-2">
355:                             <Label

```

```

356:         htmlFor={`image-${key}`}
357:         className="text-sm font-medium"
358:     >
359:         Associated Image
360:     </Label>
361:     <Select
362:         value={entry.image}
363:         onChange={(val) =>
364:             handleFieldChange(key, "image", val)
365:         }
366:     >
367:         <SelectTrigger>
368:             <SelectValue placeholder="Select an image" />
369:         </SelectTrigger>
370:         <SelectContent>
371:             {Object.entries(images).map(([label, url]) => (
372:                 <SelectItem key={label} value={label}>
373:                     <div className="flex items-center gap-3">
374:                         <Image
375:                             src={url}
376:                             alt={label}
377:                             width={24}
378:                             height={24}
379:                             className="rounded"
380:                         />
381:                         <span className="capitalize">
382:                             {label.replace(/_/g, " ")}
383:                         </span>
384:                     </div>
385:                 </SelectItem>
386:             ))}
387:         </SelectContent>
388:     </Select>
389: </div>
390:
391: <div className="space-y-2">
392:     <Label
393:         htmlFor={`intervention-${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:             updatesSportsAndFitness(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-center">
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"

```



```

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:       >
103:         <div className="text-center">
104:           <h2 className="text-lg font-bold text-gray-800 mb-3">
105:             No Report Data Available
106:           </h2>
107:           <p className="text-gray-600 text-sm">
108:             Please provide valid report data to display the comprehensive
109:             report.
110:           </p>
111:         </div>
112:       </div>
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:                 <p
260:                   className="text-base sm:text-lg md:text-xl lg:text-2xl font-light mb-4 sm:mb-6 md:mb-8 lead"

```

```

261:         style={{ fontFamily: settings.fonts.secondary }}
262:     >
263:         {settings.subtitle}
264:     </p>
265: </div>
266: </div>
267:
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:         <h2
272:             className="text-lg sm:text-xl md:text-2xl font-bold text-center mb-4 sm:mb-6 md:mb-8 print:mb-10"
273:             style={{
274:                 fontFamily: settings.fonts.primary,
275:                 color: settings.headerColor,
276:             }}
277:         >
278:             Patient Information
279:         </h2>
280:
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:m
343:     <div className="w-16 sm:w-20 md:w-24 h-0.5 bg-white mx-auto opacity-60 rounded print:w-32"></di
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 pri
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:fo
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 bord
393:         <div className="divide-y divide-gray-200 print:divide-gray-600">
394:             { /* Patient Name Row */ }
395:             <div className="grid grid-cols-4 items-center py-4 px-6 hover:bg-white/50 transition-all dur
396:                 <div className="col-span-1">
397:                     <label className="text-sm font-bold text-gray-700 uppercase tracking-wide print:text-xs p
398:                         Name
399:                     </label>
400:                 </div>
401:                 <div className="col-span-3">
402:                     <p className="text-base font-semibold text-gray-900 break-words print:text-sm print:font-
403:                         {patientInfo.name}
404:                     </p>
405:                 </div>
406:             </div>

```



```

407:
408:     { /* Gender Row */}
409:     <div className="grid grid-cols-4 items-center py-4 px-6 hover:bg-white/50 transition-all dura
410:         <div className="col-span-1">
411:             <label className="text-sm font-bold text-gray-700 uppercase tracking-wide print:text-xs p
412:                 Gender
413:             </label>
414:         </div>
415:         <div className="col-span-3">
416:             <p className="text-base font-semibold text-gray-900 capitalize print:text-sm print:font-b
417:                 {patientInfo.gender}
418:             </p>
419:         </div>
420:     </div>
421:
422:     { /* Birth Date Row */}
423:     <div className="grid grid-cols-4 items-center py-4 px-6 hover:bg-white/50 transition-all dura
424:         <div className="col-span-1">
425:             <label className="text-sm font-bold text-gray-700 uppercase tracking-wide print:text-xs p
426:                 Birth Date
427:             </label>
428:         </div>
429:         <div className="col-span-3">
430:             <p className="text-base font-semibold text-gray-900 print:text-sm print:font-bold">
431:                 {patientInfo.birthDate}
432:             </p>
433:         </div>
434:     </div>
435:
436:     { /* Sample Code Row */}
437:     <div className="grid grid-cols-4 items-center py-4 px-6 hover:bg-white/50 transition-all dura
438:         <div className="col-span-1">
439:             <label className="text-sm font-bold text-gray-700 uppercase tracking-wide print:text-xs p
440:                 Sample Code
441:             </label>
442:         </div>
443:         <div className="col-span-3">
444:             <p className="text-base font-mono font-bold text-orange-600 break-all tracking-wide print
445:                 {patientInfo.sampleCode}
446:             </p>
447:         </div>
448:     </div>
449:
450:     { /* Sample/Data Date Row */}
451:     <div className="grid grid-cols-4 items-center py-4 px-6 hover:bg-white/50 transition-all dura
452:         <div className="col-span-1">
453:             <label className="text-sm font-bold text-gray-700 uppercase tracking-wide print:text-xs p
454:                 Sample/Data Date
455:             </label>
456:         </div>
457:         <div className="col-span-3">
458:             <p className="text-base font-semibold text-gray-900 print:text-sm print:font-bold">
459:                 {patientInfo.sampleDate}
460:             </p>
461:         </div>
462:     </div>
463:
464:     { /* Report Date Row */}
465:     <div className="grid grid-cols-4 items-center py-4 px-6 hover:bg-white/50 transition-all dura
466:         <div className="col-span-1">
467:             <label className="text-sm font-bold text-gray-700 uppercase tracking-wide print:text-xs p
468:                 Report Date
469:             </label>
470:         </div>
471:         <div className="col-span-3">
472:             <p className="text-base font-semibold text-gray-900 print:text-sm print:font-bold">
473:                 {patientInfo.reportDate}
474:             </p>
475:         </div>
476:     </div>
477: </div>
478: </div>
479: </div>

```

```

480:
481:     { /* Authentication Section */ }
482:     <div className="mt-6 mb-6">
483:         <Separator className="mb-4" />
484:         <h3 className="text-center font-bold mb-4 text-gray-600 text-sm">
485:             Report Authentication & Analytics
486:         </h3>
487:         <div className="grid grid-cols-2 gap-6 mb-5">
488:             <div className="text-center space-y-3">
489:                 <p className="text-xs font-semibold text-gray-700">
490:                     Genomic Data Analytics - Checked & Verified By
491:                 </p>
492:                 <div className="flex justify-center py-2">
493:                     {patientInfo.signature1 ? (
494:                         <img
495:                             src={patientInfo.signature1 || "/placeholder.svg"}
496:                             alt="Primary Signature"
497:                             className="max-h-12 w-auto object-contain"
498:                         />
499:                     ) : (
500:                         <p className="italic text-gray-500 text-xs">
501:                             No signature available
502:                         </p>
503:                     )}
504:                 </div>
505:                 <p className="text-xs break-words text-gray-600">
506:                     {patientInfo.checkedBy}
507:                 </p>
508:             </div>
509:             <div className="text-center space-y-3">
510:                 <p className="text-xs font-semibold text-gray-700">
511:                     Scientific Content - Checked & Verified By
512:                 </p>
513:                 <div className="flex justify-center py-2">
514:                     {patientInfo.signature2 ? (
515:                         <img
516:                             src={patientInfo.signature2 || "/placeholder.svg"}
517:                             alt="Secondary Signature"
518:                             className="max-h-12 w-auto object-contain"
519:                         />
520:                     ) : (
521:                         <p className="italic text-gray-500 text-xs">
522:                             No signature available
523:                         </p>
524:                     )}
525:                 </div>
526:                 <p className="text-xs break-words text-gray-600">
527:                     {patientInfo.scientificContent}
528:                 </p>
529:             </div>
530:         </div>
531:         <DataCard title="Disclaimer" className="text-center">
532:             <p className="text-justify whitespace-pre-wrap break-words text-xs leading-relaxed">
533:                 {patientInfo.disclaimer}
534:             </p>
535:         </DataCard>
536:     </div>
537:
538:     <Separator className="my-4" />
539:     { /* Preface Section */ }
540:     <div className="text-center mb-3">
541:         <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
542:             PREFACE
543:         </h2>
544:     </div>
545:
546:     <div className="m-3 w-1/2">
547:         {content.introduction.split(/\n\s*\n/).map((para, idx) => (
548:             <p
549:                 key={idx}
550:                 className="text-xs mb-4"
551:                 style={{ textAlign: "justify", textJustify: "inter-word" }}
552:             >

```

```

553:         {para}
554:     </p>
555:   )})
556: </div>
557:
558: <div className="space-y-3 mb-4">
559:   <div className="text-center mb-3">
560:     <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
561:       WELCOME TO THE WORLD OF GENOMICS
562:     </h2>
563:   </div>
564:
565:   <div className="grid grid-cols-2 gap-3 mb-4">
566:     <DataCard title="What is Genomics?">
567:       <p className="text-justify whitespace-pre-line text-xs">
568:         {content.genomicsExplanation}
569:       </p>
570:     </DataCard>
571:     <div className="flex justify-center items-center">
572:       
577:     </div>
578:   </div>
579:
580:   <div className="grid grid-cols-2 gap-3 mb-4">
581:     <DataCard title="Genes and Health Impact">
582:       <p className="text-justify whitespace-pre-line text-xs">
583:         {content.genesHealthImpact}
584:       </p>
585:     </DataCard>
586:     <div className="flex justify-center items-center">
587:       
592:     </div>
593:   </div>
594:
595:   <Separator className="my-4" />
596:
597:   <div className="text-center mb-3">
598:     <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
599:       WHAT IS NUTRIGENOMICS
600:     </h2>
601:   </div>
602:
603:   <div className="grid grid-cols-2 gap-3 mb-4">
604:     <DataCard title="Fundamentals and PRS">
605:       <p className="text-justify whitespace-pre-line text-xs ">
606:         {content.fundamentalsPRS}
607:       </p>
608:     </DataCard>
609:     <div className="flex justify-center items-center ">
610:       
615:     </div>
616:   </div>
617:
618:   <div className="grid grid-cols-2 gap-3 mb-4">
619:     <DataCard title="Utility for Doctors and Dietitians">
620:       <p className="text-justify whitespace-pre-line text-xs">
621:         {content.utilityDoctors}
622:       </p>
623:     </DataCard>
624:     <div className="flex justify-center items-center ">
625:       
630: </div>
631: </div>
632:
633: <Separator className="my-4" />
634:
635: <div className="text-center mb-3">
636:     <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
637:         ABOUT MICROARRAY
638:     </h2>
639: </div>
640:
641: <div className="grid grid-cols-2 gap-3 mb-4">
642:     <div>
643:         <DataCard title="Microarray Explanation">
644:             <p className="text-justify whitespace-pre-line text-xs">
645:                 {content.microarrayExplanation}
646:             </p>
647:         </DataCard>
648:         <DataCard title="Your Microarray Data" className="mt-4">
649:             <p className="text-justify whitespace-pre-line text-xs">
650:                 {content.microarrayData}
651:             </p>
652:         </DataCard>
653:     </div>
654:
655:     <div className="flex justify-center items-center">
656:         
661:     </div>
662: </div>
663: </div>
664: <Separator className="my-4" />
665: { /* Table of Contents */ }
666: <div className="text-center mb-3">
667:     <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded uppercase">
668:         Area Covered in the Report
669:     </h2>
670: </div>
671: {categories?.map((category) => {
672:     const columns: string[][] = [[], [], []];
673:     category.parameters.forEach((param, index) => {
674:         columns[index % 3].push(param);
675:     });
676:
677:     return (
678:         <div
679:             key={category.id}
680:             className="mb-4 border-b border-gray-300 pb-3"
681:         >
682:             <div className="flex gap-3">
683:                 <div className="w-1/4 flex flex-col items-center ml-3 justify-center gap-2 mb-2 min-h-20 md:
684:                     {category.imageUrl} && (
685:                     <img
686:                         src={category.imageUrl || "/placeholder.svg"}
687:                         alt={category.category}
688:                         className="w-12 h-12 object-cover flex-shrink-0"
689:                     />
690:                     )}
691:                 <h2 className="p-2 sm:px-3 text-xs sm:text-sm md:text-base lg:text-sm font-bold text-gray
692:                     {category.category}
693:                 </h2>
694:             </div>
695:             <div className="w-3/4">
696:                 <div className="grid grid-cols-3 gap-2">
697:                     {columns.map((col, colIndex) => (
698:                         <ul key={colIndex} className="list-none space-y-1">

```

```

699:         {col.map((param, idx) => (
700:             <li
701:                 key={idx}
702:                 className="text-gray-800 text-xs break-words"
703:             >
704:                 {param}
705:             </li>
706:         )}}
707:     </ul>
708:   )}}
709: </div>
710: </div>
711: </div>
712: </div>
713:   );
714: }}
715:
716: <Separator className="my-4" />
717:
718: {/* Health Summary Section */}
719: {(healthSummary.description || healthSummary.data.length > 0) && (
720:   <div className="space-y-4">
721:     {/* Section Title */}
722:     <div className="text-center mb-3">
723:       <h2 className="text-base font-bold mb-2 bg-blue-300 text-white py-2 px-2 rounded uppercase">
724:         DISTRIBUTION OF YOUR GENOMIC ANALYSIS
725:       </h2>
726:     </div>
727:
728:     {/* Top description */}
729:     {healthSummary.description && (
730:       <p className="text-gray-700 text-sm text-justify p-3">
731:         {healthSummary.description}
732:       </p>
733:     )}
734:
735:     {/* Two-column table entries */}
736:     {healthSummary.data.map((entry, index) => (
737:       <div
738:         key={`health-summary-${index}`}
739:         className="grid grid-cols-12 gap-0 w-full"
740:       >
741:         {/* Left Column: Title */}
742:         <div className="col-span-3 flex items-center justify-center p-2 sm:p-4">
743:           <span
744:             className={`text-white font-bold text-[10px] sm:text-sm px-2 sm:px-5 py-4 sm:py-7 rounded`}
745:           >
746:             entry.title.toLowerCase() === "strengths"
747:               ? "bg-green-500"
748:               : entry.title.toLowerCase() === "weaknesses"
749:                 ? "bg-orange-500"
750:                 : entry.title.toLowerCase().includes("diet")
751:                   ? "bg-blue-500"
752:                   : entry.title.toLowerCase().includes("exercise")
753:                     ? "bg-gray-400"
754:                     : entry.title.toLowerCase().includes("preventive")
755:                       ? "bg-yellow-500 text-black"
756:                       : "bg-blue-500"
757:           </span>
758:         </div>
759:         {entry.title}
760:       </span>
761:     </div>
762:
763:     {/* Right Column: Description */}
764:     <div className="col-span-9 border border-gray-400 p-2 sm:p-4 flex flex-col justify-center r
765:       {entry.description &&
766:         (() => {
767:           const lines = entry.description
768:             .split("\n")
769:             .map((l) => l.trim())
770:             .filter(Boolean);
771:           const isBulletList =

```

```

772:             lines.length > 1 &&
773:             (lines.every((line) => /^[^-\?\\d]/.test(line)) ||
774:             lines.length <= 6);
775:
776:         return isBulletList ? (
777:             <ul className="list-disc list-inside text-gray-700 text-xs sm:text-sm normal-case 1
778:                 {lines.map((line, i) => (
779:                     <li key={i}>{line.replace(/^[^-\?\\d. ]+/, "")}</li>
780:                 )}>
781:             </ul>
782:         ) : (
783:             <p className="text-gray-700 text-xs sm:text-sm normal-case leading-relaxed text-just
784:                 {entry.description}
785:             </p>
786:         );
787:     }())}
788: </div>
789: </div>
790: )}}
791: </div>
792: )}
793:
794: <Separator className="my-4" />
795:
796: <div className="w-full max-w-4xl">
797:     <div className="relative w-full" style={{ paddingBottom: "141.4%" }}>
798:         
803:     </div>
804: </div>
805:
806: <Separator className="my-4" />
807:
808: { /* Diet Section */}
809: <div className="text-center mb-3">
810:     <h2 className="text-base font-bold mb-2 bg-green-600 text-white py-2 px-2 rounded">
811:         GENOMICS & DIET
812:     </h2>
813: </div>
814: {Object.keys(groupedDietFields).length === 0 && (
815:     <p className="text-center text-gray-500 italic mb-4">
816:         No diet analysis results available.
817:     </p>
818: )}
819:
820: {reportData?.dynamicDietFieldDefinitions?.[0]?.meta?.quote && (
821:     <div className="mb-3 text-center">
822:         <p className="text-sm font-semibold italic text-gray-800">
823:             "{reportData.dynamicDietFieldDefinitions[0].meta.quote}"
824:         </p>
825:     </div>
826: )}
827:
828: {reportData?.dynamicDietFieldDefinitions?.[0]?.meta?.description && (
829:     <div className="mb-3">
830:         <p className="text-gray-700 p-3 text-xs text-justify">
831:             {reportData.dynamicDietFieldDefinitions[0].meta.description}
832:         </p>
833:     </div>
834: )}
835:
836: <div className="w-full max-w-4xl mx-auto mb-6 print:mb-0 print:max-w-none print:w-full">
837:     <div className="overflow-x-auto overflow-y-hidden">
838:         <table className="w-full min-w-[700px] table-fixed text-xs border-2 border-gray-300 bg-white pr
839:             <tbody>
840:                 {Object.entries(groupedDietFields).flatMap(
841:                     ([category, fields]) =>
842:                     fields.map((data, index) => {
843:                         const score = Number(data.score ?? 0);
844:

```

```

845: //Determine selected level from score
846: let selectedLevel: "LOW" | "NORMAL" | "HIGH" = "NORMAL";
847: if (score <= 3) selectedLevel = "LOW";
848: else if (score >= 7) selectedLevel = "HIGH";
849:
850: const scoreColor =
851:   score <= 3
852:     ? "bg-green-600 print:bg-green-700"
853:     : score <= 6
854:     ? "bg-yellow-500 print:bg-yellow-600"
855:     : "bg-red-600 print:bg-red-600";
856:
857: const renderAdviceCell = (
858:   level: "LOW" | "NORMAL" | "HIGH"
859: ) => (
860:   <td className="p-3 border border-gray-300 align-top w-[17%] print:p-2 print:text-[7]
861:     <div
862:       className={`text-xs font-bold mb-2 uppercase tracking-wide text-center print:te
863:         selectedLevel === level
864:           ? "text-black"
865:           : "text-gray-400 print:text-gray-500"
866:         `}
867:       >
868:         {level}
869:       </div>
870:       <div
871:         className={`text-xs leading-relaxed text-center print:text-[6px] print:leading-
872:           selectedLevel === level
873:             ? "text-gray-900 font-medium"
874:             : "text-gray-400 print:text-gray-500"
875:           `}
876:         style={{
877:           wordBreak: "break-word",
878:           overflowWrap: "break-word",
879:           hyphens: "auto",
880:         }}
881:       >
882:         {(() => {
883:           // Prefer new flat field (e.g. lowRecommendation), fallback to legacy structur
884:           const key = (level.toLowerCase() +
885:             "Recommendation") as
886:             | "lowRecommendation"
887:             | "normalRecommendation"
888:             | "highRecommendation";
889:           return (
890:             (key in data
891:               ? (data as any)[key]
892:               : undefined) ||
893:             data?.recommendations?.[level] ||
894:             "-"
895:           );
896:         })()}
897:       </div>
898:     </td>
899:   );
900:
901: return (
902:   <tr
903:     key={`${category}-${index}`}
904:     className="border-t border-gray-100 print:break-inside-avoid"
905:   >
906:     { /* Category cell - show only once per group */
907:     {index === 0 && (
908:       <td
909:         rowSpan={fields.length}
910:         className="p-2 border border-gray-300 font-bold text-blue-800 bg-gray-50 text
911:       >
912:         <div className="text-center">
913:           {category
914:             .replace(/([a-z])([A-Z])/g, "$1 $2")
915:             .replace(/([A-Z])([A-Z][a-z])/g, "$1 $2")
916:             .replace(/&/g, " & ")
917:             .replace(/\s+/g, " ")

```

```

918:         .trim()})
919:     </div>
920: </td>
921: })
922:
923: { /* Trait name */}
924: <td className="p-3 border border-gray-300 text-center font-semibold text-gray-800"
925:     <div
926:         className="flex items-center justify-center break-words overflow-hidden text-
927:         style={{
928:             wordBreak: "break-word",
929:             overflowWrap: "break-word",
930:             hyphens: "auto",
931:         }}
932:     >
933:         {data.label}
934:     </div>
935: </td>
936:
937: { /* Score badge */}
938: <td className="p-3 border border-gray-300 text-center align-middle w-[10%] print:
939:     <span
940:         className={`inline-block px-5 py-1 rounded-md text-white font-bold text-sm min
941:     >
942:         {score}
943:     </span>
944: </td>
945:
946: { /* Advice columns */}
947: {renderAdviceCell("LOW")}
948: {renderAdviceCell("NORMAL")}
949: {renderAdviceCell("HIGH")}
950: </tr>
951: );
952: })
953: })
954: </tbody>
955: </table>
956: </div>
957: </div>
958:
959: <Separator className="my-4" />
960:
961: <div className="w-full max-w-4xl">
962:     <div className="relative w-full" style={{ paddingBottom: "141.4%" }}>
963:         
968:     </div>
969: </div>
970:
971: <Separator className="my-4" />
972:
973: { /* Nutrition Section */}
974: <div className="text-center mb-3">
975:     <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
976:         NUTRITION
977:     </h2>
978: </div>
979: <SectionTitle title="Nutrition Analysis" icon="" />
980: {reportData.nutritionData?.quote && (
981:     <div className="mb-3">
982:         <p className="text-sm font-semibold italic text-center text-gray-700">
983:             {reportData.nutritionData.quote}
984:         </p>
985:     </div>
986: )}
987: {reportData.nutritionData?.description && (
988:     <div className="mb-4">
989:         <p className="text-gray-600 text-justify leading-relaxed p-3 text-xs">
990:             {reportData.nutritionData.description}

```



```

991:     </p>
992:   </div>
993: })
994:
995: <div className="w-full print:w-[700px] mx-auto mb-6 print:mb-4">
996:   <div className="overflow-x-auto overflow-y-hidden">
997:     <table className="w-full min-w-[700px] text-[13px] border border-gray-300 print:text-[12px] tab
998:       <thead className="bg-gray-200 text-gray-700 uppercase text-[11px]">
999:         <tr>
1000:           <th className="p-2 border border-gray-300 w-[18%] text-center align-middle">
1001:             Category
1002:           </th>
1003:           <th className="p-2 border border-gray-300 w-[21%] text-center align-middle">
1004:             Nutrient
1005:           </th>
1006:           <th className="p-2 border border-gray-300 w-[10%] text-center align-middle">
1007:             Score
1008:           </th>
1009:           <th className="p-2 border border-gray-300 w-[18%] text-center align-middle">
1010:             Health Impact
1011:           </th>
1012:           <th className="p-2 border border-gray-300 w-[18%] text-center align-middle">
1013:             Intake Level
1014:           </th>
1015:           <th className="p-2 border border-gray-300 w-[15%] text-center align-middle">
1016:             Source
1017:           </th>
1018:         </tr>
1019:       </thead>
1020:       <tbody>
1021:         {Object.entries(reportData.nutritionData?.data || {}).flatMap(
1022:           ([sectionName, sectionData]) => {
1023:             const entries = Object.entries(sectionData);
1024:
1025:             const formattedCategory = sectionName
1026:               .replace(/[A-Z]/g, " $1")
1027:               .replace(/^./, (str) => str.toUpperCase());
1028:
1029:             // Helper to format nutrient names
1030:             const formatNutrientName = (name: string) => {
1031:               let label = name
1032:                 .replace(/[A-Z]/g, " $1") // space before capitals
1033:                 .replace(/^./, (str) => str.toUpperCase()); // uppercase first letter
1034:
1035:               // Special handling for Vitamin patterns
1036:               label = label.replace(
1037:                 /^Vitamin\s?([a-z])(\d+)?$/i,
1038:                 (_, letter, number) =>
1039:                   `Vitamin ${letter.toUpperCase()}${number || ""}`
1040:               );
1041:
1042:               return label;
1043:             };
1044:
1045:             return entries.map(([key, data], index) => {
1046:               const formattedLabel = formatNutrientName(key);
1047:
1048:               const score = Number(data.score ?? 0);
1049:               const scoreColor =
1050:                 score <= 3
1051:                   ? "bg-green-600"
1052:                   : score <= 6
1053:                   ? "bg-yellow-500"
1054:                   : "bg-orange-500";
1055:
1056:               return (
1057:                 <tr
1058:                   key={` ${sectionName}-${key}`}
1059:                   className="border-t border-gray-200 break-inside-avoid"
1060:                 >
1061:                   {index === 0 && (
1062:                     <td
1063:                       rowspan={entries.length}

```

```

1064:         className="p-2 border border-gray-300 font-bold text-blue-800 uppercase bg-g
1065:     >
1066:         {formattedCategory}
1067:     </td>
1068: })
1069: <td className="p-2 border border-gray-300 font-medium text-gray-800 text-center
1070:     {formattedLabel}
1071: </td>
1072: <td className="p-2 border border-gray-300 text-center align-middle">
1073:     <span
1074:         className={`inline-block px-5 py-1 rounded text-white font-bold ${scoreColor}
1075:     >
1076:         {score}
1077:     </span>
1078: </td>
1079: <td className="p-2 border border-gray-300 text-center text-gray-700 break-words
1080:     {data.healthImpact || "-"}
1081: </td>
1082: <td className="p-2 border border-gray-300 text-center text-gray-700 break-words
1083:     {data.intakeLevel || "-"}
1084: </td>
1085: <td className="p-2 border border-gray-300 text-center text-gray-700 break-words
1086:     {data.source || "-"}
1087: </td>
1088: </tr>
1089:     );
1090:   });
1091: }
1092: })
1093: </tbody>
1094: </table>
1095: </div>
1096: </div>
1097:
1098: <Separator className="my-4" />
1099:
1100: <div className="w-full max-w-4xl">
1101:   <div className="relative w-full" style={{ paddingBottom: "141.4%" }}>
1102:     
1107:   </div>
1108: </div>
1109:
1110: <Separator className="my-4" />
1111:
1112: <div className="text-center mb-3">
1113:   <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
1114:     SPORTS AND FITNESS
1115:   </h2>
1116: </div>
1117: <div className="space-y-4 mb-4">
1118:   { /* Quote */ }
1119:   {sportsAndFitness.quote && (
1120:     <p className="text-sm font-semibold italic text-center text-red-600 mb-3">
1121:       {sportsAndFitness.quote}
1122:     </p>
1123:   )}
1124:
1125:   { /* Description */ }
1126:   {sportsAndFitness.description && (
1127:     <p className="text-gray-600 text-justify leading-relaxed mb-4 p-3 text-xs">
1128:       {sportsAndFitness.description}
1129:     </p>
1130:   )}
1131:
1132:   { /* Table */ }
1133:   <div className="w-full overflow-x-auto mb-4">
1134:     <table className="w-full min-w-[500px] text-xs border-collapse border border-gray-300">
1135:       <thead>
1136:         <tr className="bg-gray-100">

```

```

1137:         <th className="w-1/5 border border-gray-300 p-2 text-xs font-bold text-center uppercase">
1138:             CATEGORY
1139:         </th>
1140:         <th className="w-1/4 border border-gray-300 p-2 text-xs font-bold text-center uppercase">
1141:             TRAIT
1142:         </th>
1143:         <th className="w-1/6 border border-gray-300 p-2 text-xs font-bold text-center uppercase">
1144:             IMAGE
1145:         </th>
1146:         <th className="w-1/2 border border-gray-300 p-2 text-xs font-bold text-center uppercase">
1147:             DESCRIPTION
1148:         </th>
1149:     </tr>
1150: </thead>
1151: <tbody>
1152:     {Object.entries(sportsAndFitness)
1153:       .filter(
1154:         ([key]) =>
1155:           key !== "quote" &&
1156:           key !== "description" &&
1157:           key !== "customImages"
1158:       )
1159:       .map(([categoryKey, fields]) => {
1160:         if (
1161:           !fields ||
1162:           typeof fields !== "object" ||
1163:           Array.isArray(fields)
1164:         ) {
1165:           return null;
1166:         }
1167:
1168:         const fieldEntries = Object.entries(
1169:           fields as Record<
1170:             string,
1171:             { level: string; description: string; label?: string }
1172:           >
1173:         );
1174:
1175:         return fieldEntries.map(([fieldKey, fieldData], index) => {
1176:           const selectedImageKey = fieldData.level
1177:             ?.split("-")
1178:             .pop()
1179:             ?.toLowerCase();
1180:
1181:           const imageUrl =
1182:             (
1183:               sportsAndFitness.customImages as
1184:                 | Record<string, string>
1185:                 | undefined
1186:             )?.[selectedImageKey ?? ""] ||
1187:             defaultImageMap[selectedImageKey ?? ""] ||
1188:             "/sports/default.png";
1189:
1190:           return (
1191:             <tr
1192:               key={`_${categoryKey}-${fieldKey}`}
1193:               className="border-b border-gray-300 align-middle"
1194:             >
1195:               {index === 0 && (
1196:                 <td
1197:                   rowspan={fieldEntries.length}
1198:                   className="text-center font-semibold text-red-700 w-1/5 uppercase align-middle"
1199:                 >
1200:                   {categoryKey
1201:                     .replace(/_/g, " ") // <-- replace underscores with spaces
1202:                     .replace(/([A-Z])/g, " $1")}
1203:                 </td>
1204:               )}
1205:               <td className="text-center font-medium uppercase text-gray-800 w-1/4 border border-gray-300"
1206:                 {(fieldData.label || fieldKey).replace(/_/g, " ")}{ " "}
1207:                 { /* <-- replace underscores */ }
1208:               </td>
1209:               <td className="text-center w-1/5 border border-gray-300 p-1 align-middle">

```

```

1210:                 <img
1211:                     src={imageUrl || "/placeholder.svg"}
1212:                     alt={selectedImageKey}
1213:                     className="w-16 h-17 p-1 mx-auto rounded-lg"
1214:                 />
1215:             </td>
1216:             <td className="text-center text-gray-600 text-xs w-1/2 border border-gray-300 p-1">
1217:                 {fieldData.description}
1218:             </td>
1219:         </tr>
1220:     );
1221:   });
1222:   });
1223: </tbody>
1224: </table>
1225: </div>
1226: </div>
1227:
1228: <Separator className="my-4" />
1229:
1230: <div className="w-full max-w-4xl">
1231:   <div className="relative w-full" style={{ paddingBottom: "141.4%" }}>
1232:     
1237:   </div>
1238: </div>
1239:
1240: <Separator className="my-4" />
1241:
1242: { /* Lifestyle Conditions */ }
1243: <div className="text-center mb-3">
1244:   <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
1245:     LIFESTYLE CONDITIONS
1246:   </h2>
1247: </div>
1248:
1249: {reportData.lifestyleConditions && (
1250:   <div className="mb-6">
1251:     {reportData.lifestyleConditions.quote && (
1252:       <p className="text-sm font-semibold italic text-center text-red-600 mb-3">
1253:         "{reportData.lifestyleConditions.quote}"
1254:       </p>
1255:     )}
1256:
1257:     {reportData.lifestyleConditions.description && (
1258:       <p className="text-gray-600 text-justify leading-relaxed mb-4 p-3 text-xs">
1259:         {reportData.lifestyleConditions.description}
1260:       </p>
1261:     )}
1262:
1263:     <div className="w-full overflow-x-auto mb-6">
1264:       <table className="min-w-full border border-gray-300 bg-white rounded-lg overflow-hidden text-gray-700">
1265:         <thead className="bg-gray-100 text-left text-xs font-semibold text-gray-700">
1266:           <tr>
1267:             <th className="px-2 py-2 w-64 text-center align-middle">
1268:               Category
1269:             </th>
1270:             <th className="px-2 py-2 text-center align-middle">
1271:               Condition
1272:             </th>
1273:             <th className="px-2 py-2 text-center align-middle">
1274:               Your Strengths
1275:             </th>
1276:             <th className="px-2 py-2 text-center align-middle">
1277:               Improvement Needed
1278:             </th>
1279:           </tr>
1280:         </thead>
1281:         <tbody>
1282:           {Object.entries(reportData.lifestyleConditions)

```

```

1283:         .reverse() // ? reverse order
1284:     .map(([categoryId, categoryData]) => {
1285:         if (
1286:             categoryId === "quote" ||
1287:             categoryId === "description" ||
1288:             typeof categoryData !== "object"
1289:         )
1290:             return null;
1291:
1292:         const fields = Object.entries(
1293:             categoryData as Record<string, HealthConditionStatus>
1294:         ).reverse(); // ? reverse conditions inside category
1295:
1296:         return fields.map(([condition, { status }], index) => (
1297:             <tr
1298:                 key={`_${categoryId}-${condition}`}
1299:                 className="border-t text-center align-middle "
1300:             >
1301:                 {index === 0 && (
1302:                     <td
1303:                         rowspan={fields.length}
1304:                         className="px-2 py-5 font-medium text-center align-middle text-gray-900 "
1305:                     >
1306:                         <div className="flex items-center gap-2">
1307:                             {reportData.lifestyleCategoryImages?.[
1308:                                 categoryId
1309:                             ] ? (
1310:                                 <img
1311:                                     src={
1312:                                         reportData.lifestyleCategoryImages[
1313:                                             categoryId
1314:                                         ] || "/placeholder.svg"
1315:                                     }
1316:                                     alt={categoryId}
1317:                                     className="w-12 h-12 rounded object-cover"
1318:                                 />
1319:                             ) : (
1320:                                 <div className="w-4 h-4 bg-gray-300 rounded " />
1321:                             )}
1322:                             <span className="capitalize">
1323:                                 {categoryId === categoryId.toUpperCase()
1324:                                     ? categoryId
1325:                                     : categoryId.replace(/([A-Z])/g, " $1")}
1326:                             </span>
1327:                         </div>
1328:                     </td>
1329:                 )}
1330:                 <td className="px-3 py-3 ">{condition}</td>
1331:                 <td className="px-2 py-1 text-center">
1332:                     {status === "strength" && (
1333:                         <Check className="text-green-600 inline w-3 h-3" />
1334:                     )}
1335:                 </td>
1336:                 <td className="px-2 py-1 text-center">
1337:                     {status === "improvement" && (
1338:                         <Check className="text-red-600 inline w-3 h-3" />
1339:                     )}
1340:                 </td>
1341:             </tr>
1342:         ));
1343:     }));
1344: </tbody>
1345: </table>
1346: </div>
1347:
1348: <div className="text-center mb-7">
1349:     <h2 className=" mb- bg-gray-300 text-white py-1 px-1 rounded"></h2>
1350: </div>
1351:
1352: {Object.entries(reportData.lifestyleConditions)
1353:     .filter(([key]) => ![ "quote", "description" ].includes(key))
1354:     .map(([categoryId, conditions]) => (
1355:         <div key={categoryId} className="mb-6">

```

```

1356: <div className="text-center">
1357:   <div className="text-base font-bold mb-3 bg-red-400 text-white py-0.5 px-2 rounded inl
1358:     {categoryId
1359:       .replace(/[a-z]([A-Z])/g, "$1 $2")
1360:       .replace(/[A-Z]([A-Z][a-z])/g, "$1 $2")
1361:       .replace(/&/g, " & ")
1362:       .replace(/\\s+/g, " ")
1363:       .trim()}
1364:   </div>
1365: </div>
1366:
1367: <div className="grid grid-cols-1 gap-3">
1368:   {Object.entries(
1369:     conditions as Record<string, HealthConditionStatus>
1370:   ).map(([fieldKey, data]) => (
1371:     <div
1372:       key={fieldKey}
1373:       className="border rounded-lg border-gray-400 shadow-md bg-white p-3 space-y-2"
1374:     >
1375:       <div className="flex justify-between gap-2 items-stretch">
1376:         <div className="w-[76%]">
1377:           <h3 className="text-sm font-bold text-red-400 mb-1 uppercase">
1378:             {fieldKey
1379:               .replace(/[a-z]([A-Z])/g, "$1 $2")
1380:               .replace(/[A-Z]([A-Z][a-z])/g, "$1 $2")
1381:               .replace(/&/g, " & ")
1382:               .replace(/\\s+/g, " ")
1383:               .trim()}
1384:           </h3>
1385:
1386:           {data.description && (
1387:             <p className="text-xs text-gray-700 leading-relaxed text-justify">
1388:               {data.description}
1389:             </p>
1390:           )}
1391:         </div>
1392:         <div className="w-[24%] flex items-center p-3 justify-center">
1393:           {data.sensitivity && (
1394:             <img
1395:               src={`./sensitivity/${data.sensitivity}.png`}
1396:               alt={data.sensitivity}
1397:               className="w-auto h-auto object-contain max-w-full max-h-29 print:max-h-25
1398:             />
1399:           )}
1400:         </div>
1401:       </div>
1402:       <hr className="border-t border-gray-200" />
1403:       <div className="grid grid-cols-4 gap-2">
1404:         {(
1405:           ["avoid", "follow", "consume", "monitor"] as const
1406:         ).map((key) =>
1407:           data[key]?.length && data[key]?.length > 0 ? (
1408:             <div key={key}>
1409:               <p className="text-xs font-bold uppercase mb-1 ml-2">
1410:                 {data[`${key}Label`] || key.toUpperCase()}
1411:               </p>
1412:               <ul className="list-disc list-outside pl-5 text-xs text-gray-700 space-y-1
1413:                 {data[key]?.flatMap((item, idx) =>
1414:                   typeof item === "string"
1415:                     ? item
1416:                       .split("\\n")
1417:                       .filter((line) => line.trim() !== "")
1418:                       .map((line, subIdx) => (
1419:                         <li key={`-${idx}-${subIdx}`}>
1420:                           {line.trim()}
1421:                         </li>
1422:                       ))
1423:                     : null
1424:                 )}
1425:               </ul>
1426:             </div>
1427:           ) : null
1428:         )}

```



```

1502:         .replace(/[A-Z])([A-Z][a-z])/g, "$1 $2")
1503:         .replace(/&/g, " & ")
1504:         .replace(/\\s+/g, " ")
1505:         .trim()}
1506:     </td>
1507: </>
1508: ) : null}
1509: <td className="px-2 py-1">{gene.name}</td>
1510: <td className="px-2 py-1 font-mono">
1511:     {gene.genotype}
1512: </td>
1513: <td className="px-2 py-1">
1514:     <div className="flex justify-center items-center gap-2">
1515:         <span
1516:             className={`w-7 h-7 rounded-full inline-block ${
1517:                 gene.impact?.toLowerCase().includes("high")
1518:                 ? "bg-red-500"
1519:                 : gene.impact?.toLowerCase().includes("low")
1520:                 ? "bg-slate-400"
1521:                 : gene.impact
1522:                     ?.toLowerCase()
1523:                     .includes("moderate") ||
1524:                     gene.impact
1525:                     ?.toLowerCase()
1526:                     .includes("average")
1527:                 ? "bg-yellow-400"
1528:                 : "bg-green-500"
1529:             }`}>
1530:         </span>
1531:     </div>
1532: </td>
1533: {index === 0 ? (
1534:     <td
1535:         rowSpan={totalRows}
1536:         className="px-2 py-1 text-left align-middle border-1 border-slate-200"
1537:     >
1538:         <div className="text-center whitespace-pre-wrap text-xs">
1539:             {areaData.advice || "No advice provided"}
1540:         </div>
1541:     </td>
1542: ) : null}
1543: </tr>
1544: ))
1545: ) : (
1546:     // Show row even if no genes, to display area advice
1547:     <tr key={areaKey} className="border-t border-slate-200">
1548:         <td className="font-bold px-2 py-2.5 text-blue-700 uppercase text-center border-r 1
1549:             {areaKey
1550:                 .replace(/[a-z])([A-Z])/g, "$1 $2")
1551:                 .replace(/[A-Z])([A-Z][a-z])/g, "$1 $2")
1552:                 .replace(/&/g, " & ")
1553:                 .replace(/\\s+/g, " ")
1554:                 .trim()}
1555:             </td>
1556:             <td className="px-2 py-1 text-gray-400">No genes</td>
1557:             <td className="px-2 py-1 text-gray-400"></td>
1558:             <td className="px-2 py-1 text-gray-400"></td>
1559:             <td className="px-2 py-1 text-left">
1560:                 <div className="whitespace-pre-wrap text-xs">
1561:                     {areaData.advice || "No advice provided"}
1562:                 </div>
1563:             </td>
1564:         </tr>
1565:     );
1566: }}
1567: </tbody>
1568: </table>
1569: </div>
1570: </div>
1571:
1572: <Separator className="my-4" />
1573:
1574: <div className="w-full max-w-4xl">

```



```

1575:     <div className="relative w-full" style={{ paddingBottom: "141.4%" }}>
1576:       
1581:     </div>
1582:   </div>
1583:
1584:   <Separator className="my-4" />
1585:
1586:   { /* Digestive Health */ }
1587:   <div className="text-center mb-3">
1588:     <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
1589:       DIGESTIVE HEALTH
1590:     </h2>
1591:   </div>
1592:
1593:   {digestiveHealth?.quote && (
1594:     <p className="text-sm font-semibold italic text-center text-red-600 mb-3">
1595:       "{digestiveHealth.quote}"
1596:     </p>
1597:   )}
1598:   {digestiveHealth?.description && (
1599:     <p className="text-gray-600 text-justify leading-relaxed mb-4 p-3 text-xs">
1600:       {digestiveHealth.description}
1601:     </p>
1602:   )}
1603:   <div className="overflow-x-auto rounded-sm shadow">
1604:     {digestiveHealth?.data &&
1605:       Object.keys(digestiveHealth.data).length > 0 ? (
1606:         <table className="min-w-full text-xs text-center border border-gray-300 table-fixed">
1607:           <thead className="bg-red-500 text-white">
1608:             <tr>
1609:               <th className="w-[20%] p-2 border border-gray-300">Title</th>
1610:               <th className="w-[20%] p-2 border border-gray-300">Icon</th>
1611:               <th className="w-[30%] p-2 text-center border border-gray-300">
1612:                 LOW
1613:               </th>
1614:               <th className="w-[30%] p-2 text-center border border-gray-300">
1615:                 HIGH
1616:               </th>
1617:             </tr>
1618:           </thead>
1619:           <tbody>
1620:             {Object.entries(digestiveHealth.data).map(([key, item]) => {
1621:               const iconPath = item.icon
1622:                 ? `/digestive/${item.icon}.png`
1623:                 : null;
1624:               const isLow = item.sensitivity === "Low";
1625:               const isHigh = item.sensitivity === "High";
1626:
1627:               return (
1628:                 <tr key={key} className="border-t border-gray-200">
1629:                   <td className="p-2 border border-gray-300 font-medium text-center uppercase">
1630:                     {item.title || <span className="italic">Untitled</span>}
1631:                   </td>
1632:                   <td className="p-2 border border-gray-300">
1633:                     {iconPath ? (
1634:                       <img
1635:                         src={iconPath}
1636:                         alt={item.title || "icon"}
1637:                         className="w-15 h-auto mx-auto object-contain"
1638:                         onError={(e) => {
1639:                           e.currentTarget.onerror = null;
1640:                           e.currentTarget.src =
1641:                             "/digestive/placeholder.png";
1642:                         }}
1643:                       />
1644:                     ) : (
1645:                       <span className="text-slate-400 italic">No Icon</span>
1646:                     )}
1647:                   </td>

```

```

1648:         <td
1649:             className={`p-2 border border-gray-300 text-center whitespace-pre-line ${
1650:                 isLow ? "text-gray-900" : "text-gray-400"
1651:             }`}
1652:         >
1653:             {item.good || <span className="italic">?</span>}
1654:         </td>
1655:         <td
1656:             className={`p-2 border border-gray-300 text-center whitespace-pre-line ${
1657:                 isHigh ? "text-gray-900" : "text-gray-400"
1658:             }`}
1659:         >
1660:             {item.bad || <span className="italic">?</span>}
1661:         </td>
1662:     </tr>
1663:     );
1664:   }}}
1665: </tbody>
1666: </table>
1667: ) : (
1668:   <p className="text-center text-sm text-gray-500 p-4 italic">
1669:     No digestive health data available.
1670:   </p>
1671: )}
1672: </div>
1673:
1674: <Separator className="my-4" />
1675:
1676: <div className="w-full max-w-4xl">
1677:   <div className="relative w-full" style={{ paddingBottom: "141.4%" }}>
1678:     
1683:   </div>
1684: </div>
1685:
1686: <Separator className="my-" />
1687:
1688: {/* Genes & Addiction */}
1689: <div className="text-center mb-3">
1690:   <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded uppercase">
1691:     Genes & Addiction
1692:   </h2>
1693: </div>
1694: <div className="mb-4">
1695:   {genesAndAddiction.quote && (
1696:     <p className="text-sm font-semibold italic text-center text-red-600 mb-3">
1697:       "{genesAndAddiction.quote}"
1698:     </p>
1699:   )}
1700:
1701:   {genesAndAddiction.description && (
1702:     <p className="text-gray-600 text-justify leading-relaxed mb-4 p-3 text-xs">
1703:       {genesAndAddiction.description}
1704:     </p>
1705:   )}
1706:
1707:   <div className="w-full">
1708:     <table className="w-full table-fixed border border-gray-300 bg-white border-collapse text-xs">
1709:       <thead className="bg-blue-300">
1710:         <tr>
1711:           <th className="p-2 text-xs font-bold text-white text-left border border-gray-300 uppercase">
1712:             Addictive Habit
1713:           </th>
1714:           <th className="p-2 text-xs font-bold text-white text-center border border-gray-300 uppercase">
1715:             Chance of Addiction
1716:           </th>
1717:         </tr>
1718:       </thead>
1719:       <tbody>
1720:         {Object.entries(genesAndAddiction.data || {}).map(

```

```

1721:         ([key, entry], index) => (
1722:             <tr
1723:                 key={key}
1724:                 className={`transition-all duration-200 text-center ${
1725:                     index % 2 === 0 ? "bg-white" : "bg-gray-50"
1726:                 }`}
1727:             >
1728:                 <td className="p-0 border border-gray-300">
1729:                     <table className="w-full h-full table-fixed border-collapse">
1730:                         <tbody>
1731:                             <tr>
1732:                                 <td className="w-1/3 p-2 align-middle bg-blue-100">
1733:                                     <span className="text-xs font-semibold text-gray-800 uppercase">
1734:                                         {entry.title}
1735:                                     </span>
1736:                                 </td>
1737:                                 <td className="w-[2px] border-gray-300"></td>
1738:                                 <td className="w-2/3 p-2 text-center">
1739:                                     {entry.icon && (
1740:                                         <img
1741:                                             src={`addiction/icons/${entry.icon}.png`}
1742:                                             alt={entry.title}
1743:                                             className="w-20 h-24 object-contain mx-auto"
1744:                                         />
1745:                                     )}
1746:                                 </td>
1747:                             </tr>
1748:                         </tbody>
1749:                     </table>
1750:                 </td>
1751:                 <td
1752:                     className="text-center border border-gray-300 align-middle"
1753:                     style={{ width: "200px" }} // lock column width
1754:                 >
1755:                     {entry.sensitivityIcon && (
1756:                         <div className="flex justify-center items-center h-full">
1757:                             <img
1758:                                 src={`addiction/sensitivity/${entry.sensitivityIcon}.png`}
1759:                                 alt="Chance of Addiction"
1760:                                 className="max-w-[170px] max-h-[100px] object-contain"
1761:                             />
1762:                         </div>
1763:                     )}
1764:                 </td>
1765:             </tr>
1766:         )
1767:     )}
1768: </tbody>
1769: </table>
1770: </div>
1771: </div>
1772:
1773: <Separator className="mt-7" />
1774:
1775: <div className="w-full max-w-4xl">
1776:     <div className="relative w-full" style={{ paddingBottom: "141.4%" }}>
1777:         
1782:     </div>
1783: </div>
1784:
1785: <Separator className="my-4" />
1786: { /* Sleep & Rest */ }
1787:
1788: <div className="text-center mb-3">
1789:     <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded uppercase">
1790:         Sleep & Rest
1791:     </h2>
1792: </div>
1793: <SectionTitle title="Sleep & Rest" icon="" />

```

```

1794:
1795: {sleepAndRest?.quote && (
1796:   <p className="text-sm font-semibold italic text-center text- text-red-600 mb-3">
1797:     "{sleepAndRest.quote}"
1798:   </p>
1799: )}
1800:
1801: {sleepAndRest?.description && (
1802:   <p className="text-gray-600 text-justify leading-relaxed mb-4 p-3 text-xs">
1803:     {sleepAndRest.description}
1804:   </p>
1805: )}
1806:
1807: <div className="overflow-x-auto rounded-sm shadow mb-4">
1808:   <table className="min-w-full text-xs text-center border border-gray-300 table-fixed">
1809:     <thead className="bg-blue-600 text-white">
1810:       <tr>
1811:         <th className="w-[25%] p-1 border border-gray-300">
1812:           Sleep Parameter
1813:         </th>
1814:         <th className="w-[25%] p-1 border border-gray-300">
1815:           Genetic Icon
1816:         </th>
1817:         <th className="w-[50%] p-1 border border-gray-300">
1818:           Intervention Advice
1819:         </th>
1820:       </tr>
1821:     </thead>
1822:     <tbody>
1823:       {sleepAndRest?.data &&
1824:         Object.entries(sleepAndRest.data).map(([key, entry], index) => (
1825:           <tr>
1826:             key={key}
1827:             className={index % 2 === 0 ? "bg-white" : "bg-gray-50"}
1828:           >
1829:             {/* Sleep Parameter */}
1830:             <td className="p-1 border border-gray-300 bg-blue-200 font-medium text-center capitali
1831:               {entry.title?.trim()
1832:                 ? entry.title
1833:                 : key?.replace(/_/g, " ") || "No parameter"}
1834:             </td>
1835:
1836:             {/* Genetic Icon */}
1837:             <td className="p-1 border border-gray-300">
1838:               {entry.image ? (
1839:                 <img
1840:                   src={`~/sleep/${entry.image}.png`}
1841:                   alt={entry.title || key}
1842:                   className="w-24 h-26 object-contain mx-auto"
1843:                   onError={(e) => {
1844:                     e.currentTarget.onerror = null;
1845:                     e.currentTarget.src = "/sleep/placeholder.png";
1846:                   }}
1847:                 />
1848:               ) : (
1849:                 <span className="text-slate-400 italic">No Image</span>
1850:               )}
1851:             </td>
1852:
1853:             {/* Intervention Advice */}
1854:             <td className="p-1 border border-gray-300 text-left whitespace-pre-line text-gray-700">
1855:               {entry.intervention?.trim() ? (
1856:                 entry.intervention
1857:               ) : (
1858:                 <span className="italic text-gray-400">
1859:                   No advice available
1860:                 </span>
1861:               )}
1862:             </td>
1863:           </tr>
1864:         )}
1865:       </tbody>
1866:     </table>

```

```

1867: </div>
1868:
1869: <Separator className="my-4" />
1870:
1871: <div className="w-full max-w-4xl">
1872:   <div className="relative w-full" style={{ paddingBottom: "141.4%" }}>
1873:     
1878:   </div>
1879: </div>
1880:
1881: <Separator className="my-4" />
1882:
1883: { /* Allergies & Sensitivity */ }
1884: <div className="text-center mb-3">
1885:   <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded uppercase">
1886:     Allergies & Sensitivity
1887:   </h2>
1888: </div>
1889:
1890: {typeof allergiesAndSensitivity === "object" &&
1891:   ("quote" in allergiesAndSensitivity ||
1892:     "description" in allergiesAndSensitivity) &&
1893:   ((allergiesAndSensitivity as any).quote ||
1894:     (allergiesAndSensitivity as any).description) && (
1895:     <div className="mb-3 space-y-2">
1896:       { "quote" in allergiesAndSensitivity &&
1897:         (allergiesAndSensitivity as any).quote && (
1898:           <div className="text-sm font-semibold italic text-center text-red-600 mb-3">
1899:             "{(allergiesAndSensitivity as any).quote}"
1900:           </div>
1901:         )}
1902:       { "description" in allergiesAndSensitivity &&
1903:         (allergiesAndSensitivity as any).description && (
1904:           <p className="text-gray-600 text-justify leading-relaxed mb-4 p-3 text-xs">
1905:             "{(allergiesAndSensitivity as any).description}"
1906:           </p>
1907:         )}
1908:     </div>
1909:   )}
1910:
1911: <div className="overflow-x-auto shadow mb-4">
1912:   <table className="min-w-full text-xs text-center border border-gray-300 table-fixed">
1913:     <thead className="bg-orange-600 text-white">
1914:       <tr>
1915:         <th className="w-[35%] p-2 border border-gray-300">
1916:           ALLERGIES
1917:         </th>
1918:         <th className="w-[65%] p-2 border border-gray-300">
1919:           TENDENCY OF ALLERGY
1920:         </th>
1921:       </tr>
1922:     </thead>
1923:     <tbody>
1924:       {allergiesAndSensitivity.data &&
1925:         Object.entries(allergiesAndSensitivity.data).map(
1926:           ([key, entry], index) => (
1927:             <tr>
1928:               <td>
1929:                 key={key}
1930:                 className={index % 2 === 0 ? "bg-white" : "bg-gray-50"}
1931:               >
1932:                 <td className="p-2 border border-gray-300 font-medium text-center capitalize">
1933:                   {entry.title || `Allergy ${index + 1}`}
1934:                 </td>
1935:                 <td>
1936:                   <div>
1937:                     <div>
1938:                       <div>
1939:                         {entry.image ? (

```

```

1940:                 src={`~/allergies/${entry.image}.png`}
1941:                 alt={entry.title}
1942:                 className="max-w-[170px] max-h-[130px] object-contain mx-auto"
1943:                 onError={(e) => {
1944:                     e.currentTarget.onerror = null;
1945:                     e.currentTarget.src =
1946:                         "/allergies/placeholder.png";
1947:                 }}
1948:             />
1949:         ) : (
1950:             <span className="text-slate-400 italic">
1951:                 No Image
1952:             </span>
1953:         )}
1954:     </td>
1955: </tr>
1956: )
1957: )}
1958: </tbody>
1959: </table>
1960: </div>
1961:
1962: {/* General Advice */}
1963: {typeof allergiesAndSensitivity === "object" &&
1964:     "generalAdvice" in allergiesAndSensitivity &&
1965:     typeof allergiesAndSensitivity.generalAdvice === "string" &&
1966:     allergiesAndSensitivity.generalAdvice.trim() && (
1967:         <div className="rounded shadow border border-gray-300 mt-4 overflow-hidden">
1968:             {/* <div className="bg-orange-600 text-white px-2 py-2 font-semibold text-xs text-left">
1969:                 General Advice
1970:             </div> */}
1971:             <div className="bg-white p-2 text-xs text-justify whitespace-pre-line break-words leading-re
1972:                 {allergiesAndSensitivity.generalAdvice}
1973:             </div>
1974:         </div>
1975:     )}
1976:
1977: <Separator className="my-4" />
1978:
1979: <div className="w-full max-w-4xl">
1980:     <div className="relative w-full" style={{ paddingBottom: "141.4%" }}>
1981:         
1986:     </div>
1987: </div>
1988:
1989: <Separator className="my-4" />
1990: {/* Preventive Health */}
1991: <div className="text-center mb-3">
1992:     <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
1993:         PREVENTIVE HEALTH
1994:     </h2>
1995: </div>
1996:
1997: <div className="grid grid-cols-2 gap-6">
1998:     {/* Diagnostic Tests Table */}
1999:     <table className="min-w-full border border-gray-300 text-sm text-left">
2000:         <thead>
2001:             <tr className="bg-gray-400 text-white uppercase">
2002:                 <th
2003:                     colspan={3}
2004:                     className="px-3 py-2 border text-black border-gray-400 text-center"
2005:                 >
2006:                     Diagnostic Tests
2007:                 </th>
2008:             </tr>
2009:             <tr className="bg-gray-300 text-gray-900 uppercase">
2010:                 <th className="px-3 py-2 border border-gray-400 text-center">
2011:                     Tests
2012:                 </th>

```

```

2013:         <th className="px-3 py-2 border border-gray-400 text-center">
2014:             Half Yearly
2015:         </th>
2016:         <th className="px-3 py-2 border border-gray-400 text-center">
2017:             Yearly
2018:         </th>
2019:     </tr>
2020: </thead>
2021: <tbody>
2022:     {[
2023:         ...preventiveHealth.diagnosticTests.halfYearly.map((test) => ({
2024:             test,
2025:             half: "YES",
2026:             year: "",
2027:         })),
2028:         ...preventiveHealth.diagnosticTests.yearly.map((test) => ({
2029:             test,
2030:             half: "",
2031:             year: "YES",
2032:         })),
2033:     ].map((row, index) => (
2034:         <tr
2035:             key={index}
2036:             className={index % 2 === 0 ? "bg-gray-100" : "bg-white"}
2037:         >
2038:             <td className="px-3 py-2 border border-gray-300">
2039:                 {row.test}
2040:             </td>
2041:             <td className="px-3 py-2 border border-gray-300 text-center">
2042:                 {row.half}
2043:             </td>
2044:             <td className="px-3 py-2 border border-gray-300 text-center">
2045:                 {row.year}
2046:             </td>
2047:         </tr>
2048:     )]}
2049: </tbody>
2050: </table>
2051:
2052: {/* Nutritional Supplements Table */}
2053: <table className="min-w-full border border-gray-300 text-sm text-left">
2054:     <thead>
2055:         <tr className="bg-blue-500 text-white uppercase">
2056:             <th
2057:                 colspan={2}
2058:                 className="px-3 py-2 border border-blue-500 text-center"
2059:             >
2060:                 Nutritional Supplements
2061:             </th>
2062:         </tr>
2063:         <tr className="bg-blue-300 text-blue-900 uppercase">
2064:             <th className="px-3 py-2 border border-blue-300 text-center">
2065:                 Supplements
2066:             </th>
2067:             <th className="px-3 py-2 border border-blue-300 text-center">
2068:                 Yes/No
2069:             </th>
2070:         </tr>
2071:     </thead>
2072:     <tbody>
2073:         {preventiveHealth.nutritionalSupplements.map(
2074:             (supplement, index) => (
2075:                 <tr
2076:                     key={index}
2077:                     className={index % 2 === 0 ? "bg-blue-100" : "bg-white"}
2078:                 >
2079:                     <td className="px-3 py-2 border border-gray-300">
2080:                         {supplement.supplement}
2081:                     </td>
2082:                     <td className="px-3 py-2 border border-gray-300 text-center">
2083:                         {supplement.needed ? "YES" : ""}
2084:                     </td>
2085:                 </tr>

```

```

2086:         )
2087:     })
2088:     </tbody>
2089: </table>
2090: </div>
2091:
2092: { /* Description Section at the Bottom */}
2093:
2094: <p className="text-sm text-black whitespace-pre-line text-justify mt-6">
2095:     {preventiveHealth.description || "No description provided."}
2096: </p>
2097:
2098: <Separator className="my-4" />
2099: { /* Family Genetic Impact */}
2100: <div className="text-center mb-3">
2101:     <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
2102:         FAMILY GENETIC IMPACT
2103:     </h2>
2104: </div>
2105: <div className="overflow-x-auto">
2106:     {familyGeneticImpactSection.description && (
2107:         <p className="text-gray-600 text-sm leading-relaxed mb-4 p-3 whitespace-pre-line">
2108:             {familyGeneticImpactSection.description}
2109:         </p>
2110:     )}
2111:
2112:     {familyGeneticImpactSection.impacts.length === 0 ? (
2113:         <p className="text-center text-gray-500 italic">
2114:             No family genetic impacts recorded.
2115:         </p>
2116:     ) : (
2117:         <table className="w-full border border-blue-500 text-xs">
2118:             <thead className="bg-blue-500 text-white text-center uppercase text-xs">
2119:                 <tr>
2120:                     <th className="py-2 px-2 w-[20%]">Genes</th>
2121:                     <th className="py-2 px-2 w-[30%]">Normal/Common Alleles</th>
2122:                     <th className="py-2 px-2 w-[15%]">Your Result</th>
2123:                     <th className="py-2 px-2 w-[35%]">Impact on Health</th>
2124:                 </tr>
2125:             </thead>
2126:             <tbody className="text-center">
2127:                 {familyGeneticImpactSection.impacts.map((impact, index) => (
2128:                     <tr key={index} className="border-t border-slate-200">
2129:                         <td className="px-2 py-2 font-bold text-blue-700 uppercase border-r border-slate-200">
2130:                             {impact.gene}
2131:                         </td>
2132:                         <td className="px-2 py-2">{impact.normalAlleles}</td>
2133:                         <td className="px-2 py-2 font-mono">{impact.yourResult}</td>
2134:                         <td className="px-2 py-2 text-center">
2135:                             {impact.healthImpact}
2136:                         </td>
2137:                     </tr>
2138:                 )})
2139:             </tbody>
2140:         </table>
2141:     )}
2142:
2143: <div className="flex justify-center items-center">
2144:     
2149: </div>
2150: </div>
2151:
2152: <Separator className="my-4" />
2153: { /* ===== Genomic Analysis Section (Moved Up) ===== */}
2154: <div className="text-center mb-3">
2155:     <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
2156:         GENOMIC ANALYSIS
2157:     </h2>
2158: </div>

```



```

2159:
2160:     { /* Show description if available */ }
2161:     {reportData.genomicAnalysisTable?.description && (
2162:         <p className="text-gray-600 text-justify leading-relaxed mb-4 p-3 text-xs">
2163:             {reportData.genomicAnalysisTable.description}
2164:         </p>
2165:     )}
2166:     <div className="mb-4 overflow-x-auto">
2167:         {reportData.genomicAnalysisTable?.categories?.length === 0 ? (
2168:             <p className="text-center text-gray-500 italic">
2169:                 No genomic analysis data recorded.
2170:             </p>
2171:         ) : (
2172:             <table className="w-full border border-blue-500 text-xs text-center">
2173:                 <thead className="text-white bg-blue-400 uppercase text-xs">
2174:                     <tr>
2175:                         <th className="py-2 px-2 w-[20%] border border-blue-500">
2176:                             Category
2177:                         </th>
2178:                         <th className="py-2 px-2 w-[22%] border border-blue-500">
2179:                             Area
2180:                         </th>
2181:                         <th className="py-2 px-2 w-[28%] border border-blue-500">
2182:                             Traits
2183:                         </th>
2184:                         <th className="py-2 px-2 w-[30%] border border-blue-500">
2185:                             Genes
2186:                         </th>
2187:                     </tr>
2188:                 </thead>
2189:                 <tbody>
2190:                     {reportData.genomicAnalysisTable?.categories?.map(
2191:                         (cat, catIndex) => {
2192:                             // Sort by area while keeping other data intact
2193:                             const sortedSubs = [...(cat.subcategories || [])].sort(
2194:                                 (a, b) => (a.area || "").localeCompare(b.area || "")
2195:                             );
2196:
2197:                             const subLen = sortedSubs.length || 1;
2198:                             return sortedSubs.length > 0 ? (
2199:                                 sortedSubs.map((sub, subIndex) => (
2200:                                     <tr>
2201:                                         <th key={` ${catIndex}-${subIndex}`}
2202:                                             className="bg-white"
2203:                                         >
2204:                                             {subIndex === 0 && (
2205:                                                 <td>
2206:                                                     className="px-3 py-2 border border-gray-300 text-center font-bold align-middle"
2207:                                                     rowspan={subLen}
2208:                                                 >
2209:                                                     {cat.category}
2210:                                                 </td>
2211:                                             )}
2212:                                                 <td className="px-3 py-2 border border-gray-300 text-center break-words text-xs">
2213:                                                     {sub.area || "-"}
2214:                                                 </td>
2215:                                                 <td className="px-3 py-2 border border-gray-300 text-center break-words text-xs">
2216:                                                     {sub.trait || "-"}
2217:                                                 </td>
2218:                                                 <td className="px-3 py-2 border border-gray-300 text-center break-words text-xs">
2219:                                                     {sub.genes?.join(", ") || "-"}
2220:                                                 </td>
2221:                                             </tr>
2222:                                         )}
2223:                                     ) : (
2224:                                         <tr key={` ${catIndex}-empty`} className="bg-white">
2225:                                             <td className="px-3 py-2 border border-gray-300 text-center font-bold align-middle">
2226:                                                 {cat.category}
2227:                                             </td>
2228:                                             <td>
2229:                                                 colspan={3}
2230:                                                 className="px-3 py-2 border border-gray-300 text-center italic text-gray-500 tex
2231:                                             >

```

```

2232:                 No rows added
2233:             </td>
2234:         </tr>
2235:     );
2236: }
2237: }}
2238: </tbody>
2239: </table>
2240: }}
2241: </div>
2242:
2243: { /* Now render the rest of the sections BELOW this */}
2244: <Separator className="my-4" />
2245: { /* Gene Test Results */}
2246: <div className="text-center mb-3">
2247:     <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
2248:         GENE TEST RESULTS
2249:     </h2>
2250: </div>
2251: <SectionTitle title="Gene Test Results" icon="" />
2252: <div className="mb-4 overflow-x-auto">
2253:     {geneTestResults.length === 0 ? (
2254:         <p className="text-center text-gray-500 italic">
2255:             No gene test results recorded.
2256:         </p>
2257:     ) : (
2258:         <table className="min-w-full border border-gray-200 text-xs text-left">
2259:             <thead className="bg-blue-300 text-white">
2260:                 <tr>
2261:                     <th className="px-3 py-2 border text-center uppercase">
2262:                         Gene Code
2263:                     </th>
2264:                     <th className="px-3 py-2 border text-center uppercase">
2265:                         Gene Name
2266:                     </th>
2267:                     <th className="px-3 py-2 border text-center uppercase">
2268:                         Variation
2269:                     </th>
2270:                     <th className="px-3 py-2 border text-center uppercase">
2271:                         Result
2272:                     </th>
2273:                 </tr>
2274:             </thead>
2275:             <tbody>
2276:                 {geneTestResults.map((result, index) => (
2277:                     <tr key={index}>
2278:                         <td className="px-3 py-2 border border-gray-300 font-mono whitespace-normal text-center">
2279:                             {result.genecode || "-"}
2280:                         </td>
2281:                         <td className="px-3 py-2 border border-gray-300 whitespace-normal text-center break-wo">
2282:                             {result.geneName || "-"}
2283:                         </td>
2284:                         <td className="px-3 py-2 border border-gray-300 font-mono whitespace-normal text-center">
2285:                             {result.variation || "-"}
2286:                         </td>
2287:                         <td className="px-3 py-2 border border-gray-300 font-mono font-bold whitespace-normal">
2288:                             {result.result || "-"}
2289:                         </td>
2290:                     </tr>
2291:                 ))}
2292:             </tbody>
2293:         </table>
2294:     )}
2295: </div>
2296: <Separator className="my-4" />
2297: { /* Enhanced Comprehensive Summary */}
2298: <div className="text-center mb-3">
2299:     <h2 className="text-base font-bold mb-2 bg-blue-600 text-white py-2 px-2 rounded">
2300:         COMPREHENSIVE REPORT SUMMARY
2301:     </h2>
2302: </div>
2303: { /* Generate comprehensive summary based on all data */}
2304: {(() => {

```

```

2305: // Analyze diet data in detail
2306: const dietAnalysis = Object.entries(groupedDietFields).reduce(
2307:   (acc, [category, fields]) => {
2308:     const scores = fields.map((f) => f.score);
2309:     const levels = fields.map((f) => f.level);
2310:     const highScores = fields.filter((f) => f.score >= 7);
2311:     const mediumScores = fields.filter(
2312:       (f) => f.score >= 4 && f.score < 7
2313:     );
2314:     const lowScores = fields.filter((f) => f.score < 4);
2315:     const avgScore =
2316:       scores.reduce((sum, score) => sum + score, 0) / scores.length ||
2317:       0;
2318:
2319:     const recommendations = fields.reduce((recAcc, field) => {
2320:       if (field.recommendations) {
2321:         Object.entries(field.recommendations).forEach(
2322:           ([level, rec]) => {
2323:             if (field.selectedLevel === level) {
2324:               recAcc.push(`${field.label}: ${rec}`);
2325:             }
2326:           }
2327:         );
2328:       } else if (field.recommendation) {
2329:         recAcc.push(`${field.label}: ${field.recommendation}`);
2330:       }
2331:       return recAcc;
2332:     }, [] as string[]);
2333:
2334:     acc[category] = {
2335:       highScores: highScores.length,
2336:       mediumScores: mediumScores.length,
2337:       lowScores: lowScores.length,
2338:       avgScore,
2339:       total: fields.length,
2340:       fields: fields,
2341:       recommendations: recommendations.slice(0, 3), // Top 3 recommendations
2342:     };
2343:     return acc;
2344:   },
2345:   {} as Record<
2346:     string,
2347:     {
2348:       highScores: number;
2349:       mediumScores: number;
2350:       lowScores: number;
2351:       avgScore: number;
2352:       total: number;
2353:       fields: any[];
2354:       recommendations: string[];
2355:     }
2356:   >
2357: );
2358:
2359: // Analyze nutrition data in detail
2360: const nutritionAnalysis = Object.entries(
2361:   reportData.nutritionData?.data || {}
2362: ).reduce(
2363:   (acc, [section, data]) => {
2364:     const entries = Object.entries(data);
2365:     const highImpact = entries.filter(
2366:       ([_, item]) =>
2367:         item.healthImpact?.toLowerCase().includes("high") ||
2368:         item.healthImpact?.toLowerCase().includes("increased")
2369:     );
2370:     const lowImpact = entries.filter(
2371:       ([_, item]) =>
2372:         item.healthImpact?.toLowerCase().includes("low") ||
2373:         item.healthImpact?.toLowerCase().includes("reduced")
2374:     );
2375:     const normalImpact = entries.filter(
2376:       ([_, item]) =>
2377:         item.healthImpact?.toLowerCase().includes("normal") ||

```

```

2378:         item.healthImpact?.toLowerCase().includes("average")
2379:     );
2380:     const avgScore =
2381:         entries.reduce((sum, [_ , item]) => sum + (item.score || 0), 0) /
2382:         entries.length || 0;
2383:
2384:     const detailedItems = entries.map(([key, item]) => ({
2385:         name: key
2386:             .replace(/[A-Z]/g, " $1")
2387:             .replace(/^.\/, (str) => str.toUpperCase()),
2388:         score: item.score,
2389:         impact: item.healthImpact,
2390:         level: item.intakeLevel,
2391:         source: item.source,
2392:     }));
2393:
2394:     acc[section] = {
2395:         highImpact: highImpact.length,
2396:         lowImpact: lowImpact.length,
2397:         normalImpact: normalImpact.length,
2398:         avgScore,
2399:         total: entries.length,
2400:         items: detailedItems,
2401:     };
2402:     return acc;
2403: },
2404: {} as Record<
2405:     string,
2406:     {
2407:         highImpact: number;
2408:         lowImpact: number;
2409:         normalImpact: number;
2410:         avgScore: number;
2411:         total: number;
2412:         items: any[];
2413:     }
2414: >
2415: );
2416:
2417: // Analyze lifestyle conditions in detail
2418: const lifestyleAnalysis = Object.entries(
2419:     reportData.lifestyleConditions || {}
2420: )
2421:     .filter(([key]) => !["quote", "description"].includes(key))
2422:     .reduce(
2423:         (acc, [category, conditions]) => {
2424:             const entries = Object.entries(
2425:                 conditions as Record<string, HealthConditionStatus>
2426:             );
2427:             const strengths = entries.filter(
2428:                 ([_, data]) => data.status === "strength"
2429:             );
2430:             const improvements = entries.filter(
2431:                 ([_, data]) => data.status === "improvement"
2432:             );
2433:
2434:             const detailedConditions = entries.map(([condition, data]) => ({
2435:                 name: condition,
2436:                 status: data.status,
2437:                 description: data.description,
2438:                 sensitivity: data.sensitivity,
2439:                 recommendations: {
2440:                     avoid: data.avoid || [],
2441:                     follow: data.follow || [],
2442:                     consume: data.consume || [],
2443:                     monitor: data.monitor || [],
2444:                 },
2445:             }));
2446:
2447:             acc[category] = {
2448:                 strengths: strengths.length,
2449:                 improvements: improvements.length,
2450:                 total: entries.length,

```

```

2451:         conditions: detailedConditions,
2452:         hasImages: !!reportData.lifestyleCategoryImages?.[category],
2453:     };
2454:     return acc;
2455: },
2456: {} as Record<
2457: string,
2458: {
2459:     strengths: number;
2460:     improvements: number;
2461:     total: number;
2462:     conditions: any[];
2463:     hasImages: boolean;
2464: }
2465: >
2466: );
2467:
2468: // Analyze metabolic core in detail
2469: const metabolicAnalysis = Object.entries(metabolicCore)
2470:   .filter(([key]) => key !== "quote" && key !== "description")
2471:   .reduce(
2472:     (acc, [area, data]) => {
2473:       let genes: any[] = [];
2474:       if (
2475:         typeof data === "object" &&
2476:         data !== null &&
2477:         "genes" in data &&
2478:         Array.isArray((data as any).genes)
2479:       ) {
2480:         genes = (data as any).genes;
2481:       }
2482:
2483:       // Filter genes by impact level
2484:       const highImpact = genes.filter((g) =>
2485:         g.impact?.toLowerCase().includes("high")
2486:       );
2487:
2488:       const lowImpact = genes.filter((g) =>
2489:         g.impact?.toLowerCase().includes("low")
2490:       );
2491:
2492:       const moderateImpact = genes.filter(
2493:         (g) =>
2494:           g.impact?.toLowerCase().includes("moderate") ||
2495:           g.impact?.toLowerCase().includes("average")
2496:       );
2497:
2498:       // Add normal impact filter
2499:       const normalImpact = genes.filter((g) =>
2500:         g.impact?.toLowerCase().includes("normal")
2501:       );
2502:
2503:       const detailedGenes = genes.map((gene) => ({
2504:         name: gene.name,
2505:         genotype: gene.genotype,
2506:         impact: gene.impact,
2507:         advice: gene.advice,
2508:       }));
2509:
2510:       acc[area] = {
2511:         highImpact: highImpact.length,
2512:         moderateImpact: moderateImpact.length,
2513:         normalImpact: normalImpact.length, // Add this line
2514:         lowImpact: lowImpact.length,
2515:         total: genes.length,
2516:         genes: detailedGenes,
2517:       };
2518:
2519:       return acc;
2520:     },
2521:     {} as Record<
2522: string,
2523: {

```

```

2524:         highImpact: number;
2525:         moderateImpact: number;
2526:         normalImpact: number; // Add this to type definition
2527:         lowImpact: number;
2528:         total: number;
2529:         genes: any[];
2530:     }
2531:     >
2532: );
2533:
2534: // Analyze sports and fitness in detail
2535: const sportsAnalysis = Object.entries(sportsAndFitness)
2536:   .filter(
2537:     ([key]) => ![ "quote", "description", "customImages" ].includes(key)
2538:   )
2539:   .reduce((acc, [section, groups]) => {
2540:     if (Array.isArray(groups)) {
2541:       const allFields = groups.reduce((fieldAcc, group) => {
2542:         return fieldAcc.concat(Object.entries(group.fields || {}));
2543:       }, [] as any[]);
2544:
2545:       const detailedTraits = allFields.map(
2546:         ([trait, data]: [string, any]) => {
2547:           const selectedImageKey = data.level
2548:             ?.split("-")
2549:             .pop()
2550:             ?.toLowerCase();
2551:           const hasImage = !! (
2552:             reportData.sportsAndFitness.customImages?.[
2553:               selectedImageKey
2554:             ] || defaultImageMap[selectedImageKey]
2555:           );
2556:
2557:           return {
2558:             name: trait,
2559:             level: data.level,
2560:             description: data.description,
2561:             label: data.label,
2562:             imageKey: selectedImageKey,
2563:             hasImage: hasImage,
2564:           };
2565:         }
2566:       );
2567:
2568:       acc[section] = {
2569:         totalTraits: allFields.length,
2570:         traits: detailedTraits,
2571:       };
2572:     }
2573:     return acc;
2574:   }, {} as Record<string, { totalTraits: number; traits: any[] }>);
2575:
2576: // Analyze digestive health in detail
2577: const digestiveAnalysis = Object.entries(
2578:   digestiveHealth.data || {}
2579: ).reduce((acc, [key, item]) => {
2580:   acc[key] = {
2581:     title: item.title,
2582:     sensitivity: item.sensitivity,
2583:     good: item.good,
2584:     bad: item.bad,
2585:     icon: item.icon,
2586:     hasIcon: !!item.icon,
2587:   };
2588:   return acc;
2589: }, {} as Record<string, any>);
2590:
2591: // Analyze genes & addiction in detail
2592: // Analyze genes & addiction in detail
2593: const addictionAnalysis = Object.entries(
2594:   genesAndAddiction.data || {}
2595: ).reduce((acc, [key, entry]) => {
2596:   acc[key] = {

```

```

2597:         title: entry.title,
2598:         icon: entry.icon, // Add this line
2599:         sensitivityIcon: entry.sensitivityIcon, // Add this line
2600:         hasIcon: !!entry.icon,
2601:         hasSensitivityIcon: !!entry.sensitivityIcon,
2602:     };
2603:     return acc;
2604: }, {} as Record<string, any>);
2605:
2606: // Analyze sleep & rest in detail
2607: const sleepAnalysis = Object.entries(sleepAndRest.data || {}).reduce(
2608:     (acc, [key, entry]) => {
2609:         acc[key] = {
2610:             title: entry.title,
2611:             intervention: entry.intervention,
2612:             image: (entry as any).image,
2613:             hasImage: !!entry.image,
2614:         };
2615:         return acc;
2616:     },
2617:     {} as Record<string, any>
2618: );
2619:
2620: // Analyze allergies in detail
2621: const allergyAnalysis = Object.entries(
2622:     allergiesAndSensitivity.data || {}
2623: ).reduce((acc, [key, entry]) => {
2624:     const allergyEntry = entry as { title?: string; image?: string };
2625:     acc[key] = {
2626:         title: allergyEntry.title,
2627:         image: allergyEntry.image,
2628:         hasImage: !!allergyEntry.image,
2629:     };
2630:     return acc;
2631: }, {} as Record<string, any>);
2632:
2633: // Calculate overall scores and metrics
2634: const totalDietScore =
2635:     Object.values(dietAnalysis).reduce(
2636:         (sum, cat) => sum + cat.avgScore,
2637:         0
2638:     ) / Object.keys(dietAnalysis).length || 0;
2639: const totalNutritionScore =
2640:     Object.values(nutritionAnalysis).reduce(
2641:         (sum, cat) => sum + cat.avgScore,
2642:         0
2643:     ) / Object.keys(nutritionAnalysis).length || 0;
2644: const totalLifestyleStrengths = Object.values(
2645:     lifestyleAnalysis
2646: ).reduce((sum, cat) => sum + cat.strengths, 0);
2647: const totalLifestyleImprovements = Object.values(
2648:     lifestyleAnalysis
2649: ).reduce((sum, cat) => sum + cat.improvements, 0);
2650: const totalMetabolicHigh = Object.values(metabolicAnalysis).reduce(
2651:     (sum, area) => sum + area.highImpact,
2652:     0
2653: );
2654: const totalMetabolicLow = Object.values(metabolicAnalysis).reduce(
2655:     (sum, area) => sum + area.lowImpact,
2656:     0
2657: );
2658: const totalMetabolicModerate = Object.values(
2659:     metabolicAnalysis
2660: ).reduce((sum, area) => sum + area.moderateImpact, 0);
2661:
2662: const totalMetabolicNormal = Object.values(metabolicAnalysis).reduce(
2663:     (total, area) => total + (area.normalImpact || 0),
2664:     0
2665: );
2666:
2667: return (
2668:     <div className="space-y-4 mb-4">
2669:         /* Detailed Diet Analysis */

```

```

2670:         {Object.keys(dietAnalysis).length > 0 && (
2671:             <Card className="border shadow-sm">
2672:                 <CardHeader className="pb-2">
2673:                     <CardTitle className="text-sm font-bold text-center mb-5 text-green-800">
2674:                         COMPREHENSIVE DIETARY GENETIC ANALYSIS
2675:                     </CardTitle>
2676:                 </CardHeader>
2677:
2678:                 <CardContent className="text-xs text-gray-700">
2679:                     <div className="w-full h-full text-center border rounded p-2 bg-gradient-to-br from-gr
2680:                         <p className="font-semibold text-green-700 mb-1">
2681:                             GENOMICS & DIET
2682:                         </p>
2683:                     <div className="h-80">
2684:                         { " " }
2685:                         { /* doubled height */ }
2686:                     <ResponsiveContainer width="100%" height="100%">
2687:                         <LineChart
2688:                             data={Object.values(groupedDietFields).flatMap(
2689:                                 (fields) =>
2690:                                     fields.map((data) => ({
2691:                                         field: data.label,
2692:                                         score: Number(
2693:                                             data.score ??
2694:                                             (data.high ? 8 : data.medium ? 6 : 4)
2695:                                         ),
2696:                                     })
2697:                                 )}
2698:                             margin={{ top: 10, right: 8, left: 30, bottom: 60 }}
2699:                         >
2700:                             <XAxis
2701:                                 dataKey="field"
2702:                                 axisLine
2703:                                 tickLine
2704:                                 tick={{ fontSize: 9, fill: "#16a34a" }}
2705:                                 stroke="#16a34a"
2706:                                 height={40}
2707:                                 angle={-45}
2708:                                 textAnchor="end"
2709:                                 interval={0}
2710:                                 tickFormatter={(value) => {
2711:                                     const str = String(value).toLowerCase();
2712:                                     return (
2713:                                         str.charAt(0).toUpperCase() + str.slice(1)
2714:                                     );
2715:                                 }}
2716:                             />
2717:
2718:                             <YAxis
2719:                                 axisLine
2720:                                 tickLine
2721:                                 tick={{ fontSize: 9, fill: "#16a34a" }}
2722:                                 stroke="#16a34a"
2723:                                 width={30}
2724:                                 domain={[1, 10]}
2725:                                 ticks={[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]}
2726:                             />
2727:                             <Tooltip
2728:                                 contentStyle={{
2729:                                     backgroundColor: "#f0fdf4",
2730:                                     border: "1px solid #16a34a",
2731:                                     borderRadius: "4px",
2732:                                     fontSize: "10px",
2733:                                 }}
2734:                                 formatter={(value, name, props) => [
2735:                                     value,
2736:                                     props.payload.field,
2737:                                 ]}
2738:                             />
2739:                             <Line
2740:                                 type="monotone"
2741:                                 dataKey="score"
2742:                                 stroke="#16a34a"

```



```

2743:         strokeWidth={2}
2744:         dot={{ fill: "#16a34a", strokeWidth: 1, r: 3 }}
2745:         activeDot={{
2746:             r: 4,
2747:             fill: "#16a34a",
2748:             stroke: "#fff",
2749:             strokeWidth: 1,
2750:         }}
2751:     />
2752: </LineChart>
2753: </ResponsiveContainer>
2754: </div>
2755: <p className="text-xs mt-1">
2756:     Diet-related genetic scores
2757: </p>
2758: </div>
2759:
2760: <div className="mb-3 p-2">
2761:     <p className="font-semibold my-3">
2762:         OVERALL DIET COMPATIBILITY: {totalDietScore.toFixed(1)}
2763:         /10
2764:     </p>
2765:     <div className="grid grid-cols-3 gap-2 text-center">
2766:         <div>
2767:             <p className="text-green-600 font-bold">
2768:                 {Object.values(dietAnalysis).reduce(
2769:                     (sum, cat) => sum + cat.highScores,
2770:                     0
2771:                 )}
2772:             </p>
2773:             <p className="text-xs">High Performers (7-10)</p>
2774:         </div>
2775:         <div>
2776:             <p className="text-yellow-600 font-bold">
2777:                 {Object.values(dietAnalysis).reduce(
2778:                     (sum, cat) => sum + cat.mediumScores,
2779:                     0
2780:                 )}
2781:             </p>
2782:             <p className="text-xs">Moderate (4-6)</p>
2783:         </div>
2784:         <div>
2785:             <p className="text-red-600 font-bold">
2786:                 {Object.values(dietAnalysis).reduce(
2787:                     (sum, cat) => sum + cat.lowScores,
2788:                     0
2789:                 )}
2790:             </p>
2791:             <p className="text-xs">Need Attention (0-3)</p>
2792:         </div>
2793:     </div>
2794: </div>
2795:
2796: { /* Visual Graph Section */ }
2797:
2798: <div className="mb-10 h-70">
2799:     <ResponsiveContainer width="100%" height="100%">
2800:         <LineChart
2801:             data={Object.entries(dietAnalysis).map(
2802:                 ([category, data]) => ({
2803:                     category:
2804:                         category.charAt(0).toUpperCase() +
2805:                         category.slice(1),
2806:                     avgScore: Number(data.avgScore.toFixed(1)),
2807:                     highScores: data.highScores,
2808:                     mediumScores: data.mediumScores,
2809:                     lowScores: data.lowScores,
2810:                 })
2811:             )}
2812:             margin={{ top: 10, right: 8, left: 35, bottom: 60 }}
2813:             >
2814:             <XAxis
2815:                 dataKey="category"

```

```

2816:         axisLine
2817:         tickLine
2818:         tick={{ { fontSize: 9, fill: "#16a34a" } }}
2819:         stroke="#16a34a"
2820:         height={30}
2821:         angle={-45}
2822:         textAnchor="end"
2823:         interval={0}
2824:     />
2825: <YAxis
2826:     axisLine
2827:     tickLine
2828:     tick={{ { fontSize: 9, fill: "#16a34a" } }}
2829:     stroke="#16a34a"
2830:     width={30}
2831:     domain={[0, 10]}
2832:     ticks={[0, 2, 4, 6, 8, 10]}
2833: />
2834: <Tooltip
2835:     contentStyle={{
2836:         backgroundColor: "#f0fdf4",
2837:         border: "1px solid #16a34a",
2838:         borderRadius: "4px",
2839:         fontSize: "10px",
2840:     }}
2841:     formatter={(value, name, props) => {
2842:         if (name === "avgScore") {
2843:             return [`${value}/10`, "Average Score"];
2844:         }
2845:         return [value, name];
2846:     }}
2847: />
2848: <Line
2849:     type="monotone"
2850:     dataKey="avgScore"
2851:     stroke="#16a34a"
2852:     strokeWidth={2}
2853:     dot={{ fill: "#16a34a", strokeWidth: 1, r: 4 }}
2854:     activeDot={{
2855:         r: 5,
2856:         fill: "#16a34a",
2857:         stroke: "#fff",
2858:         strokeWidth: 2,
2859:     }}
2860: />
2861: </LineChart>
2862: </ResponsiveContainer>
2863: </div>
2864:
2865: <div className="grid grid-cols-2 gap-3">
2866:     {Object.entries(dietAnalysis).map(([category, data]) => (
2867:         <div key={category} className="rounded p-2">
2868:             <p className="font-semibold text-center capitalize mb-1">
2869:                 {category}
2870:             </p>
2871:             <div className="mb-2">
2872:                 <p>
2873:                     Average Score:{" "}
2874:                     <span
2875:                         className={`font-bold ${
2876:                             data.avgScore >= 6
2877:                             ? "text-green-600"
2878:                             : data.avgScore >= 4
2879:                             ? "text-yellow-600"
2880:                             : "text-red-600"
2881:                         }`}
2882:                     >
2883:                         {data.avgScore.toFixed(1)}/10
2884:                     </span>
2885:                 </p>
2886:                 <p>
2887:                     Distribution: {data.highScores}High /{" "}
2888:                     {data.mediumScores}Medium / {data.lowScores}Low

```

```

2889:         </p>
2890:     </div>
2891:
2892:     { /* Mini Bar Chart for Distribution */ }
2893:     <div className=" p-1 h-auto mt-3 rounded">
2894:         <div className="h-40">
2895:             <ResponsiveContainer width="100%" height="100%">
2896:                 <BarChart
2897:                     data={[
2898:                         {
2899:                             name: "High",
2900:                             value: data.highScores,
2901:                             color: "#dc2626",
2902:                         },
2903:                         {
2904:                             name: "Medium",
2905:                             value: data.mediumScores,
2906:                             color: "#ca8a04",
2907:                         },
2908:                         {
2909:                             name: "Low",
2910:                             value: data.lowScores,
2911:                             color: "#16a34a",
2912:                         },
2913:                     ]}
2914:                     margin={{
2915:                         top: 5,
2916:                         right: 5,
2917:                         left: 5,
2918:                         bottom: 15,
2919:                     }}
2920:                 >
2921:                     <XAxis
2922:                         dataKey="name"
2923:                         axisLine={false}
2924:                         tickLine={false}
2925:                         tick={{ fontSize: 8, fill: "#374151" }}
2926:                     />
2927:                     <YAxis hide />
2928:                     <Tooltip
2929:                         contentStyle={{
2930:                             backgroundColor: "#f9fafb",
2931:                             border: "1px solid #d1d5db",
2932:                             borderRadius: "4px",
2933:                             fontSize: "10px",
2934:                             height: "full",
2935:                             borderBottomColor: "black",
2936:                         }}
2937:                         formatter={(value, name) => [
2938:                             `${value} genes`,
2939:                             name,
2940:                         ]}
2941:                     />
2942:                     <Bar
2943:                         dataKey="value"
2944:                         fill="#16a34a"
2945:                         radius={[2, 2, 0, 0]}
2946:                     >
2947:                         {[
2948:                             {
2949:                                 name: "High",
2950:                                 value: data.highScores,
2951:                                 color: "#dc2626",
2952:                             },
2953:                             {
2954:                                 name: "Medium",
2955:                                 value: data.mediumScores,
2956:                                 color: "#ca8a04",
2957:                             },
2958:                             {
2959:                                 name: "Low",
2960:                                 value: data.lowScores,
2961:                                 color: "#16a34a",

```

```

2962:         },
2963:         ].map((entry, index) => (
2964:             <Cell
2965:                 key={`cell-${index}`}
2966:                 fill={entry.color}
2967:             />
2968:         )))
2969:     </Bar>
2970: </BarChart>
2971: </ResponsiveContainer>
2972: </div>
2973: </div>
2974:
2975: {data.recommendations.length > 0 && (
2976:     <div>
2977:         <p className="font-semibold text-xs mb-1">
2978:             Key Recommendations:
2979:         </p>
2980:         <ul className="list-disc list-inside space-y-0">
2981:             {data.recommendations.map((rec, idx) => (
2982:                 <li
2983:                     key={idx}
2984:                     className="text-xs truncate"
2985:                     title={rec}
2986:                 >
2987:                     {rec}
2988:                 </li>
2989:             ))}
2990:         </ul>
2991:     </div>
2992: )}
2993: </div>
2994: )}
2995: </div>
2996: </CardContent>
2997: </Card>
2998: )}
2999:
3000: /* Detailed Nutrition Analysis */
3001: {Object.keys(nutritionAnalysis).length > 0 && (
3002:     <Card className="border shadow-sm">
3003:         <CardHeader className="pb-2">
3004:             <CardTitle className="text-sm font-bold text-center text-blue-800">
3005:                 DETAILED NUTRITIONAL GENOMICS PROFILE
3006:             </CardTitle>
3007:         </CardHeader>
3008:         <CardContent className="text-xs text-gray-700">
3009:             <div className="w-full print:w-[700px] mx-auto mb-6 print:mb-4">
3010:                 /* Chart for Nutrition */
3011:                 <div className="mb-4 text-center border rounded p-2 bg-gradient-to-br from-blue-50 to-white">
3012:                     <p className="font-semibold text-blue-700 mb-1">
3013:                         NUTRITION SCORE OVERVIEW
3014:                     </p>
3015:                     <div className="h-80">
3016:                         <ResponsiveContainer width="100%" height="100%">
3017:                             <LineChart
3018:                                 data={Object.entries(
3019:                                     reportData.nutritionData?.data || {}
3020:                                 ).flatMap(([,sectionName, sectionData]) =>
3021:                                     Object.entries(sectionData).map(
3022:                                         ([key, data]) => ({
3023:                                             field: key
3024:                                                 .replace(/([A-Z])/g, " $1")
3025:                                                 .replace(/^../, (str) =>
3026:                                                     str.toUpperCase()
3027:                                                 )
3028:                                             ,
3029:                                             score: Number(
3030:                                                 data.score ??
3031:                                                 (data.high ? 8 : data.medium ? 6 : 4)
3032:                                             )
3033:                                         })
3034:                                     )
3035:                                 )}
3036:                             </LineChart>
3037:                         </ResponsiveContainer>
3038:                     </div>
3039:                 </div>
3040:             </div>
3041:         </CardContent>
3042:     </Card>
3043: )}

```

```

3035:         margin={{
3036:             top: 10,
3037:             right: 8,
3038:             left: 30,
3039:             bottom: 60,
3040:         }}
3041:     >
3042:         <XAxis
3043:             dataKey="field"
3044:             axisLine
3045:             tickLine
3046:             tick={{ fontSize: 9, fill: "#2563eb" }}
3047:             stroke="#2563eb"
3048:             height={40}
3049:             angle={-45}
3050:             textAnchor="end"
3051:             interval={0}
3052:         />
3053:         <YAxis
3054:             axisLine
3055:             tickLine
3056:             tick={{ fontSize: 9, fill: "#2563eb" }}
3057:             stroke="#2563eb"
3058:             width={30}
3059:             domain={[1, 10]}
3060:             ticks={[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]}
3061:         />
3062:         <Tooltip
3063:             contentStyle={{
3064:                 backgroundColor: "#eff6ff",
3065:                 border: "1px solid #2563eb",
3066:                 borderRadius: "4px",
3067:                 fontSize: "10px",
3068:             }}
3069:             formatter={(value, name, props) => [
3070:                 value,
3071:                 props.payload.field,
3072:             ]}
3073:         />
3074:         <Line
3075:             type="monotone"
3076:             dataKey="score"
3077:             stroke="#2563eb"
3078:             strokeWidth={2}
3079:             dot={{ fill: "#2563eb", strokeWidth: 1, r: 3 }}
3080:             activeDot={{
3081:                 r: 4,
3082:                 fill: "#2563eb",
3083:                 stroke: "#fff",
3084:                 strokeWidth: 1,
3085:             }}
3086:         />
3087:     </LineChart>
3088: </ResponsiveContainer>
3089: </div>
3090: <p className="text-xs mt-1">
3091:     Nutrient-level genetic scores
3092: </p>
3093: </div>
3094: </div>
3095: <div className="mb-3 p-2 ">
3096:     <p className="font-semibold my-3">
3097:         NUTRITION GENETIC SCORE:{" "}
3098:         {totalNutritionScore.toFixed(1)}/10
3099:     </p>
3100:     <div className="grid grid-cols-3 gap-2 text-center">
3101:         <div>
3102:             <p className="text-red-600 font-bold">
3103:                 {Object.values(nutritionAnalysis).reduce(
3104:                     (sum, cat) => sum + cat.highImpact,
3105:                     0
3106:                 )}
3107:             </p>

```

```

3108:         <p className="text-xs">High Impact Factors</p>
3109:     </div>
3110:     <div>
3111:         <p className="text-yellow-600 font-bold">
3112:             {Object.values(nutritionAnalysis).reduce(
3113:                 (sum, cat) => sum + cat.normalImpact,
3114:                 0
3115:             )}
3116:         </p>
3117:         <p className="text-xs">Normal Impact</p>
3118:     </div>
3119:     <div>
3120:         <p className="text-green-600 font-bold">
3121:             {Object.values(nutritionAnalysis).reduce(
3122:                 (sum, cat) => sum + cat.lowImpact,
3123:                 0
3124:             )}
3125:         </p>
3126:         <p className="text-xs">Low Impact Factors</p>
3127:     </div>
3128: </div>
3129: </div>
3130:
3131: <div className="grid grid-cols-2 gap-3">
3132:     {Object.entries(nutritionAnalysis).map(
3133:         ([section, data]) => (
3134:             <div
3135:                 key={section}
3136:                 className="border rounded p-2 bg-white"
3137:             >
3138:                 <p className="font-semibold text-center uppercase mb-1">
3139:                     {section.replace(/[A-Z]/g, " $1")}
3140:                 </p>
3141:                 <div className="mb-2">
3142:                     <p>
3143:                         Avg Score:{" "}
3144:                         <span className="font-bold text-blue-600">
3145:                             {data.avgScore.toFixed(1)}/10
3146:                         </span>
3147:                     </p>
3148:                     <p>
3149:                         Impact Distribution: {data.highImpact}H / {" "}
3150:                         {data.normalImpact}N / {data.lowImpact}L
3151:                     </p>
3152:                 </div>
3153:
3154:                 {/* Simple Visual Bars for Impact Distribution */}
3155:                 {/* <div className="mb-2">
3156:                     <p className="text-xs font-medium mb-1">
3157:                         Impact Levels:
3158:                     </p>
3159:                     <div className="space-y-1">
3160:                         <div className="flex items-center">
3161:                             <div className="w-12 bg-gray-200 rounded-full h-2 mr-2">
3162:                                 <div
3163:                                     className="bg-red-500 h-2 rounded-full"
3164:                                     style={{
3165:                                         width: `${
3166:                                             data.highImpact > 0
3167:                                                 ? Math.max(
3168:                                                     (data.highImpact /
3169:                                                         Math.max(
3170:                                                             data.highImpact +
3171:                                                             data.normalImpact +
3172:                                                             data.lowImpact,
3173:                                                             1
3174:                                                         )) *
3175:                                                     100,
3176:                                                     10
3177:                                                 )
3178:                                                 : 0
3179:                                         }%`,
3180:                                     }}

```

```

3181:         ></div>
3182:     </div>
3183:     <span className="text-xs text-red-600">
3184:         High ({data.highImpact})
3185:     </span>
3186: </div>
3187: <div className="flex items-center">
3188:     <div className="w-12 bg-gray-200 rounded-full h-2 mr-2">
3189:         <div
3190:             className="bg-yellow-500 h-2 rounded-full"
3191:             style={{
3192:                 width: `${
3193:                     data.normalImpact > 0
3194:                     ? Math.max(
3195:                         (data.normalImpact /
3196:                         Math.max(
3197:                             data.highImpact +
3198:                             data.normalImpact +
3199:                             data.lowImpact,
3200:                             1
3201:                         )) *
3202:                         100,
3203:                         10
3204:                     )
3205:                     : 0
3206:                 }%`,
3207:             }}
3208:         ></div>
3209:     </div>
3210:     <span className="text-xs text-yellow-600">
3211:         Normal ({data.normalImpact})
3212:     </span>
3213: </div>
3214: <div className="flex items-center">
3215:     <div className="w-12 bg-gray-200 rounded-full h-2 mr-2">
3216:         <div
3217:             className="bg-green-500 h-2 rounded-full"
3218:             style={{
3219:                 width: `${
3220:                     data.lowImpact > 0
3221:                     ? Math.max(
3222:                         (data.lowImpact /
3223:                         Math.max(
3224:                             data.highImpact +
3225:                             data.normalImpact +
3226:                             data.lowImpact,
3227:                             1
3228:                         )) *
3229:                         100,
3230:                         10
3231:                     )
3232:                     : 0
3233:                 }%`,
3234:             }}
3235:         ></div>
3236:     </div>
3237:     <span className="text-xs text-green-600">
3238:         Low ({data.lowImpact})
3239:     </span>
3240: </div>
3241: </div>
3242: </div> */}
3243:
3244: {data.items.length > 0 && (
3245:     <div>
3246:         <p className="font-semibold text-xs mb-1">
3247:             Key Nutrients:
3248:         </p>
3249:         <div className="space-y-1">
3250:             {data.items.slice(0, 3).map((item, idx) => (
3251:                 <div
3252:                     key={idx}
3253:                     className="text-xs flex items-center"

```

```

3254:         >
3255:         <div
3256:             className={`w-2 h-2 rounded-full mr-1 ${
3257:                 item.score >= 7
3258:                 ? "bg-green-500"
3259:                 : item.score >= 4
3260:                 ? "bg-yellow-500"
3261:                 : "bg-red-500"
3262:             }`}
3263:         ></div>
3264:         <span className="font-medium">
3265:             {item.name}:
3266:         </span>{ " " }
3267:         {item.impact} ({item.score}/10)
3268:     </div>
3269:   )))
3270: </div>
3271: </div>
3272:   })
3273: </div>
3274: )
3275:   })
3276: </div>
3277: </CardContent>
3278: </Card>
3279: })
3280:
3281: { /* Detailed Lifestyle Analysis */ }
3282: {Object.keys(lifestyleAnalysis).length > 0 && (
3283:   <Card className="border shadow-sm">
3284:     <CardHeader className="pb-2">
3285:       <CardTitle className="text-sm text-center font-bold mb-4 text-yellow-800">
3286:         COMPREHENSIVE LIFESTYLE GENETIC ASSESSMENT
3287:       </CardTitle>
3288:     </CardHeader>
3289:     <CardContent className="text-xs text-gray-700">
3290:       <div className="mb-3 p-2 ">
3291:         <p className="font-semibold mb-3">
3292:           LIFESTYLE BALANCE OVERVIEW
3293:         </p>
3294:         <div className="grid grid-cols-2 gap-2 text-center">
3295:           <div>
3296:             <p className="text-green-600 font-bold text-lg">
3297:               {totalLifestyleStrengths}
3298:             </p>
3299:             <p className="text-xs">
3300:               Genetic Strengths to Leverage
3301:             </p>
3302:           </div>
3303:           <div>
3304:             <p className="text-red-600 font-bold text-lg">
3305:               {totalLifestyleImprovements}
3306:             </p>
3307:             <p className="text-xs">Areas Requiring Attention</p>
3308:           </div>
3309:         </div>
3310:       </div>
3311:
3312:       { /* Simple Overview Chart */ }
3313:       <div className="mb-4 p-2 bg-gradient-to-br from-yellow-50 to-yellow-100 rounded border">
3314:         <p className="font-semibold text-yellow-700 mb-2 text-center">
3315:           YOUR LIFESTYLE BALANCE
3316:         </p>
3317:         <div className="h-48">
3318:           <ResponsiveContainer width="100%" height="100%">
3319:             <BarChart
3320:               data={Object.entries(lifestyleAnalysis).map(
3321:                 ([category, data]) => ({
3322:                   category: category
3323:                     .replace(/([a-z])([A-Z])/g, "$1 $2")
3324:                     .replace(/([A-Z])([A-Z][a-z])/g, "$1 $2")
3325:                     .replace(/&/g, " & ")
3326:                     .replace(/\\s+/g, " ")

```



```

3327:         .trim(),
3328:         strengths: data.strengths,
3329:         improvements: data.improvements,
3330:     })
3331: })
3332: margin={{ top: 10, right: 8, left: 15, bottom: 40 }}
3333: >
3334: <XAxis
3335:   dataKey="category"
3336:   axisLine
3337:   tickLine
3338:   tick={{ fontSize: 9, fill: "#a16207" }}
3339:   stroke="#a16207"
3340:   angle={-45}
3341:   textAnchor="end"
3342:   interval={0}
3343: />
3344: <YAxis
3345:   axisLine
3346:   tickLine
3347:   tick={{ fontSize: 9, fill: "#a16207" }}
3348:   stroke="#a16207"
3349: />
3350: <Tooltip
3351:   contentStyle={{
3352:     backgroundColor: "#fffbeb",
3353:     border: "1px solid #a16207",
3354:     borderRadius: "4px",
3355:     fontSize: "10px",
3356:   }}
3357: />
3358: <Bar
3359:   dataKey="strengths"
3360:   fill="#16a34a"
3361:   name="Strengths"
3362:   radius={[2, 2, 0, 0]}
3363: />
3364: <Bar
3365:   dataKey="improvements"
3366:   fill="#dc2626"
3367:   name="Need Attention"
3368:   radius={[2, 2, 0, 0]}
3369: />
3370: </BarChart>
3371: </ResponsiveContainer>
3372: </div>
3373: <div className="flex justify-center gap-4 mt-1">
3374:   <div className="flex items-center gap-1">
3375:     <div className="w-3 h-3 bg-green-500 rounded"></div>
3376:     <span className="text-xs text-yellow-700">
3377:       Your Strengths
3378:     </span>
3379:   </div>
3380:   <div className="flex items-center gap-1">
3381:     <div className="w-3 h-3 bg-red-500 rounded"></div>
3382:     <span className="text-xs text-yellow-700">
3383:       Need Attention
3384:     </span>
3385:   </div>
3386: </div>
3387: </div>
3388:
3389: <div className="grid grid-cols-2 gap-3">
3390:   {Object.entries(lifestyleAnalysis).map(
3391:     ([category, data]) => (
3392:       <div
3393:         key={category}
3394:         className="border rounded p-2 bg-white"
3395:       >
3396:         <div className="flex items-center gap-2 mb-1">
3397:           {data.hasImages &&
3398:             reportData.lifestyleCategoryImages?.[
3399:               category

```

```

3400:         ] && (
3401:             <img
3402:                 src={
3403:                     reportData.lifestyleCategoryImages[
3404:                         category
3405:                     ] || "/placeholder.svg"
3406:                 }
3407:                 alt={category}
3408:                 className="w-4 h-4 rounded object-cover"
3409:             />
3410:         )}
3411:         <p className="font-semibold capitalize">
3412:             {category.replace(/[A-Z]/g, " $1")}
3413:         </p>
3414:     </div>
3415:
3416:     <div className="mb-2">
3417:         <p>
3418:             Balance: { " " }
3419:             <span className="text-green-600 font-bold">
3420:                 {data.strengths}
3421:             </span> { " " }
3422:             strengths / { " " }
3423:             <span className="text-red-600 font-bold">
3424:                 {data.improvements}
3425:             </span> { " " }
3426:             improvements
3427:         </p>
3428:         <p>Total Conditions: {data.total}</p>
3429:     </div>
3430:
3431:     { /* Simple Balance Visualization */ }
3432:     <div className="mb-2">
3433:         <div className="flex items-center justify-between text-xs mb-1">
3434:             <span className="text-green-600">
3435:                 Strengths
3436:             </span>
3437:             <span className="text-red-600">
3438:                 Need Attention
3439:             </span>
3440:         </div>
3441:         <div className="flex h-3 bg-gray-200 rounded-full overflow-hidden">
3442:             <div
3443:                 className="bg-green-500"
3444:                 style={{
3445:                     width:
3446:                         data.total > 0
3447:                             ? `${
3448:                                 (data.strengths / data.total) * 100
3449:                             }%`
3450:                             : "0%",
3451:                 }}
3452:             ></div>
3453:             <div
3454:                 className="bg-red-500"
3455:                 style={{
3456:                     width:
3457:                         data.total > 0
3458:                             ? `${
3459:                                 (data.improvements / data.total) *
3460:                                 100
3461:                             }%`
3462:                             : "0%",
3463:                 }}
3464:             ></div>
3465:         </div>
3466:         <div className="flex justify-between text-xs mt-1">
3467:             <span className="text-green-600">
3468:                 {data.strengths}
3469:             </span>
3470:             <span className="text-red-600">
3471:                 {data.improvements}
3472:             </span>

```

```

3473:         </div>
3474:     </div>
3475:
3476:     {data.conditions.length > 0 && (
3477:         <div>
3478:             <p className="font-semibold text-xs mb-1">
3479:                 Key Conditions:
3480:             </p>
3481:             <div className="space-y-1">
3482:                 {data.conditions
3483:                     .slice(0, 3)
3484:                     .map((condition, idx) => (
3485:                         <div
3486:                             key={idx}
3487:                             className="text-xs flex items-center gap-1"
3488:                         >
3489:                             {condition.status === "strength" ? (
3490:                                 <Check className="text-green-600 w-2 h-2" />
3491:                             ) : (
3492:                                 <Check className="text-red-600 w-2 h-2" />
3493:                             )}
3494:                             <span className="truncate">
3495:                                 {condition.name}
3496:                             </span>
3497:                         </div>
3498:                     ))}
3499:             </div>
3500:         </div>
3501:     )}
3502: </div>
3503: )
3504: })
3505: </div>
3506: </CardContent>
3507: </Card>
3508: )}
3509:
3510: {/* Detailed Metabolic Analysis */}
3511: {Object.keys(metabolicAnalysis).length > 0 && (
3512:     <Card className="border shadow-sm">
3513:         {/* HEADER SECTION */}
3514:         <CardHeader className="pb-2">
3515:             <CardTitle className="text-sm font-bold text-red-800">
3516:                 COMPREHENSIVE METABOLIC GENETIC PROFILE
3517:             </CardTitle>
3518:         </CardHeader>
3519:
3520:         <CardContent className="text-xs text-gray-700">
3521:             {/* TOP STATISTICS GRID - Shows total counts for each impact category */}
3522:             <div className="mb-3 p-2">
3523:                 <p className="font-semibold mb-1">
3524:                     METABOLIC GENE IMPACT DISTRIBUTION
3525:                 </p>
3526:                 <div className="grid grid-cols-4 gap-2 text-center">
3527:                     {/* High Impact Count */}
3528:                     <div>
3529:                         <p className="text-red-600 font-bold text-lg">
3530:                             {totalMetabolicHigh}
3531:                         </p>
3532:                         <p className="text-xs">High Impact</p>
3533:                     </div>
3534:
3535:                     {/* Moderate Impact Count */}
3536:                     <div>
3537:                         <p className="text-yellow-600 font-bold text-lg">
3538:                             {totalMetabolicModerate}
3539:                         </p>
3540:                         <p className="text-xs">Moderate</p>
3541:                     </div>
3542:
3543:                     {/* Normal Impact Count - NEW */}
3544:                     <div>
3545:                         <p className="text-blue-600 font-bold text-lg">

```

```

3546:         {totalMetabolicNormal}
3547:     </p>
3548:     <p className="text-xs">Normal</p>
3549: </div>
3550:
3551:     { /* Low Impact Count */ }
3552:     <div>
3553:         <p className="text-green-600 font-bold text-lg">
3554:             {totalMetabolicLow}
3555:         </p>
3556:         <p className="text-xs">Low Impact</p>
3557:     </div>
3558: </div>
3559: </div>
3560:
3561: { /* MAIN CHART CONTAINER */ }
3562: <div className="mb-4 p-2 bg-gradient-to-br from-red-50 to-red-100 rounded border">
3563:     <p className="font-semibold text-red-700 mb-2 text-center">
3564:         YOUR METABOLIC GENE IMPACT
3565:     </p>
3566:
3567:     { /* OVERALL IMPACT PROGRESS BAR */ }
3568:     <div className="mb-3">
3569:         { /* Labels above the bar */ }
3570:         <div className="flex justify-between text-xs mb-1">
3571:             <span className="text-red-600">High</span>
3572:             <span className="text-yellow-600">Moderate</span>
3573:             <span className="text-blue-600">Normal</span>
3574:             <span className="text-green-600">Low</span>
3575:         </div>
3576:
3577:         { /* The actual progress bar */ }
3578:         <div className="flex h-6 bg-gray-200 rounded-full overflow-hidden">
3579:             { /* High Impact Section - Red */ }
3580:             <div
3581:                 className="bg-red-500 flex items-center justify-center text-white text-xs font
3582:                 style={{
3583:                     width: `${
3584:                         (totalMetabolicHigh /
3585:                         (totalMetabolicHigh +
3586:                         totalMetabolicModerate +
3587:                         totalMetabolicNormal +
3588:                         totalMetabolicLow)) *
3589:                         100 || 0
3590:                     }%`,
3591:                 }}
3592:             >
3593:                 { /* Only show number if greater than 0 */ }
3594:                 {totalMetabolicHigh > 0 && totalMetabolicHigh}
3595:             </div>
3596:
3597:             { /* Moderate Impact Section - Yellow */ }
3598:             <div
3599:                 className="bg-yellow-500 flex items-center justify-center text-white text-xs f
3600:                 style={{
3601:                     width: `${
3602:                         (totalMetabolicModerate /
3603:                         (totalMetabolicHigh +
3604:                         totalMetabolicModerate +
3605:                         totalMetabolicNormal +
3606:                         totalMetabolicLow)) *
3607:                         100 || 0
3608:                     }%`,
3609:                 }}
3610:             >
3611:                 {totalMetabolicModerate > 0 &&
3612:                 totalMetabolicModerate}
3613:             </div>
3614:
3615:             { /* Normal Impact Section - Blue (NEW) */ }
3616:             <div
3617:                 className="bg-blue-500 flex items-center justify-center text-white text-xs fon
3618:                 style={{

```

```

3619:         width: `${
3620:             (totalMetabolicNormal /
3621:             (totalMetabolicHigh +
3622:             totalMetabolicModerate +
3623:             totalMetabolicNormal +
3624:             totalMetabolicLow)) *
3625:             100 || 0
3626:         }%`,
3627:     }}
3628: >
3629:     {totalMetabolicNormal > 0 && totalMetabolicNormal}
3630: </div>
3631:
3632: { /* Low Impact Section - Green */}
3633: <div
3634:     className="bg-green-500 flex items-center justify-center text-white text-xs font
3635:     style={{
3636:         width: `${
3637:             (totalMetabolicLow /
3638:             (totalMetabolicHigh +
3639:             totalMetabolicModerate +
3640:             totalMetabolicNormal +
3641:             totalMetabolicLow)) *
3642:             100 || 0
3643:         }%`,
3644:     }}
3645: >
3646:     {totalMetabolicLow > 0 && totalMetabolicLow}
3647: </div>
3648: </div>
3649: </div>
3650:
3651: { /* STACKED BAR CHART BY METABOLIC AREA */}
3652: <div className="h-60">
3653:     <ResponsiveContainer width="100%" height="100%">
3654:         <BarChart
3655:             data={Object.entries(metabolicAnalysis).map(
3656:                 ([area, data]) => ({
3657:                     // Format area name for display
3658:                     area: area
3659:                         .replace(/[a-z]([A-Z])/g, "$1 $2")
3660:                         .replace(/[A-Z]([A-Z][a-z])/g, "$1 $2")
3661:                         .replace(/&/g, " & ")
3662:                         .replace(/\\s+/g, " ")
3663:                         .trim(),
3664:                     // Data for each impact level
3665:                     high: data.highImpact,
3666:                     moderate: data.moderateImpact,
3667:                     normal: data.normalImpact, // NEW
3668:                     low: data.lowImpact,
3669:                     total: data.total,
3670:                 })
3671:             )}
3672:             margin={{
3673:                 top: 10,
3674:                 right: 8,
3675:                 left: 15,
3676:                 bottom: 100,
3677:             }}
3678: >
3679:             { /* X-Axis - Metabolic Area Names */}
3680:             <XAxis
3681:                 dataKey="area"
3682:                 axisLine
3683:                 tickLine
3684:                 tick={{ fontSize: 9, fill: "#991b1b" }}
3685:                 stroke="#991b1b"
3686:                 angle={-45}
3687:                 textAnchor="end"
3688:                 interval={0}
3689:             />
3690:
3691:             { /* Y-Axis - Gene Count */}

```

```

3692:         <YAxis
3693:             axisLine
3694:             tickLine
3695:             tick={{ fontSize: 9, fill: "#991b1b" }}
3696:             stroke="#991b1b"
3697:             label={{
3698:                 value: "Gene Count",
3699:                 angle: -90,
3700:                 position: "insideLeft",
3701:             }}
3702:         />
3703:
3704:         { /* Tooltip on hover */ }
3705:         <Tooltip
3706:             contentStyle={{
3707:                 backgroundColor: "#fef2f2",
3708:                 border: "1px solid #991b1b",
3709:                 borderRadius: "4px",
3710:                 fontSize: "10px",
3711:             }}
3712:         />
3713:
3714:         { /* Stacked Bars - Each impact level */ }
3715:         { /* High Impact Bar - Red */ }
3716:         <Bar
3717:             dataKey="high"
3718:             stackId="impact"
3719:             fill="#dc2626"
3720:             name="High Impact"
3721:             radius={[0, 0, 0, 0]}
3722:         />
3723:
3724:         { /* Moderate Impact Bar - Yellow */ }
3725:         <Bar
3726:             dataKey="moderate"
3727:             stackId="impact"
3728:             fill="#eab308"
3729:             name="Moderate Impact"
3730:             radius={[0, 0, 0, 0]}
3731:         />
3732:
3733:         { /* Normal Impact Bar - Blue (NEW) */ }
3734:         <Bar
3735:             dataKey="normal"
3736:             stackId="impact"
3737:             fill="#3b82f6"
3738:             name="Normal"
3739:             radius={[0, 0, 0, 0]}
3740:         />
3741:
3742:         { /* Low Impact Bar - Green */ }
3743:         <Bar
3744:             dataKey="low"
3745:             stackId="impact"
3746:             fill="#16a34a"
3747:             name="Low Impact"
3748:             radius={[2, 2, 0, 0]} // Rounded top corners
3749:         />
3750:     </BarChart>
3751: </ResponsiveContainer>
3752: </div>
3753:
3754: { /* CHART LEGEND */ }
3755: <div className="flex justify-center gap-3 mt-1">
3756:     { /* High Impact Legend */ }
3757:     <div className="flex items-center gap-1">
3758:         <div className="w-3 h-3 bg-red-500 rounded"></div>
3759:         <span className="text-xs text-red-700">
3760:             High Impact
3761:         </span>
3762:     </div>
3763:
3764:     { /* Moderate Impact Legend */ }

```

```

3765:         <div className="flex items-center gap-1">
3766:             <div className="w-3 h-3 bg-yellow-500 rounded"></div>
3767:             <span className="text-xs text-red-700">Moderate</span>
3768:         </div>
3769:
3770:         { /* Normal Impact Legend - NEW */ }
3771:         <div className="flex items-center gap-1">
3772:             <div className="w-3 h-3 bg-blue-500 rounded"></div>
3773:             <span className="text-xs text-red-700">Normal</span>
3774:         </div>
3775:
3776:         { /* Low Impact Legend */ }
3777:         <div className="flex items-center gap-1">
3778:             <div className="w-3 h-3 bg-green-500 rounded"></div>
3779:             <span className="text-xs text-red-700">
3780:                 Low Impact
3781:             </span>
3782:         </div>
3783:     </div>
3784: </div>
3785:
3786: { /* DETAILED AREA BREAKDOWN CARDS */ }
3787: <div className="grid grid-cols-2 gap-3">
3788:     {Object.entries(metabolicAnalysis).map(([area, data]) => (
3789:         <div key={area} className="border rounded p-2 bg-white">
3790:             { /* Area Name */ }
3791:             <p className="font-semibold capitalize mb-1">
3792:                 {area
3793:                     .replace(/[a-z]([A-Z])/g, "$1 $2")
3794:                     .replace(/[A-Z]([A-Z][a-z])/g, "$1 $2")
3795:                     .replace(/&/g, " & ")
3796:                     .replace(/\\s+/g, " ")
3797:                     .trim()}
3798:             </p>
3799:
3800:             { /* Area Statistics */ }
3801:             <div className="mb-2">
3802:                 <p>Total Genes: {data.total}</p>
3803:                 <p>
3804:                     Impact: { " " }
3805:                     <span className="text-red-600">
3806:                         {data.highImpact}H
3807:                     </span>{ " " }
3808:                     /{ " " }
3809:                     <span className="text-yellow-600">
3810:                         {data.moderateImpact}M
3811:                     </span>{ " " }
3812:                     /{ " " }
3813:                     <span className="text-blue-600">
3814:                         {data.normalImpact}N
3815:                     </span>{ " " }
3816:                     /{ " " }
3817:                     <span className="text-green-600">
3818:                         {data.lowImpact}L
3819:                     </span>
3820:                 </p>
3821:             </div>
3822:
3823:             { /* Area-Specific Progress Bar */ }
3824:             <div className="mb-2">
3825:                 <div className="flex h-4 bg-gray-200 rounded-full overflow-hidden">
3826:                     { /* High Impact Section */ }
3827:                     <div
3828:                         className="bg-red-500"
3829:                         style={{
3830:                             width:
3831:                                 data.total > 0
3832:                                 ? `${
3833:                                     (data.highImpact / data.total) * 100
3834:                                 }%`
3835:                                 : "0%",
3836:                         }}
3837:                     ></div>

```



```

3911:         gene.impact
3912:         ?.toLowerCase()
3913:         .includes("high")
3914:         ? "bg-red-500"
3915:         : gene.impact
3916:         ?.toLowerCase()
3917:         .includes("moderate") ||
3918:         gene.impact
3919:         ?.toLowerCase()
3920:         .includes("average")
3921:         ? "bg-yellow-500"
3922:         : gene.impact
3923:         ?.toLowerCase()
3924:         .includes("normal")
3925:         ? "bg-blue-500" // NEW
3926:         : "bg-green-500" // Low impact
3927:     }`
3928: ></span>
3929: { /* Gene name and genotype */}
3930: <span className="font-medium">
3931:   {gene.name}:
3932: </span>{" "}
3933: {gene.genotype}
3934: </div>
3935: </div>
3936:   )}
3937: </div>
3938: </div>
3939:   )}
3940: </div>
3941:   )}
3942: </div>
3943: </CardContent>
3944: </Card>
3945: )}
3946:
3947: { /* Sports & Fitness Analysis (only if data exists) */}
3948: {Object.keys(sportsAnalysis).length > 0 && (
3949:   <Card className="border shadow-sm">
3950:     <CardHeader className="pb-2">
3951:       <CardTitle className="text-sm font-bold text-orange-800">
3952:         DETAILED FITNESS GENETIC POTENTIAL
3953:       </CardTitle>
3954:     </CardHeader>
3955:     <CardContent className="text-xs text-gray-700">
3956:       <div className="mb-3 p-2 bg-white rounded border">
3957:         <p className="font-semibold mb-1">
3958:           FITNESS GENETIC OVERVIEW
3959:         </p>
3960:         <p>
3961:           Total Traits Analyzed:{" "}
3962:           {Object.values(sportsAnalysis).reduce(
3963:             (sum, section) => sum + section.totalTraits,
3964:             0
3965:           )}{ " "}
3966:           across {Object.keys(sportsAnalysis).length} categories
3967:         </p>
3968:       </div>
3969:
3970:       <div className="grid grid-cols-2 gap-3">
3971:         {Object.entries(sportsAnalysis).map(([section, data]) => (
3972:           <div
3973:             key={section}
3974:             className="border rounded p-2 bg-white"
3975:           >
3976:             <p className="font-semibold capitalize mb-1">
3977:               {section.replace(/[A-Z]/g, " $1")}
3978:             </p>
3979:             <p className="mb-2">Traits: {data.totalTraits}</p>
3980:
3981:             {data.traits.length > 0 && (
3982:               <div>
3983:                 <p className="font-semibold text-xs mb-1">

```

```

3984:         Key Traits:
3985:     </p>
3986:     <div className="space-y-1">
3987:         {data.traits.slice(0, 3).map((trait, idx) => (
3988:             <div
3989:                 key={idx}
3990:                 className="text-xs flex items-center gap-2"
3991:             >
3992:                 {trait.hasImage && (
3993:                     <img
3994:                         src={
3995:                             reportData.sportsAndFitness
3996:                                 .customImages?.[trait.imageKey] ||
3997:                             defaultImageMap[trait.imageKey] ||
3998:                             "/sports/default.png"
3999:                         }
4000:                         alt={trait.imageKey}
4001:                         className="w-4 h-4 object-contain"
4002:                     />
4003:                 )}
4004:                 <div>
4005:                     <span className="font-medium">
4006:                         {trait.label}:
4007:                     </span>{" "}
4008:                     {trait.level
4009:                         ?.split("-")
4010:                         .pop()
4011:                         ?.toUpperCase()}
4012:                 </div>
4013:             </div>
4014:         )})
4015:     </div>
4016: </div>
4017: )}
4018: </div>
4019: )})
4020: </div>
4021: </CardContent>
4022: </Card>
4023: )}
4024:
4025: {/* Digestive Health Analysis (only if data exists) */}
4026: {Object.keys(digestiveAnalysis).length > 0 && (
4027:     <Card className="border shadow-sm">
4028:         <CardHeader className="pb-2">
4029:             <CardTitle className="text-sm font-bold text-pink-800">
4030:                 DIGESTIVE HEALTH GENETIC ANALYSIS
4031:             </CardTitle>
4032:         </CardHeader>
4033:         <CardContent className="text-xs text-gray-700">
4034:             <div className="mb-3 p-2">
4035:                 <p className="font-semibold mb-1">
4036:                     DIGESTIVE SENSITIVITY OVERVIEW
4037:                 </p>
4038:                 <p>
4039:                     Total Parameters:{" "}
4040:                     {Object.keys(digestiveAnalysis).length}
4041:                 </p>
4042:                 <div className="grid grid-cols-2 gap-2 text-center mt-2">
4043:                     <div>
4044:                         <p className="text-red-600 font-bold">
4045:                             {
4046:                                 Object.values(digestiveAnalysis).filter(
4047:                                     (item) => item.sensitivity === "High"
4048:                                 ).length
4049:                             }
4050:                         </p>
4051:                         <p className="text-xs">High Sensitivity</p>
4052:                     </div>
4053:                     <div>
4054:                         <p className="text-green-600 font-bold">
4055:                             {
4056:                                 Object.values(digestiveAnalysis).filter(

```

```

4057:             (item) => item.sensitivity === "Low"
4058:         ).length
4059:     }
4060: </p>
4061:     <p className="text-xs">Low Sensitivity</p>
4062: </div>
4063: </div>
4064: </div>
4065:
4066: { /* Simple Overview Chart */ }
4067: <div className="mb-4 p-2 bg-gradient-to-br from-pink-50 to-pink-100 rounded border">
4068:   <p className="font-semibold text-pink-700 mb-2 text-center">
4069:     YOUR DIGESTIVE SENSITIVITY PROFILE
4070:   </p>
4071:
4072:   { /* Overall Sensitivity Bar */ }
4073:   <div className="mb-3">
4074:     <div className="flex justify-between text-xs mb-1">
4075:       <span className="text-red-600">High Sensitivity</span>
4076:       <span className="text-green-600">
4077:         Low Sensitivity
4078:       </span>
4079:     </div>
4080:     <div className="flex h-6 bg-gray-200 rounded-full overflow-hidden">
4081:       <div
4082:         className="bg-red-500 flex items-center justify-center text-white text-xs font
4083:         style={{
4084:           width: `${
4085:             (Object.values(digestiveAnalysis).filter(
4086:               (item) => item.sensitivity === "High"
4087:             ).length /
4088:             Object.keys(digestiveAnalysis).length) *
4089:             100 || 0
4090:           }%`,
4091:         }}
4092:       >
4093:         {Object.values(digestiveAnalysis).filter(
4094:           (item) => item.sensitivity === "High"
4095:         ).length > 0 &&
4096:           Object.values(digestiveAnalysis).filter(
4097:             (item) => item.sensitivity === "High"
4098:           ).length}
4099:       </div>
4100:       <div
4101:         className="bg-green-500 flex items-center justify-center text-white text-xs font
4102:         style={{
4103:           width: `${
4104:             (Object.values(digestiveAnalysis).filter(
4105:               (item) => item.sensitivity === "Low"
4106:             ).length /
4107:             Object.keys(digestiveAnalysis).length) *
4108:             100 || 0
4109:           }%`,
4110:         }}
4111:       >
4112:         {Object.values(digestiveAnalysis).filter(
4113:           (item) => item.sensitivity === "Low"
4114:         ).length > 0 &&
4115:           Object.values(digestiveAnalysis).filter(
4116:             (item) => item.sensitivity === "Low"
4117:           ).length}
4118:       </div>
4119:     </div>
4120:     <p className="text-xs text-center mt-1 text-pink-600">
4121:       {Object.values(digestiveAnalysis).filter(
4122:         (item) => item.sensitivity === "High"
4123:       ).length >
4124:       Object.values(digestiveAnalysis).filter(
4125:         (item) => item.sensitivity === "Low"
4126:       ).length
4127:       ? "You may need to be more careful with certain foods"
4128:       : "You have good genetic tolerance for most foods"}
4129:     </p>

```

```

4130:         </div>
4131:
4132:         { /* Simple Visual List */ }
4133:         <div className="h-full overflow-y-auto">
4134:             <div className="space-y-2">
4135:                 {Object.entries(digestiveAnalysis)
4136:                     .sort(([, a], [, b]) =>
4137:                         a.sensitivity === "High" ? -1 : 1
4138:                     )
4139:                     .map(([key, item]) => (
4140:                         <div
4141:                             key={key}
4142:                             className="flex items-center justify-between bg-white rounded p-2 border"
4143:                         >
4144:                             <div className="flex items-center gap-2">
4145:                                 {item.icon && (
4146:                                     <img
4147:                                         src={`~/digestive/${item.icon}.png`}
4148:                                         alt={item.title}
4149:                                         className="w-3 h-3 object-contain"
4150:                                     />
4151:                                 )}
4152:                                 <span className="text-xs font-medium">
4153:                                     {item.title}
4154:                                 </span>
4155:                             </div>
4156:                             <div className="flex items-center gap-2">
4157:                                 <div
4158:                                     className={`w-3 h-3 rounded-full ${
4159:                                         item.sensitivity === "High"
4160:                                             ? "bg-red-500"
4161:                                             : "bg-green-500"
4162:                                     }`}
4163:                                 ></div>
4164:                                 <span
4165:                                     className={`text-xs font-bold ${
4166:                                         item.sensitivity === "High"
4167:                                             ? "text-red-600"
4168:                                             : "text-green-600"
4169:                                     }`}
4170:                                 >
4171:                                     {item.sensitivity}
4172:                                 </span>
4173:                             </div>
4174:                         </div>
4175:                     )})}
4176:             </div>
4177:         </div>
4178:
4179:         <div className="flex justify-center gap-4 mt-2">
4180:             <div className="flex items-center gap-1">
4181:                 <div className="w-3 h-3 bg-red-500 rounded-full"></div>
4182:                 <span className="text-xs text-pink-700">
4183:                     High Sensitivity
4184:                 </span>
4185:             </div>
4186:             <div className="flex items-center gap-1">
4187:                 <div className="w-3 h-3 bg-green-500 rounded-full"></div>
4188:                 <span className="text-xs text-pink-700">
4189:                     Low Sensitivity
4190:                 </span>
4191:             </div>
4192:         </div>
4193:     </div>
4194:
4195:     <div className="grid grid-cols-2 gap-3 h-full">
4196:         {Object.entries(digestiveAnalysis).map(([key, item]) => (
4197:             <div
4198:                 key={key}
4199:                 className="border rounded p-2 h-full bg-white"
4200:             >
4201:                 <div className="flex items-center gap-2 mb-1">
4202:                     {item.icon && (

```

```

4203:         <img
4204:             src={`digestive/${item.icon}.png`}
4205:             alt={item.title}
4206:             className="w-4 h-4 object-contain"
4207:         />
4208:     )}
4209:     <p className="font-semibold">{item.title}</p>
4210: </div>
4211:
4212:     { /* Sensitivity Indicator Bar */ }
4213:     <div className="mb-2">
4214:         <div className="flex justify-between items-center mb-1">
4215:             <span className="text-xs">
4216:                 Sensitivity Level:
4217:             </span>
4218:             <span
4219:                 className={`text-xs font-bold ${
4220:                     item.sensitivity === "High"
4221:                     ? "text-red-600"
4222:                     : "text-green-600"
4223:                 }`}
4224:             >
4225:                 {item.sensitivity}
4226:             </span>
4227:         </div>
4228:         <div className="h-3 bg-gray-200 rounded-full overflow-hidden">
4229:             <div
4230:                 className={`h-full ${
4231:                     item.sensitivity === "High"
4232:                     ? "bg-red-500"
4233:                     : "bg-green-500"
4234:                 }`}
4235:                 style={{
4236:                     width:
4237:                         item.sensitivity === "High" ? "85%" : "25%",
4238:                 }}
4239:             ></div>
4240:         </div>
4241:     </div>
4242:
4243:     {item.sensitivity === "High" && item.bad && (
4244:         <p className="text-xs text-red-700">
4245:             {item.bad.substring(0, 5000)}...
4246:         </p>
4247:     )}
4248:     {item.sensitivity === "Low" && item.good && (
4249:         <p className="text-xs text-green-700">
4250:             {item.good.substring(0, 5000)}...
4251:         </p>
4252:     )}
4253:     </div>
4254:   )}
4255: </div>
4256: </CardContent>
4257: </Card>
4258: )}
4259:
4260: { /* Addiction Analysis (only if data exists) */ }
4261: {Object.keys(addictionAnalysis).length > 0 && (
4262:     <Card className="border shadow-sm">
4263:         <CardHeader className="pb-2">
4264:             <CardTitle className="text-sm font-bold text-indigo-800">
4265:                 ADDICTION SUSCEPTIBILITY GENETIC PROFILE
4266:             </CardTitle>
4267:         </CardHeader>
4268:         <CardContent className="text-xs text-gray-700">
4269:             <div className="mb-3 p-2 bg-white rounded ">
4270:                 <p className="font-semibold mb-1">
4271:                     ADDICTION GENETIC FACTORS
4272:                 </p>
4273:                 <p>
4274:                     Total Substances Analyzed:{ " "}
4275:                     {Object.keys(addictionAnalysis).length}

```

```

4276:         </p>
4277:     </div>
4278:
4279:     { /* Simple Risk Overview */}
4280:     <div className="mb-4 p-2 bg-gradient-to-br from-indigo-50 to-indigo-100 rounded border
4281:     <p className="font-semibold text-indigo-700 mb-2 text-center">
4282:         YOUR GENETIC SUSCEPTIBILITY OVERVIEW
4283:     </p>
4284:
4285:     { /* Risk Level Summary */}
4286:     <div className="mb-3 p-2 bg-white rounded border">
4287:         <div className="grid grid-cols-3 gap-2 text-center">
4288:             <div>
4289:                 <p className="text-red-600 font-bold text-lg">
4290:                     {
4291:                         Object.values(addictionAnalysis).filter(
4292:                             (item) =>
4293:                                 item.sensitivityIcon?.includes("high") ||
4294:                                 item.title?.toLowerCase().includes("high")
4295:                             ).length
4296:                     }
4297:                 </p>
4298:                 <p className="text-xs">Higher Risk</p>
4299:             </div>
4300:             <div>
4301:                 <p className="text-yellow-600 font-bold text-lg">
4302:                     {
4303:                         Object.values(addictionAnalysis).filter(
4304:                             (item) =>
4305:                                 item.sensitivityIcon?.includes("medium") ||
4306:                                 item.sensitivityIcon?.includes(
4307:                                     "moderate"
4308:                                 ) ||
4309:                                 item.title
4310:                                     ?.toLowerCase()
4311:                                     .includes("medium") ||
4312:                                 item.title
4313:                                     ?.toLowerCase()
4314:                                     .includes("moderate")
4315:                             ).length
4316:                     }
4317:                 </p>
4318:                 <p className="text-xs">Moderate Risk</p>
4319:             </div>
4320:             <div>
4321:                 <p className="text-green-600 font-bold text-lg">
4322:                     {
4323:                         Object.values(addictionAnalysis).filter(
4324:                             (item) =>
4325:                                 item.sensitivityIcon?.includes("low") ||
4326:                                 item.title?.toLowerCase().includes("low")
4327:                             ).length
4328:                     }
4329:                 </p>
4330:                 <p className="text-xs">Lower Risk</p>
4331:             </div>
4332:         </div>
4333:     </div>
4334:
4335:     { /* Awareness Message */}
4336:     <div className="bg-blue-50 border border-blue-200 rounded p-2 mb-3">
4337:         <p className="text-xs text-blue-800 text-center">
4338:             <strong>Important:</strong> Genetics is just one
4339:             factor. Environment, lifestyle, and personal choices
4340:             play major roles in addiction risk.
4341:         </p>
4342:     </div>
4343:
4344:     { /* Simple Visual List with Risk Levels */}
4345:     <div className="h-48 overflow-y-auto">
4346:         <div className="space-y-2">
4347:             {Object.entries(addictionAnalysis)
4348:                 .sort(([, a], [, b]) => {

```

```

4349: // Sort by risk level (high risk first)
4350: const getRiskLevel = (item) => {
4351:   if (
4352:     item.sensitivityIcon?.includes("high") ||
4353:     item.title?.toLowerCase().includes("high")
4354:   )
4355:     return 3;
4356:   if (
4357:     item.sensitivityIcon?.includes("medium") ||
4358:     item.sensitivityIcon?.includes("moderate") ||
4359:     item.title
4360:       ?.toLowerCase()
4361:       .includes("medium") ||
4362:     item.title?.toLowerCase().includes("moderate")
4363:   )
4364:     return 2;
4365:   return 1;
4366: };
4367: return getRiskLevel(b) - getRiskLevel(a);
4368: })
4369: .map(([key, item]) => {
4370:   const isHighRisk =
4371:     item.sensitivityIcon?.includes("high") ||
4372:     item.title?.toLowerCase().includes("high");
4373:   const isMediumRisk =
4374:     item.sensitivityIcon?.includes("medium") ||
4375:     item.sensitivityIcon?.includes("moderate") ||
4376:     item.title?.toLowerCase().includes("medium") ||
4377:     item.title?.toLowerCase().includes("moderate");
4378:   const riskLevel = isHighRisk
4379:     ? "Higher"
4380:     : isMediumRisk
4381:     ? "Moderate"
4382:     : "Lower";
4383:   const riskColor = isHighRisk
4384:     ? "red"
4385:     : isMediumRisk
4386:     ? "yellow"
4387:     : "green";
4388:
4389:   return (
4390:     <div
4391:       key={key}
4392:       className="flex items-center justify-between bg-white rounded p-2 border
4393:     >
4394:     <div className="flex items-center gap-2">
4395:       {item.icon && (
4396:         <img
4397:           src={`~/addiction/icons/${item.icon}.png`}
4398:           alt={item.title}
4399:           className="w-4 h-4 object-contain"
4400:         />
4401:       )}
4402:       <span className="text-xs font-medium">
4403:         {item.title}
4404:       </span>
4405:     </div>
4406:     <div className="flex items-center gap-2">
4407:       <div className="flex items-center gap-1">
4408:         <div
4409:           className={`w-3 h-3 rounded-full ${
4410:             riskColor === "red"
4411:               ? "bg-red-500"
4412:               : riskColor === "yellow"
4413:               ? "bg-yellow-500"
4414:               : "bg-green-500"
4415:           }`}
4416:         ></div>
4417:         <span
4418:           className={`text-xs font-bold ${
4419:             riskColor === "red"
4420:               ? "text-red-600"
4421:               : riskColor === "yellow"

```

```

4422:         ? "text-yellow-600"
4423:         : "text-green-600"
4424:     }` }
4425:     >
4426:     {riskLevel}
4427:   </span>
4428: </div>
4429: </div>
4430: </div>
4431:   );
4432:   }}
4433: </div>
4434: </div>
4435:
4436: <div className="flex justify-center gap-4 mt-2">
4437:   <div className="flex items-center gap-1">
4438:     <div className="w-3 h-3 bg-red-500 rounded-full"></div>
4439:     <span className="text-xs text-indigo-700">
4440:       Higher Risk
4441:     </span>
4442:   </div>
4443:   <div className="flex items-center gap-1">
4444:     <div className="w-3 h-3 bg-yellow-500 rounded-full"></div>
4445:     <span className="text-xs text-indigo-700">
4446:       Moderate Risk
4447:     </span>
4448:   </div>
4449:   <div className="flex items-center gap-1">
4450:     <div className="w-3 h-3 bg-green-500 rounded-full"></div>
4451:     <span className="text-xs text-indigo-700">
4452:       Lower Risk
4453:     </span>
4454:   </div>
4455: </div>
4456: </div>
4457:
4458: <div className="grid grid-cols-2 gap-3">
4459:   {Object.entries(addictionAnalysis).map(([key, item]) => {
4460:     const isHighRisk =
4461:       item.sensitivityIcon?.includes("high") ||
4462:       item.title?.toLowerCase().includes("high");
4463:     const isMediumRisk =
4464:       item.sensitivityIcon?.includes("medium") ||
4465:       item.sensitivityIcon?.includes("moderate") ||
4466:       item.title?.toLowerCase().includes("medium") ||
4467:       item.title?.toLowerCase().includes("moderate");
4468:     const riskLevel = isHighRisk
4469:       ? "Higher Risk"
4470:       : isMediumRisk
4471:       ? "Moderate Risk"
4472:       : "Lower Risk";
4473:     const riskColor = isHighRisk
4474:       ? "red"
4475:       : isMediumRisk
4476:       ? "yellow"
4477:       : "green";
4478:
4479:     return (
4480:       <div
4481:         key={key}
4482:         className="border rounded p-2 bg-white"
4483:       >
4484:         <div className="flex items-center gap-2 mb-2">
4485:           {item.icon && (
4486:             <img
4487:               src={`../addiction/icons/${item.icon}.png`}
4488:               alt={item.title}
4489:               className="w-6 h-6 object-contain"
4490:             />
4491:           )}
4492:           <p className="font-semibold">{item.title}</p>
4493:         </div>
4494:

```



```

4495:         { /* Risk Level Indicator */}
4496:         <div className="mb-2">
4497:             <div className="flex justify-between items-center mb-1">
4498:                 <span className="text-xs">Genetic Risk:</span>
4499:                 <span
4500:                     className={`text-xs font-bold ${
4501:                         riskColor === "red"
4502:                         ? "text-red-600"
4503:                         : riskColor === "yellow"
4504:                         ? "text-yellow-600"
4505:                         : "text-green-600"
4506:                     }`}
4507:                 >
4508:                     {riskLevel}
4509:                 </span>
4510:             </div>
4511:             <div className="h-3 bg-gray-200 rounded-full overflow-hidden">
4512:                 <div
4513:                     className={`h-full ${
4514:                         riskColor === "red"
4515:                         ? "bg-red-500"
4516:                         : riskColor === "yellow"
4517:                         ? "bg-yellow-500"
4518:                         : "bg-green-500"
4519:                     }`}
4520:                     style={{
4521:                         width: isHighRisk
4522:                         ? "80%"
4523:                         : isMediumRisk
4524:                         ? "55%"
4525:                         : "30%",
4526:                     }}
4527:                 ></div>
4528:             </div>
4529:         </div>
4530:
4531:         <div className="flex justify-center">
4532:             {item.sensitivityIcon && (
4533:                 <img
4534:                     src={`\addiction/sensitivity/${item.sensitivityIcon}.png`}
4535:                     alt="Addiction Sensitivity"
4536:                     className="w-16 h-auto object-contain"
4537:                 />
4538:             )}
4539:         </div>
4540:
4541:         { /* Helpful Context */}
4542:         <div className="mt-2 p-1 bg-gray-50 rounded">
4543:             <p className="text-xs text-gray-600 text-center">
4544:                 {isHighRisk
4545:                     ? "Extra awareness recommended"
4546:                     : isMediumRisk
4547:                     ? "Moderate caution advised"
4548:                     : "Lower genetic predisposition"}
4549:             </p>
4550:         </div>
4551:     </div>
4552: );
4553: }}
4554: </div>
4555: </CardContent>
4556: </Card>
4557: )}
4558:
4559: { /* Sleep Analysis (only if data exists) */}
4560: {Object.keys(sleepAnalysis).length > 0 && (
4561:     <Card className="border shadow-sm">
4562:         <CardHeader className="pb-2">
4563:             <CardTitle className="text-sm font-bold text-teal-800">
4564:                 SLEEP & REST GENETIC ANALYSIS
4565:             </CardTitle>
4566:         </CardHeader>
4567:         <CardContent className="text-xs text-gray-700">

```

```

4568:         <div className="mb-3 p-2 bg-white rounded border">
4569:             <p className="font-semibold mb-1">
4570:                 SLEEP GENETIC PARAMETERS
4571:             </p>
4572:             <p>
4573:                 Total Sleep Factors: {Object.keys(sleepAnalysis).length}
4574:             </p>
4575:         </div>
4576:
4577:         <div className="grid grid-cols-2 gap-3">
4578:             {Object.entries(sleepAnalysis).map(([key, item]) => (
4579:                 <div key={key} className="border rounded p-2 bg-white">
4580:                     <div className="flex items-center gap-2 mb-1">
4581:                         {item.hasImage && (
4582:                             <img
4583:                                 src={`\sleep/${item.image}.png`}
4584:                                 alt={item.title}
4585:                                 className="w-6 h-6 object-contain"
4586:                             />
4587:                         )}
4588:                         <p className="font-semibold">{item.title}</p>
4589:                     </div>
4590:                     {item.intervention && (
4591:                         <p className="text-xs">
4592:                             {item.intervention.substring(0, 80)}...
4593:                         </p>
4594:                     )}
4595:                 </div>
4596:             )})}
4597:         </div>
4598:     </CardContent>
4599: </Card>
4600: )}
4601:
4602: {/* Allergy Analysis (only if data exists) */}
4603: {Object.keys(allergyAnalysis).length > 0 && (
4604:     <Card className="border shadow-sm">
4605:         <CardHeader className="pb-2">
4606:             <CardTitle className="text-sm font-bold text-amber-800">
4607:                 ALLERGY & SENSITIVITY GENETIC PROFILE
4608:             </CardTitle>
4609:         </CardHeader>
4610:         <CardContent className="text-xs text-gray-700">
4611:             <div className="mb-3 p-2 bg-white rounded border">
4612:                 <p className="font-semibold mb-1">
4613:                     ALLERGY GENETIC FACTORS
4614:                 </p>
4615:                 <p>
4616:                     Total Allergies Analyzed:{" "}
4617:                     {Object.keys(allergyAnalysis).length}
4618:                 </p>
4619:             </div>
4620:
4621:             <div className="grid grid-cols-2 gap-3">
4622:                 {Object.entries(allergyAnalysis).map(([key, item]) => (
4623:                     <div key={key} className="border rounded p-2 bg-white">
4624:                         <div className="flex items-center gap-2 mb-1">
4625:                             {item.hasImage && (
4626:                                 <img
4627:                                     src={`\allergies/${item.image}.png`}
4628:                                     alt={item.title}
4629:                                     className="w-6 h-6 object-contain"
4630:                                 />
4631:                             )}
4632:                             <p className="font-semibold">{item.title}</p>
4633:                         </div>
4634:                     </div>
4635:                 )})}
4636:             </div>
4637:         </CardContent>
4638:     </Card>
4639: )}
4640:

```

```

4641:      { /* Comprehensive Recommendations */ }
4642:      <Card className="border-2 shadow-sm">
4643:        <CardHeader className="pb-2">
4644:          <CardTitle className="text-sm font-bold text-indigo-800">
4645:            COMPREHENSIVE GENETIC RECOMMENDATIONS
4646:          </CardTitle>
4647:        </CardHeader>
4648:        <CardContent className="text-xs text-gray-700">
4649:          <div className="grid grid-cols-2 gap-3">
4650:            <div className="space-y-2">
4651:              <p className="font-semibold text-red-700">
4652:                IMMEDIATE PRIORITY ACTIONS
4653:              </p>
4654:              <ul className="list-disc list-inside space-y-1 text-xs">
4655:                {totalLifestyleImprovements > 0 && (
4656:                  <li>
4657:                    Address {totalLifestyleImprovements} lifestyle
4658:                    genetic improvement areas immediately
4659:                  </li>
4660:                )}
4661:                {totalMetabolicHigh > 0 && (
4662:                  <li>
4663:                    Closely monitor and manage {totalMetabolicHigh}{" "}
4664:                    high-impact metabolic genetic factors
4665:                  </li>
4666:                )}
4667:                {totalDietScore < 5 && (
4668:                  <li>
4669:                    Implement comprehensive dietary modifications based
4670:                    on genetic profile
4671:                  </li>
4672:                )}
4673:                {Object.values(dietAnalysis).some(
4674:                  (cat) => cat.lowScores > cat.highScores
4675:                ) && (
4676:                  <li>
4677:                    Focus intensive effort on underperforming dietary
4678:                    genetic categories
4679:                  </li>
4680:                )}
4681:                {Object.values(nutritionAnalysis).reduce(
4682:                  (sum, cat) => sum + cat.highImpact,
4683:                  0
4684:                ) > 0 && (
4685:                  <li>
4686:                    Address{" "}
4687:                    {Object.values(nutritionAnalysis).reduce(
4688:                      (sum, cat) => sum + cat.highImpact,
4689:                      0
4690:                    )}{" "}
4691:                    high-impact nutritional genetic factors
4692:                  </li>
4693:                )}
4694:              </ul>
4695:            </div>
4696:
4697:            <div className="space-y-2">
4698:              <p className="font-semibold text-green-700">
4699:                GENETIC STRENGTHS TO LEVERAGE
4700:              </p>
4701:              <ul className="list-disc list-inside space-y-1 text-xs">
4702:                {totalLifestyleStrengths > 0 && (
4703:                  <li>
4704:                    Maximize and build upon {totalLifestyleStrengths}{" "}
4705:                    genetic lifestyle strengths
4706:                  </li>
4707:                )}
4708:                {totalMetabolicLow > 0 && (
4709:                  <li>
4710:                    Maintain current approach for {totalMetabolicLow}{" "}
4711:                    favorable metabolic genetic traits
4712:                  </li>
4713:                )}

```

```

4714:         {totalDietScore >= 6 && (
4715:             <li>
4716:                 Continue and optimize current dietary approach -
4717:                 excellent genetic compatibility
4718:             </li>
4719:         )}
4720:         {Object.values(dietAnalysis).some(
4721:             (cat) => cat.highScores > cat.lowScores
4722:         ) && (
4723:             <li>
4724:                 Further optimize high-performing dietary genetic
4725:                 categories for maximum benefit
4726:             </li>
4727:         )}
4728:         {Object.values(nutritionAnalysis).reduce(
4729:             (sum, cat) => sum + cat.lowImpact,
4730:             0
4731:         ) > 0 && (
4732:             <li>
4733:                 Leverage{" "}
4734:                 {Object.values(nutritionAnalysis).reduce(
4735:                     (sum, cat) => sum + cat.lowImpact,
4736:                     0
4737:                 )}{"}
4738:                 low-impact nutritional factors as genetic advantages
4739:             </li>
4740:         )}
4741:     </ul>
4742: </div>
4743: </div>
4744:
4745: <div className="mt-3 p-3 bg-white rounded border">
4746:     <p className="font-semibold mb-3">
4747:         GENETIC HEALTH SCORE BREAKDOWN
4748:     </p>
4749:
4750:     {/* Chart Container */}
4751:     <div className="h-64 mb-4">
4752:         <svg viewBox="0 0 600 240" className="w-full h-full">
4753:             {/* Grid lines */}
4754:             <defs>
4755:                 <pattern
4756:                     id="grid"
4757:                     width="60"
4758:                     height="24"
4759:                     patternUnits="userSpaceOnUse"
4760:                 >
4761:                     <path
4762:                         d="M 60 0 L 0 0 0 24"
4763:                         fill="none"
4764:                         stroke="#e5e7eb"
4765:                         strokeWidth="0.5"
4766:                     />
4767:                 </pattern>
4768:             </defs>
4769:             <rect width="600" height="240" fill="url(#grid)" />
4770:
4771:             {/* Y-axis labels */}
4772:             <text
4773:                 x="20"
4774:                 y="20"
4775:                 fontSize="10"
4776:                 fill="#6b7280"
4777:                 textAnchor="middle"
4778:             >
4779:                 10
4780:             </text>
4781:             <text
4782:                 x="20"
4783:                 y="44"
4784:                 fontSize="10"
4785:                 fill="#6b7280"
4786:                 textAnchor="middle"

```

```

4787: >
4788: 8
4789: </text>
4790: <text
4791:   x="20"
4792:   y="68"
4793:   fontSize="10"
4794:   fill="#6b7280"
4795:   textAnchor="middle"
4796: >
4797: 6
4798: </text>
4799: <text
4800:   x="20"
4801:   y="92"
4802:   fontSize="10"
4803:   fill="#6b7280"
4804:   textAnchor="middle"
4805: >
4806: 4
4807: </text>
4808: <text
4809:   x="20"
4810:   y="116"
4811:   fontSize="10"
4812:   fill="#6b7280"
4813:   textAnchor="middle"
4814: >
4815: 2
4816: </text>
4817: <text
4818:   x="20"
4819:   y="140"
4820:   fontSize="10"
4821:   fill="#6b7280"
4822:   textAnchor="middle"
4823: >
4824: 0
4825: </text>
4826:
4827: { /* Bars */
4828: { /* Diet Compatibility */
4829: <rect
4830:   x="60"
4831:   y={140 - totalDietScore * 12}
4832:   width="100"
4833:   height={totalDietScore * 12}
4834:   fill={
4835:     totalDietScore >= 6
4836:     ? "#10b981"
4837:     : totalDietScore >= 4
4838:     ? "#f59e0b"
4839:     : "#ef4444"
4840:   }
4841:   rx="4"
4842: />
4843: <text
4844:   x="110"
4845:   y="160"
4846:   fontSize="11"
4847:   fill="#374151"
4848:   textAnchor="middle"
4849: >
4850:   Diet
4851: </text>
4852: <text
4853:   x="110"
4854:   y="175"
4855:   fontSize="11"
4856:   fill="#374151"
4857:   textAnchor="middle"
4858: >
4859:   Compatibility

```

```

4860:         </text>
4861:         <text
4862:             x="110"
4863:             y="192"
4864:             fontSize="12"
4865:             fontWeight="bold"
4866:             fill={
4867:                 totalDietScore >= 6
4868:                 ? "#059669"
4869:                 : totalDietScore >= 4
4870:                 ? "#d97706"
4871:                 : "#dc2626"
4872:             }
4873:             textAnchor="middle"
4874:         >
4875:             {totalDietScore.toFixed(1)}
4876:         </text>
4877:
4878:         { /* Nutrition Alignment */ }
4879:         <rect
4880:             x="180"
4881:             y={140 - totalNutritionScore * 12}
4882:             width="100"
4883:             height={totalNutritionScore * 12}
4884:             fill={
4885:                 totalNutritionScore >= 6
4886:                 ? "#10b981"
4887:                 : totalNutritionScore >= 4
4888:                 ? "#f59e0b"
4889:                 : "#ef4444"
4890:             }
4891:             rx="4"
4892:         />
4893:         <text
4894:             x="230"
4895:             y="160"
4896:             fontSize="11"
4897:             fill="#374151"
4898:             textAnchor="middle"
4899:         >
4900:             Nutrition
4901:         </text>
4902:         <text
4903:             x="230"
4904:             y="175"
4905:             fontSize="11"
4906:             fill="#374151"
4907:             textAnchor="middle"
4908:         >
4909:             Alignment
4910:         </text>
4911:         <text
4912:             x="230"
4913:             y="192"
4914:             fontSize="12"
4915:             fontWeight="bold"
4916:             fill={
4917:                 totalNutritionScore >= 6
4918:                 ? "#059669"
4919:                 : totalNutritionScore >= 4
4920:                 ? "#d97706"
4921:                 : "#dc2626"
4922:             }
4923:             textAnchor="middle"
4924:         >
4925:             {totalNutritionScore.toFixed(1)}
4926:         </text>
4927:
4928:         { /* Lifestyle Balance */ }
4929:         <rect
4930:             x="300"
4931:             y={
4932:                 140 -

```

```

4933:         (totalLifestyleStrengths > 0
4934:         ? (totalLifestyleStrengths /
4935:           (totalLifestyleStrengths +
4936:             totalLifestyleImprovements)) *
4937:           10
4938:         : 0) *
4939:         12
4940:     }
4941:     width="100"
4942:     height={
4943:       (totalLifestyleStrengths > 0
4944:       ? (totalLifestyleStrengths /
4945:         (totalLifestyleStrengths +
4946:           totalLifestyleImprovements)) *
4947:         10
4948:       : 0) * 12
4949:     }
4950:     fill={
4951:       totalLifestyleStrengths > totalLifestyleImprovements
4952:       ? "#10b981"
4953:       : totalLifestyleStrengths ===
4954:         totalLifestyleImprovements
4955:       ? "#f59e0b"
4956:       : "#ef4444"
4957:     }
4958:     rx="4"
4959:   />
4960:   <text
4961:     x="350"
4962:     y="160"
4963:     fontSize="11"
4964:     fill="#374151"
4965:     textAnchor="middle"
4966:   >
4967:     Lifestyle
4968:   </text>
4969:   <text
4970:     x="350"
4971:     y="175"
4972:     fontSize="11"
4973:     fill="#374151"
4974:     textAnchor="middle"
4975:   >
4976:     Balance
4977:   </text>
4978:   <text
4979:     x="350"
4980:     y="192"
4981:     fontSize="12"
4982:     fontWeight="bold"
4983:     fill={
4984:       totalLifestyleStrengths > totalLifestyleImprovements
4985:       ? "#059669"
4986:       : totalLifestyleStrengths ===
4987:         totalLifestyleImprovements
4988:       ? "#d97706"
4989:       : "#dc2626"
4990:     }
4991:     textAnchor="middle"
4992:   >
4993:     {totalLifestyleStrengths > 0
4994:     ? (
4995:       (totalLifestyleStrengths /
4996:       (totalLifestyleStrengths +
4997:         totalLifestyleImprovements)) *
4998:       10
4999:       ).toFixed(1)
5000:     : "0.0"}
5001:   </text>
5002:
5003:   { /* Metabolic Risk - inverted scale (lower is better) */ }
5004:   <rect
5005:     x="420"

```

```

5006:         y={
5007:             140 -
5008:             (totalMetabolicHigh === 0
5009:               ? 10
5010:               : totalMetabolicHigh <= 2
5011:               ? 6
5012:               : 3) *
5013:             12
5014:         }
5015:         width="100"
5016:         height={
5017:             (totalMetabolicHigh === 0
5018:               ? 10
5019:               : totalMetabolicHigh <= 2
5020:               ? 6
5021:               : 3) * 12
5022:         }
5023:         fill={
5024:             totalMetabolicHigh === 0
5025:               ? "#10b981"
5026:               : totalMetabolicHigh <= 2
5027:               ? "#f59e0b"
5028:               : "#ef4444"
5029:         }
5030:         rx="4"
5031:     />
5032: <text
5033:     x="470"
5034:     y="160"
5035:     fontSize="11"
5036:     fill="#374151"
5037:     textAnchor="middle"
5038: >
5039:     Metabolic
5040: </text>
5041: <text
5042:     x="470"
5043:     y="175"
5044:     fontSize="11"
5045:     fill="#374151"
5046:     textAnchor="middle"
5047: >
5048:     Risk
5049: </text>
5050: <text
5051:     x="470"
5052:     y="192"
5053:     fontSize="12"
5054:     fontWeight="bold"
5055:     fill={
5056:         totalMetabolicHigh === 0
5057:           ? "#059669"
5058:           : totalMetabolicHigh <= 2
5059:           ? "#d97706"
5060:           : "#dc2626"
5061:     }
5062:     textAnchor="middle"
5063: >
5064:     {totalMetabolicHigh === 0
5065:       ? "Low"
5066:       : totalMetabolicHigh <= 2
5067:       ? "Med"
5068:       : "High"}
5069: </text>
5070:
5071: { /* Reference lines */}
5072: <line
5073:     x1="45"
5074:     y1="68"
5075:     x2="540"
5076:     y2="68"
5077:     stroke="#10b981"
5078:     strokeWidth="1"

```



```

5079:         strokeDasharray="3,3"
5080:         opacity="0.6"
5081:     />
5082:     <text x="545" y="72" fontSize="9" fill="#10b981">
5083:         Good (6+)
5084:     </text>
5085:
5086:     <line
5087:         x1="45"
5088:         y1="92"
5089:         x2="540"
5090:         y2="92"
5091:         stroke="#f59e0b"
5092:         strokeWidth="1"
5093:         strokeDasharray="3,3"
5094:         opacity="0.6"
5095:     />
5096:     <text x="545" y="96" fontSize="9" fill="#f59e0b">
5097:         Fair (4+)
5098:     </text>
5099: </svg>
5100: </div>
5101:
5102: { /* Legend */ }
5103: <div className="flex justify-center space-x-4 text-xs">
5104:     <div className="flex items-center">
5105:         <div className="w-3 h-3 bg-green-500 rounded mr-1"></div>
5106:         <span>Excellent/Low Risk</span>
5107:     </div>
5108:     <div className="flex items-center">
5109:         <div className="w-3 h-3 bg-yellow-500 rounded mr-1"></div>
5110:         <span>Good/Medium Risk</span>
5111:     </div>
5112:     <div className="flex items-center">
5113:         <div className="w-3 h-3 bg-red-500 rounded mr-1"></div>
5114:         <span>Needs Attention/High Risk</span>
5115:     </div>
5116: </div>
5117: </div>
5118: </CardContent>
5119: </Card>
5120:
5121: { /* Expert Summaries with Enhanced Detail */ }
5122: <div className="grid grid-cols-2 gap-3">
5123:     <DataCard title="Expert Nutrigenomics Analysis">
5124:         <div className="space-y-2">
5125:             <div className="p-2 bg-blue-50 rounded">
5126:                 <p className="font-semibold text-xs mb-1">
5127:                     Professional Assessment:
5128:                 </p>
5129:                 <p className="text-justify break-words text-xs">
5130:                     {summaries.nutrigenomicsSummary}
5131:                 </p>
5132:             </div>
5133:             <div className="text-xs">
5134:                 <p className="font-semibold">Key Metrics:</p>
5135:                 <ul className="list-disc list-inside space-y-0">
5136:                     <li>
5137:                         Nutritional genetic factors:{" "}
5138:                         {Object.values(nutritionAnalysis).reduce(
5139:                             (sum, cat) => sum + cat.total,
5140:                             0
5141:                         )}
5142:                     </li>
5143:                     <li>
5144:                         Average compatibility score:{" "}
5145:                         {totalNutritionScore.toFixed(1)}/10
5146:                     </li>
5147:                     <li>
5148:                         High-impact factors requiring attention:{" "}
5149:                         {Object.values(nutritionAnalysis).reduce(
5150:                             (sum, cat) => sum + cat.highImpact,
5151:                             0

```

```

5152:         })
5153:     </li>
5154: </ul>
5155: </div>
5156: </div>
5157: </DataCard>
5158:
5159: <DataCard title="Expert Exercise Genomics Analysis">
5160:   <div className="space-y-2">
5161:     <div className="p-2 bg-orange-50 rounded">
5162:       <p className="font-semibold text-xs mb-1">
5163:         Professional Assessment:
5164:       </p>
5165:       <p className="text-justify break-words text-xs">
5166:         {summaries.exerciseGenomicsSummary}
5167:       </p>
5168:     </div>
5169:     <div className="text-xs">
5170:       <p className="font-semibold">Key Metrics:</p>
5171:       <ul className="list-disc list-inside space-y-0">
5172:         {Object.keys(sportsAnalysis).length > 0 && (
5173:           <li>
5174:             Fitness genetic traits analyzed:{" "}
5175:             {Object.values(sportsAnalysis).reduce(
5176:               (sum, section) => sum + section.totalTraits,
5177:               0
5178:             )}
5179:           </li>
5180:         )}
5181:         <li>
5182:           Lifestyle genetic strengths: {totalLifestyleStrengths}
5183:         </li>
5184:         <li>
5185:           Areas for improvement: {totalLifestyleImprovements}
5186:         </li>
5187:         <li>
5188:           Metabolic genetic considerations:{" "}
5189:           {totalMetabolicHigh +
5190:             totalMetabolicModerate +
5191:             totalMetabolicLow}{" "}
5192:           genes
5193:         </li>
5194:       </ul>
5195:     </div>
5196:   </div>
5197: </DataCard>
5198: </div>
5199: </div>
5200: );
5201: })()
5202:
5203: <Separator className="my-4" />
5204:
5205: {/* Final Thank You Page */}
5206: <div className="h-[230mm] bg-gradient-to-br from-blue-100 rounded-lg via-purple-100 to-pink-100 pa
5207: {/* Main content section */}
5208: <div className="flex flex-1 pt-16">
5209:   {/* Left - Thank you text */}
5210:   <div className="flex-1 flex flex-col justify-start px-8">
5211:     <h1 className="text-5xl tracking-wide mb-8 text-gray-800">
5212:       Thank You...
5213:     </h1>
5214:     <div className="mb-6">
5215:       <p className="text-lg font-medium mb-2 text-gray-700">To,</p>
5216:       <p className="text-xl font-semibold text-gray-900">
5217:         {patientInfo.name}
5218:       </p>
5219:     </div>
5220:     <p className="text-base leading-relaxed mb-8 text-gray-800 font-light max-w-lg text-justify">
5221:       Thank you for taking the time to review our nutrigenomics
5222:       report. We appreciate the opportunity to conduct this analysis
5223:       for you. For further inquiries or personalized guidance, please
5224:       feel free to connect with us using the contact details provided

```

```

5225:         below.
5226:     </p>
5227: <div className="text-sm leading-relaxed text-gray-700 space-y-4">
5228:     <div>
5229:         <p className="font-bold text-gray-900 mb-1">
5230:             Molsys Private Limited
5231:         </p>
5232:         <p className="font-light leading-relaxed">
5233:             Regd. Office: 1st Floor, #524, 13th Main, 23rd Cross,
5234:             Judicial Layout, Yelahanka
5235:         </p>
5236:         <p className="font-light leading-relaxed">
5237:             { " " }
5238:             Bengaluru, Karnataka, Pin: 560065
5239:         </p>
5240:         <a
5241:             href="https://www.molsys.in"
5242:             className="font-light "
5243:             target="_blank"
5244:             rel="noopener noreferrer"
5245:         >
5246:             Website: https://www.molsys.in
5247:         </a>
5248:
5249:         <br />
5250:
5251:         <a href="mailto:wellness@molsys.in" className="font-light ">
5252:             Email: wellness@molsys.in
5253:         </a>
5254:
5255:         <p className="font-light">
5256:             Contact no: 8271502582 / 8073193805
5257:         </p>
5258:     </div>
5259:
5260:     <div className="pt-5">
5261:         <p className="font-bold text-gray-900 mb-1">
5262:             Molsys Private Limited (R&D)
5263:         </p>
5264:         <p className="font-light leading-relaxed">
5265:             Yenepoya Technology Incubator, Yenepoya (deemed-to-be)
5266:             University,
5267:         </p>
5268:         <p className="font-light leading-relaxed">
5269:             Deralakatte, Ullal, DK Pin: 575020
5270:         </p>
5271:     </div>
5272: </div>
5273: </div>
5274: </div>
5275: { /* Bottom logos */ }
5276: <div className="absolute bottom-4 left-4">
5277:     
5282: </div>
5283: <div className="absolute bottom-4 right-4">
5284:     
5289: </div>
5290: </div>
5291:
5292: { /* <footer className="text-center mt-4 py-3 text-gray-600 text-xs border-t border-gray-200">
5293:     <div className="space-y-1">
5294:         <p>
5295:             &copy; {new Date().getFullYear()} {settings.companyName}. All
5296:             rights reserved.
5297:         </p>

```

```

5298:         <p className="break-words">
5299:             Report Authentication: {patientInfo.reportAuth}
5300:         </p>
5301:     </div>
5302: </footer> */}
5303: </div>
5304: </>
5305: );
5306: }

```

■ 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=open]:animate-accordion-down"
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-[state=closed]:fade-out-0 data-[state=open]:fade-in-0",
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 border border-gray-200 bg-white p-6 shadow-lg",
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/utlis";
6:
7: const buttonVariants = cva(
8:   "inline-flex items-center justify-center gap-2 whitespace-nowrap rounded-md text-sm font-medium ring-offse
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
18:         secondary:
19:           "bg-gradient-to-r from-gray-200 to-gray-300 text-gray-800 hover:from-gray-300 hover:to-gray-400 s
20:         ghost:
21:           "bg-gradient-to-r from-transparent to-transparent hover:from-blue-50 hover:to-purple-50 text-gray
22:         link: "bg-gradient-to-r from-blue-600 to-purple-600 bg-clip-text text-transparent underline-offset-
23:         success:
24:           "bg-gradient-to-r from-green-400 to-emerald-500 text-white hover:from-green-500 hover:to-emerald-
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/Utils"
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 (

```



```

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-
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 [&_[cmdk-group-heading]]:text-sm",
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-md",
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-[s
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 b
42:       className
43:     )}
44:     {...props}
45:   >
46:     {children}
47:     <DialogPrimitive.Close className="absolute right-4 top-4 rounded-sm opacity-70 ring-offset-background
48:       <X className="h-4 w-4" />
49:       <span className="sr-only">Close</span>
50:     </DialogPrimitive.Close>
51:   </DialogPrimitive.Content>
52: </DialogPortal>
53: ))
54: DialogContent.displayName = DialogPrimitive.Content.displayName
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-n
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-no
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 transit
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-ba
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",
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-visible:outline-2 focus-visible:outline-blue-500",
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-none focus-visible:outline-2 focus-visible:outline-blue-500",
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-none focus-visible:outline-2 focus-visible:outline-blue-500",
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-]:animate-out"
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-hidden"
90:       )}
91:       ref={ref}
92:       {...props}
93:     />
94:   </div>
95: ))
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 data-[state=closed]:animate-out"
108:     )}
109:     {...props}
110:   />
111:   <div className="relative top-[60%] h-2 w-2 rotate-45 rounded-tl-sm bg-border shadow-md" />
112:   </NavigationMenuPrimitive.Indicator>
113: ))
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:   HTMMLIElement,
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-foreground",
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 out
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",
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
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/utils"
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:     onChange?: (open: boolean) => void
56:   }
57: >(
58:   (
59:     {
60:       defaultOpen = true,
61:       open: openProp,
62:       onChange: 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 [&gt;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:             )}
246:             {...props}
247:         >
248:             <div
249:                 data-sidebar="sidebar"
250:                 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-foreground"
251:             >
252:                 {children}
253:             </div>
254:         </div>
255:     </div>
256: )
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={
276:       (event) => {
277:         onClick?.(event)
278:         toggleSidebar()
279:       }
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:
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]
306:         "group-data-[collapsible=offcanvas]:translate-x-0 group-data-[collapsible=offcanvas]:after:left-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))]",
327:         "md:peer-data-[variant=inset]:m-2 md:
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 ta
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-1 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]: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 rou
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-
76:         <X className="h-4 w-4" />
77:       </ToastPrimitives.Close>
78:     )}
79:   </ToastPrimitives.Root>
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",
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\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\utils.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}...`);
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: },
25: "include": ["next-env.d.ts", "**/*.ts", "**/*.tsx", ".next/types/**/*.ts"],
26: "exclude": ["node_modules"]
27: }

```

■ 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: }
```
