

■ TypeScript Project Code Export

■ TypeScript/TSX Files & Safe Config:

```
- [TSX] app\admin\comprehensive-admin.tsx
- [TSX] app\admin\page.tsx
- [TS] app\api\comprehensive-report-data\route.ts
- [TS] app\api\report-data\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\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\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-content-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\admin\summaries-admin.tsx
- [TSX] components\comprehensive-report-viewer.tsx
- [TSX] components\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\utils.ts
- [TS] next-env.d.ts
- [PKG] package.json
- [TS] tailwind.config.ts
- [TSC] tsconfig.json
- [TS] types\report-types.ts

■ File: app\admin\comprehensive-admin.tsx

```
=====
1: // "use client"
2:
3: // import { useState, useEffect } from "react"
4: // import { Button } from "@/components/ui/button"
5: // import { Tabs, TabsContent, TabsList, TabsTrigger } from "@/components/ui/tabs"
6: // import { ScrollArea } from "@/components/ui/scroll-area"
7: // import { Save, ArrowLeft } from "lucide-react"
8: // import Link from "next/link"
9:
10: // // Import all admin components
11: // import PatientInfoAdmin from "@/components/admin/patient-info-admin"
12: // import ContentAdmin from "@/components/admin/content-admin"
13: // import SettingsAdmin from "@/components/admin/settings-admin"
14: // import DietAnalysisAdmin from "@/components/admin/diet-analysis-admin"
15: // import GeneResultsAdmin from "@/components/admin/gene-results-admin"
16: // import ImportExportAdmin from "@/components/admin/import-export-admin"
17: // import SummariesAdmin from "@/components/admin/summaries-admin"
18:
19: // import type {
20: //   ComprehensiveReportData,
21: //   PatientInfo,
22: //   ReportContent,
23: //   ReportSettings,
24: //   DietAnalysis,
25: //   MacronutrientData,
26: //   GeneTestResult,
27: //   ReportSummaries,
28: // } from "@/types/report-types"
29:
30: // export default function ComprehensiveAdminPage() {
31: //   const [reportData, setReportData] = useState<ComprehensiveReportData>({
32: //     patientInfo: {
33: //       name: "",
34: //       gender: "",
35: //       birthDate: "",
36: //       sampleCode: "",
37: //       sampleDate: "",
38: //       reportDate: "",
39: //       reportAuth: "",
40: //       genomicAnalytics: "",
41: //       checkedBy: "",
42: //       scientificContent: "",
43: //     },
44: //     content: {
45: //       disclaimer: `The present report tells you about genetic variations/mutations associated with certain
46: //       This report cannot replace visits to a healthcare professional. Your results serve as a reference point,
47: //       Example, pvt , a comprehensive health app that leverages genetic, metabolic, anthropometric, and clinical
48: //       introduction: `Welcome to your personalised nutrigenomics test report. It is our absolute pleasure
49: //       genomicsExplanation: `We all are aware that DNA is the blueprint of the human body. The external factors
50: //       genesHealthImpact: `The human DNA consists of more than 20000 genes. Every gene has a role to play in
51: //       fundamentalsPRS: `The risk of an individual to suffer from a chronic condition, such as lifestyle diseases,
52: //       utilityDoctors: `Your polygenic risk scores are not something you should be scared of, as you can consult
53: //       microarrayExplanation: `A microarray is a process that involves detecting several (approximately 0.5 million)
54: //       microarrayData: `The microarray data obtained from your DNA is precious and has been archived on a secure
55: //     },
56: //     settings: {
57: //       title: "COMPREHENSIVE GENOMICS REPORT",
58: //       subtitle: "NUTRIGENOMICS TEST REPORT",
59: //       companyName: "Example PVT LTD",
60: //       headerColor: "#1e40af",
61: //       accentColor: "#3b82f6",
62: //       fonts: {
63: //         primary: "Inter, system-ui, sans-serif",
64: //         secondary: "Roboto, system-ui, sans-serif",
65: //         mono: "JetBrains Mono, Consolas, monospace",
66: //       },
67: //     },
68: //     dietAnalysis: {
69: //       macronutrients: {
70: //       },
71: //     },
72: //   })
73: // }
```

```

72: //      carbohydrateSensitivity: {
73: //          score: 3,
74: //          level: "HIGH",
75: //          recommendation: "Maintain carb intake <45% For obesity & IR control",
76: //      },
77: //      fatSensitivity: { score: 6, level: "HIGH", recommendation: "Fat intake not to exceed 15% of total",
78: //      proteinRequirement: {
79: //          score: 6,
80: //          level: "HIGH",
81: //          recommendation: "Protein supplements needed along with dietary source",
82: //      },
83: //  },
84: //  mealPattern: {
85: //      mealFrequency: { score: 6, level: "FREQUENT", recommendation: "4-5 small meals suggested in a day",
86: //      mealReplacement: { score: 6, level: "SUITABLE", recommendation: "Meal replacements may help in weight loss",
87: //      weightMaintenance: { score: 6, level: "HIGH", recommendation: "Weight maintenance regime can be helpful",
88: //  },
89: //  foodSensitivities: {
90: //      alcoholSensitivity: {
91: //          score: 4,
92: //          level: "HIGH",
93: //          recommendation:
94: //              "High sensitivity, avoid alcohol if possible, especially the types of beverages that trigger sensitivity",
95: //      },
96: //      caffeineSensitivity: {
97: //          score: 6,
98: //          level: "HIGH",
99: //          recommendation: "High sensitivity, do not consume >4 cups/day",
100: //      },
101: //      glutenSensitivity: { score: 3, level: "HIGH", recommendation: "Gluten intake needs to be reduced",
102: //      lactoseSensitivity: { score: 10, level: "HIGH", recommendation: "Milk & milk products need to be avoided",
103: //      saltSensitivity: {
104: //          score: 10,
105: //          level: "HIGH",
106: //          recommendation: "Try to reduce overall salt intake to up to 3-5 gm per day",
107: //      },
108: //  },
109: //  tasteSensitivities: {
110: //      spiceTolerance: { score: 6, level: "NORMAL", recommendation: "Normal sensitivity to taste" },
111: //      sweetTaste: { score: 3, level: "HIGH", recommendation: "High sweet consumption tendency" },
112: //      tasteSensitivity: { score: 6, level: "NORMAL", recommendation: "Normal taste sensitivity" },
113: //  },
114: //  },
115: //  nutritionData: {
116: //      vitamins: {
117: //          vitaminA: { score: 3, healthImpact: "Skin & Vision", intakeLevel: "ENHANCED INTAKE", source: "DIET",
118: //          vitaminB12: {
119: //              score: 6,
120: //              healthImpact: "Nervous System & Bones",
121: //              intakeLevel: "ENHANCED INTAKE (METHYLCOBALAMIN)",
122: //              source: "DIET",
123: //          },
124: //          vitaminB2: { score: 3, healthImpact: "Energy Metabolism", intakeLevel: "ENHANCED INTAKE", source: "DIET",
125: //          vitaminB6: { score: 6, healthImpact: "Tissue growth repair", intakeLevel: "ENHANCED INTAKE", source: "DIET",
126: //          vitaminB9: {
127: //              score: 3,
128: //              healthImpact: "RBC development & Heart health",
129: //              intakeLevel: "ENHANCED INTAKE (L METHYLFOLATE)",
130: //              source: "DIET",
131: //          },
132: //          vitaminC: { score: 3, healthImpact: "Skin & Immunity", intakeLevel: "ENHANCED INTAKE", source: "DIET",
133: //          vitaminD: {
134: //              score: 6,
135: //              healthImpact: "Bones & Diabetes",
136: //              intakeLevel: "ENHANCED INTAKE & SUN EXPOSURE",
137: //              source: "DIET & SUN EXPOSURE",
138: //          },
139: //          vitaminE: { score: 5, healthImpact: "Skin & Immune system", intakeLevel: "ENHANCED INTAKE", source: "DIET",
140: //      },
141: //      fattyAcids: {
142: //          omegaFattyAcids: {
143: //              score: 5,
144: //              healthImpact: "Cardiac health & Triglycerides",

```

```

145: //         intakeLevel: "ENHANCED INTAKE (VEG SOURCES)",
146: //         source: "ENHANCED INTAKE (VEG/FISH OIL) & DIET",
147: //     },
148: // },
149: // elements: {
150: //     calcium: {
151: //         score: 3,
152: //         healthImpact: "Bone health & Nerve function",
153: //         intakeLevel: "ENHANCED INTAKE",
154: //         source: "DIET",
155: //     },
156: //     iodine: {
157: //         score: 3,
158: //         healthImpact: "Detoxification & Thyroid hormones",
159: //         intakeLevel: "ENHANCED INTAKE",
160: //         source: "DIET",
161: //     },
162: //     iron: { score: 3, healthImpact: "Haemoglobin & Energy", intakeLevel: "RESTRICTED INTAKE", source: "DIET",
163: //     magnesium: {
164: //         score: 3,
165: //         healthImpact: "Heart health & Immunity",
166: //         intakeLevel: "ENHANCED INTAKE",
167: //         source: "DIET",
168: //     },
169: //     phosphorus: {
170: //         score: 3,
171: //         healthImpact: "Bone health & Kidney function",
172: //         intakeLevel: "ENHANCED INTAKE",
173: //         source: "DIET",
174: //     },
175: //     selenium: {
176: //         score: 3,
177: //         healthImpact: "Cognitive & Muscle function",
178: //         intakeLevel: "ENHANCED INTAKE",
179: //         source: "DIET",
180: //     },
181: //     zinc: { score: 3, healthImpact: "Fertility & Skin", intakeLevel: "ENHANCED INTAKE", source: "DIET",
182: // },
183: // complexNutrients: {
184: //     antiInflammatory: {
185: //         score: 5,
186: //         healthImpact: "Pain control & Organ health",
187: //         intakeLevel: "ENHANCED INTAKE",
188: //         source: "DIET",
189: //     },
190: //     antioxidants: {
191: //         score: 6,
192: //         healthImpact: "Skin & Immune system",
193: //         intakeLevel: "ENHANCED INTAKE",
194: //         source: "DIET",
195: //     },
196: // },
197: // },
198: // sportsAndFitness: {
199: //     exerciseType: {
200: //         endurancePotential: {
201: //             level: "High Endurance Potential",
202: //             description: "Excellent genetic predisposition for endurance activities",
203: //         },
204: //         powerPotential: { level: "Average Power Potential", description: "Moderate power output capacity",
205: //         strengthProfile: { level: "High Strength Profile", description: "Good potential for strength training",
206: //         exerciseTime: { level: "Evening time is more beneficial than morning", description: "MORNING Vs. EVENING",
207: //     },
208: //     performance: {
209: //         enduranceCapacity: { level: "High Endurance Potential", description: "Superior aerobic capacity",
210: //         fatigueResistance: {
211: //             level: "Normal fatigue resistance. Average endurance capacity",
212: //             description: "Average endurance capacity",
213: //         },
214: //         bloodFlow: {
215: //             level: "Reduced vascular efficiency, affects endurance sports.",
216: //             description: "Affects endurance sports",
217: //         },

```

```

218: //      oxygenEfficiency: {
219: //          level: "Normal VO2 Max. Need practice to excel in endurance sports",
220: //          description: "Need practice to excel",
221: //      },
222: //      waterLoss: {
223: //          level:
224: //              "Increased water and electrolyte loss tendency. Focus on Hydration and electrolyte management",
225: //          description: "Focus on hydration",
226: //      },
227: //      lactateClearing: {
228: //          level: "High lactate clearance. Beneficial for Endurance sports.",
229: //          description: "Beneficial for endurance",
230: //      },
231: //      recoveryEfficiency: {
232: //          level: "High recovery efficiency. Enhanced sports potential & reduced injury risk",
233: //          description: "Enhanced sports potential",
234: //      },
235: //      bodyComposition: {
236: //          level: "Low fat accumulation. Increased muscle tendency.",
237: //          description: "Increased muscle tendency",
238: //      },
239: //      injuryRisk: {
240: //          level: "High injury tendency. Thorough injury management steps to be taken.",
241: //          description: "Thorough injury management needed",
242: //      },
243: //      achillesTendonInjury: {
244: //          level: "Low injury tendency. Ensure adequate warmup before exercises",
245: //          description: "Ensure adequate warmup",
246: //      },
247: //      ligamentInjury: {
248: //          level: "Low ACL injury tendency. Ensure adequate warmup before exercises",
249: //          description: "Ensure adequate warmup",
250: //      },
251: //      rupture: {
252: //          level: "Low rupture tendency. Ensure adequate warmup before exercises",
253: //          description: "Ensure adequate warmup",
254: //      },
255: //      muscleCramps: {
256: //          level: "Low risk of muscle cramps. Ensure adequate warmup before exercises",
257: //          description: "Ensure adequate warmup",
258: //      },
259: //  },
260: //  },
261: //  lifestyleConditions: {
262: //      heartVascularHealth: {
263: //          hypertension: { status: "strength" },
264: //          leftVentricularHypertrophy: { status: "strength" },
265: //          atherosclerosis: { status: "improvement" },
266: //          familialHypercholesterolemia: { status: "strength" },
267: //          cardiomyopathy: { status: "strength" },
268: //          cholesterolImbalance: { status: "improvement" },
269: //          stroke: { status: "improvement" },
270: //      },
271: //      diabetesity: {
272: //          obesity: { status: "improvement" },
273: //          diabetesType2: { status: "improvement" },
274: //      },
275: //      liverHealth: {
276: //          nafld: { status: "improvement" },
277: //          liverCirrhosis: { status: "strength" },
278: //      },
279: //      boneHealth: {
280: //          osteoporosis: { status: "improvement" },
281: //          osteoarthritis: { status: "improvement" },
282: //      },
283: //      gutHealth: {
284: //          irritableBowelSyndrome: { status: "improvement" },
285: //          celiacDisease: { status: "strength" },
286: //      },
287: //      hormoneSystem: {
288: //          hypothyroidism: { status: "improvement" },
289: //      },
290: //  },

```

```

291: //      metabolicCore: {
292: //          methylation: {
293: //              genes: ["MTHFR (A>C)", "MTHFR (C>T)", "MTR", "MTRR", "CBS"],
294: //              genotype: "TT, GG, GG, GG, CC",
295: //              impact: "NORMAL METHYLATION",
296: //              advice: "FOLIC ACID & METHYL COBALAMINE NEEDED TO MAINTAIN HOMOCYSTEINE",
297: //          },
298: //          inflammation: {
299: //              genes: ["TNF-A", "IL6", "CRP"],
300: //              genotype: "GG, GG, TC",
301: //              impact: "HIGH INFLAMMATION",
302: //              advice: "ANTI-INFLAMMATORY SUPPLEMENT INTAKE SUGGESTED",
303: //          },
304: //          oxidativeStress: {
305: //              genes: ["MnSOD", "SOD2"],
306: //              genotype: "AG, AG",
307: //              impact: "HIGH OXIDATIVE STRESS",
308: //              advice: "ANTIOXIDANT SUPPLEMENTS REQUIRED TO MANAGE OXIDATIVE STRESS",
309: //          },
310: //          detoxification: {
311: //              genes: ["GSTP1", "CYP1A2"],
312: //              genotype: "AA, AC",
313: //              impact: "AFFECTED DETOXIFICATION ABILITY",
314: //              advice: "SUPPORT LIVER DETOXIFICATION WITH CRUCIFEROUS VEGETABLES",
315: //          },
316: //          vitaminDMetabolism: {
317: //              genes: ["VDR-3 (Taq)", "VDR-2 (Bsm)"],
318: //              genotype: "AG, TC",
319: //              impact: "LOW VIT D. METABOLISM",
320: //              advice: "HIGHER VITAMIN D SUPPLEMENTATION NEEDED",
321: //          },
322: //          lipidMetabolism: {
323: //              genes: ["CETP", "APOE", "LPL", "APOC3"],
324: //              genotype: "AA, TC, CC, GG",
325: //              impact: "HIGH TENDENCY OF DYSLIPIDEMIA",
326: //              advice: "PCF RATIO 30:55:15% LPL= LDL, CETP= HDL, APOC3=TRIGLYCERIDES",
327: //          },
328: //          insulinResistance: {
329: //              genes: ["TCF7L2", "PPARG", "FTO"],
330: //              genotype: "TT, CC, AT",
331: //              impact: "HIGH INSULIN RESISTANCE (IR)",
332: //              advice: "FTO = IR TENDENCY TCF7L2 = OBESITY TRIGGERED IR PPARG= CENTRAL OBESITY",
333: //          },
334: //          circadianRhythm: {
335: //              genes: ["CLOCK"],
336: //              genotype: "AG",
337: //              impact: "EVENING EXERCISE TO BE PREFERRED",
338: //              advice: "EVENING EXERCISE TO BE PREFERRED",
339: //          },
340: //          exerciseResponsiveness: {
341: //              genes: ["ADRB2-2", "ADRB2-1", "ADRB3"],
342: //              genotype: "CC, GG, AG",
343: //              impact: "SLOW RESPONSE TO WORKOUT",
344: //              advice: "HIGH INTENSITY WORKOUT ADRB2-2 = CARBS SENSITIVITY",
345: //          },
346: //          snackingTendency: {
347: //              genes: ["MC4R"],
348: //              genotype: "TC",
349: //              impact: "SNACKING TENDENCY PRESENT",
350: //              advice: "CONTROL SNACKING HABITS WITH STRUCTURED MEAL TIMING",
351: //          },
352: //          oestrogenElimination: {
353: //              genes: ["COMT", "CYP1A1"],
354: //              genotype: "GG, CG",
355: //              impact: "AFFECTED OESTROGEN ELIMINATION",
356: //              advice: "SUPPORT ESTROGEN DETOX WITH CRUCIFEROUS VEGETABLES",
357: //          },
358: //          powerVsEndurance: {
359: //              genes: ["ACE", "AGT", "ACTN3"],
360: //              genotype: "AA, GG, TT",
361: //              impact: "HIGH POWER POTENTIAL",
362: //              advice: "ACE (WHITE) = ENDURANCE AGT (RED) = LVH/HT",
363: //          },

```

```

364: //      injuryRecovery: {
365: //          genes: ["COL1A1", "GDF5"],
366: //          genotype: "CC, AG",
367: //          impact: "INCREASED RISK OF INJURY",
368: //          advice: "FOCUS ON INJURY PREVENTION AND PROPER WARM-UP",
369: //      },
370: //      fluidLossTendency: {
371: //          genes: ["BDKRB2"],
372: //          genotype: "TC",
373: //          impact: "HIGH FLUID LOSS TENDENCY",
374: //          advice: "MAINTAIN PROPER HYDRATION DURING EXERCISE",
375: //      },
376: //      vo2Max: {
377: //          genes: ["VEGF"],
378: //          genotype: "GG",
379: //          impact: "AVERAGE VO2 MAX",
380: //          advice: "FOCUS ON AEROBIC TRAINING TO IMPROVE VO2 MAX",
381: //      },
382: //  },
383: //  digestiveHealth: {
384: //      gutHealth: { level: "NORMAL GUT HEALTH", description: "Good overall digestive function" },
385: //      glutenIntolerance: { level: "LOW INTOLERANCE", description: "CONDITIONAL GLUTEN ELIMINATION NEEDED" },
386: //      celiacDiseaseTendency: { level: "LOW TENDENCY", description: "Low genetic risk for celiac disease" },
387: //      impactOnBonesJoints: {
388: //          level: "NO IMPACT ON BONE HEALTH",
389: //          description: "Digestive health does not affect bone health",
390: //      },
391: //      lactoseIntolerance: { level: "HIGH INTOLERANCE", description: "LACTOSE ELIMINATION NEEDED" },
392: //      probioticsRequirement: { level: "PROBIOTICS NEEDED", description: "FOOD & SUPPLEMENTS" },
393: //  },
394: //  genesAndAddiction: {
395: //      alcoholAddiction: { tendency: "LOW", description: "Low genetic predisposition to alcohol dependence" },
396: //      smokingAddiction: { tendency: "NORMAL", description: "Average nicotine addiction susceptibility" },
397: //      snackingHabits: { tendency: "HIGH", description: "Genetic tendency toward frequent snacking" },
398: //      altruism: { tendency: "HIGH", description: "Strong genetic predisposition toward helping others" },
399: //  },
400: //  sleepAndRest: {
401: //      sleepCycle: {
402: //          impact: "DELAYED SLEEP CYCLE",
403: //          intervention: "EXERCISE IN THE EVENING AVOID CARBS DURING BREAKFAST",
404: //      },
405: //      sleepApnea: {
406: //          impact: "INCREASED CHANCES OF SLEEP APNEA",
407: //          intervention: "EXERCISE REGULARLY SEEK MEDICAL ADVICE IF ISSUE PERSISTS",
408: //      },
409: //      sleepDepth: { impact: "LOW SLEEP DEPTH", intervention: "CONSUME CURD RICE IN DINNER FOR BETTER SLEEP" },
410: //      stressSensitivity: {
411: //          impact: "LOW STRESS SENSITIVITY",
412: //          intervention: "EXERCISE REGULARLY FOR A BETTER SLEEP CYCLE",
413: //      },
414: //  },
415: //  allergiesAndSensitivity: {
416: //      dustAllergy: { tendency: "NORMAL" },
417: //      pesticideSensitivity: { tendency: "HIGH" },
418: //      smokeSensitivity: { tendency: "HIGH" },
419: //      automobileSmokeSensitivity: { tendency: "HIGH" },
420: //      generalAdvice:
421: //          "THE BEST WAY TO REDUCE ALLERGIES IS TO STAY AWAY FROM THE SOURCE, NO ROCKET SCIENCE HOWEVER THE",
422: //  },
423: //  geneTestResults: [
424: //      { geneName: "ALDH2 Aldehyde dehydrogenase 2", variation: "G>A", result: "GG" },
425: //      { geneName: "ACTN3 Alpha-actinin-3 protein", variation: "C>T", result: "TT" },
426: //      { geneName: "ACE Angiotensin I-Converting Enzyme", variation: "G>A", result: "AA" },
427: //      { geneName: "AGT Angiotensinogen", variation: "G>A", result: "GG" },
428: //      { geneName: "ApoA5 Apolipoprotein A5", variation: "G>A", result: "GG" },
429: //      { geneName: "APOC3 Apolipoprotein C 3", variation: "C>G", result: "GG" },
430: //      { geneName: "APOE Apolipoprotein E", variation: "C>T", result: "TC" },
431: //      { geneName: "APOE4 Apolipoprotein E (TOMM40)", variation: "A>G", result: "AA" },
432: //      { geneName: "ADRB2-1 BETA-2-ADRENERGIC RECEPTOR (1)", variation: "A>G", result: "GG" },
433: //      { geneName: "ADRB2-2 BETA-2-ADRENERGIC RECEPTOR (2)", variation: "C>G", result: "CC" },
434: //  ],
435: //  metabolicSummary: {
436: //      strengths: [

```



```

437: //      "NORMAL METHYLATION",
438: //      "HIGH POWER POTENTIAL",
439: //      "NORMAL GUT HEALTH",
440: //      "LOW STRESS SENSITIVITY",
441: //      "HIGH RECOVERY EFFICIENCY",
442: //  ],
443: //  weaknesses: [
444: //      "HIGH INFLAMMATION",
445: //      "HIGH OXIDATIVE STRESS",
446: //      "AFFECTED DETOXIFICATION ABILITY",
447: //      "LOW VIT D. METABOLISM",
448: //      "HIGH INSULIN RESISTANCE",
449: //      "AFFECTED OESTROGEN ELIMINATION",
450: //      "INCREASED RISK OF INJURY",
451: //      "HIGH FLUID LOSS TENDENCY",
452: //  ],
453: //  },
454: //  preventiveHealth: {
455: //      diagnosticTests: {
456: //          halfYearly: ["PLASMA SUGAR (FASTING & PP)", "VITAMIN D"],
457: //          yearly: ["HbA1c", "LIPID PROFILE", "SERUM HOMOCYSTEINE", "VITAMIN B12", "LIVER PROFILE"],
458: //      },
459: //      nutritionalSupplements: [
460: //          { supplement: "VITAMIN D", needed: true },
461: //          { supplement: "METHYL FOLATE + METHYL COBALAMINE + B6", needed: true },
462: //          { supplement: "OMEGA FATTY ACIDS (FISH OIL)", needed: true },
463: //          { supplement: "FOLIC ACID + COBALAMINE + B6", needed: true },
464: //          { supplement: "OMEGA FATTY ACIDS (VEG)", needed: true },
465: //          { supplement: "VITAMIN E", needed: true },
466: //          { supplement: "PROBIOTICS", needed: true },
467: //      ],
468: //  },
469: //  familyGeneticImpact: [
470: //      {
471: //          gene: "MnSOD",
472: //          normalAlleles: "AA",
473: //          yourResult: "AG",
474: //          healthImpact: "INCREASED TENDENCY OF OXIDATIVE STRESS NEED OF ANTI-OXIDANT SUPPLEMENTS",
475: //      },
476: //      {
477: //          gene: "FTO",
478: //          normalAlleles: "TT",
479: //          yourResult: "AT",
480: //          healthImpact:
481: //              "INCREASED TENDENCY OF INSULIN RESISTANCE WEIGHT AND CALORIE MANAGEMENT IMPORTANT TO MANAGE AND",
482: //      },
483: //  ],
484: //  summaries: {
485: //      nutrigenomicsSummary:
486: //          "Based on your genetic analysis, you have specific nutritional needs including enhanced vitamin B12 and omega-3 fatty acids.",
487: //      exerciseGenomicsSummary:
488: //          "Your genetic profile indicates high endurance potential with good recovery capacity. Focus on aerobic and strength training.",
489: //  },
490: //  })
491:
492: //  const [isSaving, setIsSaving] = useState(false)
493: //  const [loading, setLoading] = useState(true)
494:
495: //  useEffect(() => {
496: //      loadReportData()
497: //  }, [])
498:
499: //  const loadReportData = async () => {
500: //      try {
501: //          const response = await fetch("/api/comprehensive-report-data")
502: //          if (response.ok && response.headers.get("content-type").includes("application/json")) {
503: //              const data = await response.json()
504: //              setReportData(data)
505: //          }
506: //      } catch (error) {
507: //          console.error("Error loading report data:", error)
508: //      } finally {
509: //          setLoading(false)

```

```

510: //     }
511: //   }
512:
513: //   const saveReportData = async () => {
514: //     setIsSaving(true)
515: //     try {
516: //       const response = await fetch("/api/comprehensive-report-data", {
517: //         method: "POST",
518: //         headers: {
519: //           "Content-Type": "application/json",
520: //         },
521: //         body: JSON.stringify(reportData),
522: //       })
523:
524: //       if (response.ok) {
525: //         alert("Report data saved successfully!")
526: //       }
527: //     } catch (error) {
528: //       console.error("Error saving report data:", error)
529: //       alert("Error saving report data")
530: //     } finally {
531: //       setIsSaving(false)
532: //     }
533: //   }
534:
535: //   // Helper functions for updating nested data
536: //   const updatePatientInfo = (field: keyof PatientInfo, value: string) => {
537: //     setReportData((prev) => ({
538: //       ...prev,
539: //       patientInfo: { ...prev.patientInfo, [field]: value },
540: //     }))
541: //   }
542:
543: //   const updateContent = (field: keyof ReportContent, value: string) => {
544: //     setReportData((prev) => ({
545: //       ...prev,
546: //       content: { ...prev.content, [field]: value },
547: //     }))
548: //   }
549:
550: //   const updateSettings = (field: keyof ReportSettings, value: string) => {
551: //     setReportData((prev) => ({
552: //       ...prev,
553: //       settings: { ...prev.settings, [field]: value },
554: //     }))
555: //   }
556:
557: //   const updateDietAnalysis = (section: keyof DietAnalysis, field: string, data: Partial<MacronutrientData>) => {
558: //     setReportData((prev) => ({
559: //       ...prev,
560: //       dietAnalysis: {
561: //         ...prev.dietAnalysis,
562: //         [section]: {
563: //           ...prev.dietAnalysis[section],
564: //           [field]: {
565: //             ...prev.dietAnalysis[section][field as keyof (typeof prev.dietAnalysis)[typeof section]],
566: //             ...data,
567: //           },
568: //         },
569: //       },
570: //     }))
571: //   }
572:
573: //   const updateSummaries = (field: keyof ReportSummaries, value: string) => {
574: //     setReportData((prev) => ({
575: //       ...prev,
576: //       summaries: { ...prev.summaries, [field]: value },
577: //     }))
578: //   }
579:
580: //   // Gene test results management
581: //   const addGeneTestResult = () => {
582: //     setReportData((prev) => ({

```

```

583: //      ...prev,
584: //      geneTestResults: [...prev.geneTestResults, { geneName: "", variation: "", result: "" }],
585: //    )))
586: //  }
587:
588: //  const updateGeneTestResult = (index: number, field: keyof GeneTestResult, value: string) => {
589: //    setReportData((prev) => {
590: //      const newResults = [...prev.geneTestResults]
591: //      newResults[index] = { ...newResults[index], [field]: value }
592: //      return { ...prev, geneTestResults: newResults }
593: //    })
594: //  }
595:
596: //  const removeGeneTestResult = (index: number) => {
597: //    setReportData((prev) => ({
598: //      ...prev,
599: //      geneTestResults: prev.geneTestResults.filter((_, i) => i !== index),
600: //    })))
601: //  }
602:
603: //  if (loading) {
604: //    return (
605: //      <div className="min-h-screen flex items-center justify-center bg-gradient-to-br from-blue-50 to-in
606: //      <div className="text-center">
607: //        <div className="animate-spin rounded-full h-16 w-16 border-b-4 border-blue-600 mx-auto mb-6"><
608: //        <p className="text-gray-600 text-lg">Loading comprehensive admin panel...</p>
609: //      </div>
610: //    </div>
611: //  )
612: //  }
613:
614: //  return (
615: //    <div className="min-h-screen bg-gradient-to-br from-gray-50 to-blue-50">
616: //      <div className="container mx-auto px-4 py-8">
617: //        <div className="flex items-center justify-between mb-8">
618: //          <div className="flex items-center gap-4">
619: //            <Link href="/">
620: //              <Button variant="outline" size="sm" className="shadow-sm bg-transparent">
621: //                <ArrowLeft className="h-4 w-4 mr-2" />
622: //                Back to Home
623: //              </Button>
624: //            </Link>
625: //            <h1 className="text-4xl font-bold text-gray-900">? Comprehensive Genomics Admin</h1>
626: //          </div>
627: //          <div className="flex gap-2">
628: //            <Link href="/report/preview">
629: //              <Button variant="outline" className="shadow-sm bg-transparent">
630: //                ?? Preview Report
631: //              </Button>
632: //            </Link>
633: //            <Button onClick={saveReportData} disabled={isSaving} className="shadow-sm">
634: //              <Save className="h-4 w-4 mr-2" />
635: //              {isSaving ? "Saving..." : "Save All Changes"}
636: //            </Button>
637: //          </div>
638: //        </div>
639:
640: //        <Tabs defaultValue="patient-info" className="space-y-6">
641: //          <ScrollArea className="w-full whitespace-nowrap">
642: //            <TabsList className="grid w-full grid-cols-8 gap-1 bg-white shadow-sm">
643: //              <TabsTrigger value="patient-info">Patient Info</TabsTrigger>
644: //              <TabsTrigger value="content">Content</TabsTrigger>
645: //              <TabsTrigger value="settings">Settings</TabsTrigger>
646: //              <TabsTrigger value="diet">Diet Analysis</TabsTrigger>
647: //              <TabsTrigger value="genes">Gene Results</TabsTrigger>
648: //              <TabsTrigger value="summaries">Summaries</TabsTrigger>
649: //              <TabsTrigger value="import-export">Import/Export</TabsTrigger>
650: //              <TabsTrigger value="more">More Sections</TabsTrigger>
651: //            </TabsList>
652: //          </ScrollArea>
653:
654: //          {/* Patient Info Tab */}
655: //          <TabsContent value="patient-info">

```

```

656: //         <PatientInfoAdmin patientInfo={reportData.patientInfo} updatePatientInfo={updatePatientInfo}
657: //     </TabsContent>
658:
659: //     {/* Content Tab */}
660: //     <TabsContent value="content">
661: //         <ContentAdmin content={reportData.content} updateContent={updateContent} />
662: //     </TabsContent>
663:
664: //     {/* Settings Tab */}
665: //     <TabsContent value="settings">
666: //         <SettingsAdmin settings={reportData.settings} updateSettings={updateSettings} />
667: //     </TabsContent>
668:
669: //     {/* Diet Analysis Tab */}
670: //     <TabsContent value="diet">
671: //         <DietAnalysisAdmin dietAnalysis={reportData.dietAnalysis} updateDietAnalysis={updateDietAnal}
672: //     </TabsContent>
673:
674: //     {/* Gene Results Tab */}
675: //     <TabsContent value="genes">
676: //         <GeneResultsAdmin
677: //             geneTestResults={reportData.geneTestResults}
678: //             addGeneTestResult={addGeneTestResult}
679: //             updateGeneTestResult={updateGeneTestResult}
680: //             removeGeneTestResult={removeGeneTestResult}
681: //         />
682: //     </TabsContent>
683:
684: //     {/* Summaries Tab */}
685: //     <TabsContent value="summaries">
686: //         <SummariesAdmin summaries={reportData.summaries} updateSummaries={updateSummaries} />
687: //     </TabsContent>
688:
689: //     {/* Import/Export Tab */}
690: //     <TabsContent value="import-export">
691: //         <ImportExportAdmin reportData={reportData} setReportData={setReportData} />
692: //     </TabsContent>
693:
694: //     {/* More Sections Tab */}
695: //     <TabsContent value="more">
696: //         <div className="bg-white rounded-lg shadow-lg p-8 text-center">
697: //             <div className="text-6xl mb-4">?</div>
698: //             <h2 className="text-2xl font-bold text-gray-800 mb-4">Additional Sections Coming Soon</h2>
699: //             <p className="text-gray-600 mb-6">
700: //                 More admin panels for Nutrition, Sports & Fitness, Lifestyle Conditions, Metabolic Core,
701: //                 sections will be added here.
702: //             </p>
703: //             <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-4 text-sm">
704: //                 <div className="bg-blue-50 p-4 rounded-lg">
705: //                     <h4 className="font-semibold text-blue-800">Nutrition Admin</h4>
706: //                     <p className="text-blue-600">Vitamins, elements, supplements</p>
707: //                 </div>
708: //                 <div className="bg-green-50 p-4 rounded-lg">
709: //                     <h4 className="font-semibold text-green-800">Sports & Fitness Admin</h4>
710: //                     <p className="text-green-600">Exercise types, performance</p>
711: //                 </div>
712: //                 <div className="bg-purple-50 p-4 rounded-lg">
713: //                     <h4 className="font-semibold text-purple-800">Metabolic Core Admin</h4>
714: //                     <p className="text-purple-600">Gene analysis, impacts</p>
715: //                 </div>
716: //                 <div className="bg-orange-50 p-4 rounded-lg">
717: //                     <h4 className="font-semibold text-orange-800">Lifestyle Conditions</h4>
718: //                     <p className="text-orange-600">Health conditions, risks</p>
719: //                 </div>
720: //                 <div className="bg-pink-50 p-4 rounded-lg">
721: //                     <h4 className="font-semibold text-pink-800">Sleep & Allergies</h4>
722: //                     <p className="text-pink-600">Sleep patterns, sensitivities</p>
723: //                 </div>
724: //                 <div className="bg-indigo-50 p-4 rounded-lg">
725: //                     <h4 className="font-semibold text-indigo-800">Preventive Health</h4>
726: //                     <p className="text-indigo-600">Tests, supplements</p>
727: //                 </div>
728: //             </div>

```

```

729: //          </div>
730: //          </TabsContent>
731: //          </Tabs>
732: //        </div>
733: //      </div>
734: //    )
735: // }

```

■ 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 { v4 as uuidv4 } from "uuid"; // Import uuid for default data generation
8: import type {
9:   ComprehensiveReportData,
10:   PatientInfo,
11:   ReportContent,
12:   ReportSettings,
13:   // Removed DietAnalysis
14:   NutritionData,
15:   NutrientData,
16:   GeneTestResult,
17:   ReportSummaries,
18:   SportsAndFitness,
19:   ExerciseData,
20:   LifestyleConditions,
21:   HealthConditionStatus,
22:   MetabolicCore,
23:   MetabolicGeneData,
24:   DigestiveHealth,
25:   DigestiveHealthData,
26:   GenesAndAddiction,
27:   AddictionData,
28:   SleepAndRest,
29:   SleepData,
30:   AllergiesAndSensitivity,
31:   AllergyData,
32:   PreventiveHealth,
33:   FamilyGeneticImpact,
34:   DynamicDietFieldDefinition, // NEW
35:   PatientDietAnalysisResult, // NEW
36: } from "@types/report-types";
37:
38: // Corrected imports for admin components
39: import PatientInfoAdmin from "@components/admin/patient-info-admin";
40: import ContentAdmin from "@components/admin/content-admin";
41: import SettingsAdmin from "@components/admin/settings-admin";
42: // Renamed DietAnalysisAdmin to DynamicDietFieldAdmin
43: import DynamicDietFieldAdmin from "@components/admin/dynamic-diet-field-admin";
44: import GeneResultsAdmin from "@components/admin/gene-results-admin";
45: import SummariesAdmin from "@components/admin/summaries-admin";
46: import ImportExportAdmin from "@components/admin/import-export-admin";
47: import NutritionAdmin from "@components/admin/nutrition-admin";
48: import SportsFitnessAdmin from "@components/admin/sports-fitness-admin";
49: import LifestyleConditionsAdmin from "@components/admin/lifestyle-conditions-admin";
50: import DigestiveHealthAdmin from "@components/admin/digestive-health-admin";
51: import GenesAddictionAdmin from "@components/admin/genes-addiction-admin";
52: import SleepRestAdmin from "@components/admin/sleep-rest-admin";
53: import AllergiesSensitivityAdmin from "@components/admin/allergies-sensitivity-admin";
54: import PreventiveHealthAdmin from "@components/admin/preventive-health-admin";
55: import FamilyGeneticImpactAdmin from "@components/admin/family-genetic-impact-admin";
56: import PDFGenerator from "@components/admin/pdf-generator";
57: import MetabolicCoreAdmin from "@components/admin/metabolic-core-admin";
58:
59: const getDefaultReportData = (): ComprehensiveReportData => ({
60:   patientInfo: {
61:     name: "John Doe",

```

```

62:     gender: "MALE",
63:     birthDate: "1990-01-01",
64:     sampleCode: "DNL1000000110",
65:     sampleDate: "2023-01-15",
66:     reportDate: "2023-02-01",
67:     reportAuth: "Dr. Jane Smith, PhD",
68:     genomicAnalytics: "Genomics Lab Inc.",
69:     checkedBy: "Dr. Alice Brown",
70:     scientificContent: "Dr. Robert White",
71:     disclaimer: "This for informational purposes only and should not be used as a substitute for professional",
72: },
73: content: {
74:     introduction:
75:         "Welcome to your personalized nutrigenomics report. This document provides insights into how your genes",
76:     genomicsExplanation:
77:         "Genomics is the study of an organism's complete set of DNA, including all of its genes. It helps us",
78:     genesHealthImpact:
79:         "Your genes play a significant role in determining your health predispositions. This report highlights",
80:     fundamentalsPRS:
81:         "Polygenic Risk Scores (PRS) combine the effects of many genetic variants to estimate an individual's",
82:     utilityDoctors:
83:         "This report is designed to be a valuable tool for doctors and dietitians, providing actionable genet",
84:     microarrayExplanation:
85:         "Microarray technology allows for the simultaneous measurement of the expression levels of thousands",
86:     microarrayData:
87:         "Your microarray data has been analyzed to identify specific genetic variations relevant to your heal",
88: },
89: settings: {
90:     title: "Comprehensive Health Report",
91:     subtitle: "A Detailed Analysis of Your Genetics",
92:     companyName: "Health Solutions Inc.",
93:     headerColor: "#2C3E50",
94:     accentColor: "#E74C3C",
95:     fonts: {
96:         primary: "Helvetica, Arial, sans-serif",
97:         secondary: "Georgia, serif",
98:         mono: "Courier New, monospace",
99:     },
100: },
101: // NEW: Default dynamic diet field definitions
102: dynamicDietFieldDefinitions: [
103:     {
104:         _uuid: uuidv4(),
105:         id: "carbohydrate_sensitivity",
106:         label: "Carbohydrate Sensitivity",
107:         category: "Macronutrients",
108:         min: 1,
109:         max: 10,
110:         highRecommendation: "Reduce carbohydrate intake, focus on complex carbs.",
111:         normalRecommendation: "Maintain balanced carbohydrate intake.",
112:         lowRecommendation: "Ensure adequate carbohydrate intake.",
113:     },
114:     {
115:         _uuid: uuidv4(),
116:         id: "fat_sensitivity",
117:         label: "Fat Sensitivity",
118:         category: "Macronutrients",
119:         min: 1,
120:         max: 10,
121:         highRecommendation:
122:             "Reduce saturated and trans fats; focus on healthy fats.",
123:         normalRecommendation: "Maintain balanced fat intake.",
124:         lowRecommendation: "Ensure adequate healthy fat intake.",
125:     },
126:     {
127:         _uuid: uuidv4(),
128:         id: "protein_requirement",
129:         label: "Protein Requirement",
130:         category: "Macronutrients",
131:         min: 1,
132:         max: 10,
133:         highRecommendation: "Increased protein intake recommended.",
134:         normalRecommendation: "Ensure adequate protein intake.",

```

```

135:         lowRecommendation:
136:             "Consider increasing protein intake for muscle maintenance.",
137:     },
138:     {
139:         _uuid: uuidv4(),
140:         id: "meal_frequency",
141:         label: "Meal Frequency",
142:         category: "Meal Pattern",
143:         min: 1,
144:         max: 10,
145:         highRecommendation: "Consider smaller, more frequent meals.",
146:         normalRecommendation: "Maintain regular meal frequency.",
147:         lowRecommendation: "Consider fewer, larger meals.",
148:     },
149:     {
150:         _uuid: uuidv4(),
151:         id: "alcohol_sensitivity",
152:         label: "Alcohol Sensitivity",
153:         category: "Food Sensitivities",
154:         min: 1,
155:         max: 10,
156:         highRecommendation:
157:             "Limit alcohol consumption due to genetic sensitivity.",
158:         normalRecommendation: "Moderate alcohol intake is fine.",
159:         lowRecommendation: "Normal alcohol metabolism.",
160:     },
161:     {
162:         _uuid: uuidv4(),
163:         id: "caffeine_sensitivity",
164:         label: "Caffeine Sensitivity",
165:         category: "Food Sensitivities",
166:         min: 1,
167:         max: 10,
168:         highRecommendation: "Limit caffeine intake due to slow metabolism.",
169:         normalRecommendation:
170:             "Normal caffeine metabolism; moderate intake is fine.",
171:         lowRecommendation:
172:             "Fast caffeine metabolism; higher intake may be tolerated.",
173:     },
174: ],
175: // NEW: Default patient-specific results for dynamic diet fields
176: patientDietAnalysisResults: [
177:     {
178:         fieldId: "carbohydrate_sensitivity",
179:         score: 5,
180:         level: "NORMAL",
181:         recommendation: "Maintain balanced carbohydrate intake.",
182:     },
183:     {
184:         fieldId: "fat_sensitivity",
185:         score: 7,
186:         level: "HIGH",
187:         recommendation: "Reduce saturated and trans fats; focus on healthy fats.",
188:     },
189:     {
190:         fieldId: "protein_requirement",
191:         score: 3,
192:         level: "LOW",
193:         recommendation: "Ensure adequate protein intake for muscle maintenance.",
194:     },
195:     {
196:         fieldId: "meal_frequency",
197:         score: 6,
198:         level: "FREQUENT",
199:         recommendation: "Consider smaller, more frequent meals.",
200:     },
201:     {
202:         fieldId: "alcohol_sensitivity",
203:         score: 9,
204:         level: "HIGH",
205:         recommendation: "Limit alcohol consumption due to genetic sensitivity.",
206:     },
207:     {

```

```

208:         fieldId: "caffeine_sensitivity",
209:         score: 2,
210:         level: "LOW",
211:         recommendation: "Normal caffeine metabolism; moderate intake is fine.",
212:     },
213: ],
214: nutritionData: {
215:     vitamins: {
216:         vitaminA: {
217:             score: 3,
218:             healthImpact: "Skin & Vision",
219:             intakeLevel: "ENHANCED INTAKE",
220:             source: "DIET",
221:         },
222:         vitaminB12: {
223:             score: 6,
224:             healthImpact: "Nervous System & Bones",
225:             intakeLevel: "ENHANCED INTAKE (METHYLCOBALAMIN)",
226:             source: "DIET",
227:         },
228:         vitaminB2: {
229:             score: 3,
230:             healthImpact: "Energy Metabolism",
231:             intakeLevel: "ENHANCED INTAKE",
232:             source: "DIET",
233:         },
234:         vitaminB6: {
235:             score: 6,
236:             healthImpact: "Tissue growth repair",
237:             intakeLevel: "ENHANCED INTAKE",
238:             source: "DIET",
239:         },
240:         vitaminB9: {
241:             score: 3,
242:             healthImpact: "RBC development & Heart health",
243:             intakeLevel: "ENHANCED INTAKE (L METHYLFOLATE)",
244:             source: "DIET",
245:         },
246:         vitaminC: {
247:             score: 3,
248:             healthImpact: "Skin & Immunity",
249:             intakeLevel: "ENHANCED INTAKE",
250:             source: "DIET",
251:         },
252:         vitaminD: {
253:             score: 6,
254:             healthImpact: "Bones & Diabetes",
255:             intakeLevel: "ENHANCED INTAKE & SUN EXPOSURE",
256:             source: "DIET & SUN EXPOSURE",
257:         },
258:         vitaminE: {
259:             score: 5,
260:             healthImpact: "Skin & Immune system",
261:             intakeLevel: "ENHANCED INTAKE",
262:             source: "DIET",
263:         },
264:     },
265:     fattyAcids: {
266:         omegaFattyAcids: {
267:             score: 5,
268:             healthImpact: "Cardiac health & Triglycerides",
269:             intakeLevel: "ENHANCED INTAKE (VEG SOURCES)",
270:             source: "ENHANCED INTAKE (VEG/FISH OIL) & DIET",
271:         },
272:     },
273:     elements: {
274:         calcium: {
275:             score: 3,
276:             healthImpact: "Bone health & Nerve function",
277:             intakeLevel: "ENHANCED INTAKE",
278:             source: "DIET",
279:         },
280:         iodine: {

```



```

281:         score: 3,
282:         healthImpact: "Detoxification & Thyroid hormones",
283:         intakeLevel: "ENHANCED INTAKE",
284:         source: "DIET",
285:     },
286:     iron: {
287:         score: 3,
288:         healthImpact: "Haemoglobin & Energy",
289:         intakeLevel: "RESTRICTED INTAKE",
290:         source: "DIET",
291:     },
292:     magnesium: {
293:         score: 3,
294:         healthImpact: "Heart health & Immunity",
295:         intakeLevel: "ENHANCED INTAKE",
296:         source: "DIET",
297:     },
298:     phosphorus: {
299:         score: 3,
300:         healthImpact: "Bone health & Kidney function",
301:         intakeLevel: "ENHANCED INTAKE",
302:         source: "DIET",
303:     },
304:     selenium: {
305:         score: 3,
306:         healthImpact: "Cognitive & Muscle function",
307:         intakeLevel: "ENHANCED INTAKE",
308:         source: "DIET",
309:     },
310:     zinc: {
311:         score: 3,
312:         healthImpact: "Fertility & Skin",
313:         intakeLevel: "ENHANCED INTAKE",
314:         source: "DIET",
315:     },
316: },
317: complexNutrients: {
318:     antiInflammatory: {
319:         score: 5,
320:         healthImpact: "Pain control & Organ health",
321:         intakeLevel: "ENHANCED INTAKE",
322:         source: "DIET",
323:     },
324:     antioxidants: {
325:         score: 6,
326:         healthImpact: "Skin & Immune system",
327:         intakeLevel: "ENHANCED INTAKE",
328:         source: "DIET",
329:     },
330: },
331: },
332: sportsAndFitness: {
333:     exerciseType: {
334:         endurancePotential: {
335:             level: "High Endurance Potential",
336:             description:
337:                 "Excellent genetic predisposition for endurance activities",
338:         },
339:         powerPotential: {
340:             level: "Average Power Potential",
341:             description: "Moderate power output capacity",
342:         },
343:         strengthProfile: {
344:             level: "High Strength Profile",
345:             description: "Good potential for strength training",
346:         },
347:         exerciseTime: {
348:             level: "Evening time is more beneficial than morning",
349:             description: "MORNING Vs. EVENING",
350:         },
351:     },
352:     performance: {
353:         enduranceCapacity: {

```

```

354:         level: "High Endurance Potential",
355:         description: "Superior aerobic capacity",
356:     },
357:     fatigueResistance: {
358:         level: "Normal fatigue resistance. Average endurance capacity",
359:         description: "Average endurance capacity",
360:     },
361:     bloodFlow: {
362:         level: "Reduced vascular efficiency, affects endurance sports.",
363:         description: "Affects endurance sports",
364:     },
365:     oxygenEfficiency: {
366:         level: "Normal VO2 Max. Need practice to excel in endurance sports",
367:         description: "Need practice to excel",
368:     },
369:     waterLoss: {
370:         level:
371:             "Increased water and electrolyte loss tendency. Focus on Hydration and electrolyte management dur
372:         description: "Focus on hydration",
373:     },
374:     lactateClearing: {
375:         level: "High lactate clearance. Beneficial for Endurance sports.",
376:         description: "Beneficial for endurance",
377:     },
378:     recoveryEfficiency: {
379:         level:
380:             "High recovery efficiency. Enhanced sports potential & reduced injury risk",
381:         description: "Enhanced sports potential",
382:     },
383:     bodyComposition: {
384:         level: "Low fat accumulation. Increased muscle tendency.",
385:         description: "Increased muscle tendency",
386:     },
387:     injuryRisk: {
388:         level:
389:             "High injury tendency. Thorough injury management steps to be taken.",
390:         description: "Thorough injury management needed",
391:     },
392:     achillesTendonInjury: {
393:         level: "Low injury tendency. Ensure adequate warmup before exercises",
394:         description: "Ensure adequate warmup",
395:     },
396:     ligamentInjury: {
397:         level:
398:             "Low ACL injury tendency. Ensure adequate warmup before exercises",
399:         description: "Ensure adequate warmup",
400:     },
401:     rupture: {
402:         level: "Low rupture tendency. Ensure adequate warmup before exercises",
403:         description: "Ensure adequate warmup",
404:     },
405:     muscleCramps: {
406:         level:
407:             "Low risk of muscle cramps. Ensure adequate warmup before exercises",
408:         description: "Ensure adequate warmup",
409:     },
410: },
411: },
412: lifestyleConditions: {
413:     heartVascularHealth: {
414:         hypertension: { status: "strength" },
415:         leftVentricularHypertrophy: { status: "strength" },
416:         atherosclerosis: { status: "improvement" },
417:         familialHypercholesterolemia: { status: "strength" },
418:         cardiomyopathy: { status: "strength" },
419:         cholesterolImbalance: { status: "improvement" },
420:         stroke: { status: "improvement" },
421:     },
422:     diabetes: {
423:         obesity: { status: "improvement" },
424:         diabetesType2: { status: "improvement" },
425:     },
426:     liverHealth: {

```

```

427:     nafld: { status: "improvement" },
428:     liverCirrhosis: { status: "strength" },
429: },
430: boneHealth: {
431:     osteoporosis: { status: "improvement" },
432:     osteoarthritis: { status: "improvement" },
433: },
434: gutHealth: {
435:     irritableBowelSyndrome: { status: "improvement" },
436:     celiacDisease: { status: "strength" },
437: },
438: hormoneSystem: {
439:     hypothyroidism: { status: "improvement" },
440: },
441: },
442: metabolicCore: {
443:     methylation: {
444:         genes: ["MTHFR"],
445:         genotype: "C/T",
446:         impact: "Reduced folate metabolism",
447:         advice: "Consider L-methylfolate supplementation.",
448:     },
449:     inflammation: {
450:         genes: ["CRP"],
451:         genotype: "G/A",
452:         impact: "Increased inflammatory response",
453:         advice: "Incorporate anti-inflammatory foods.",
454:     },
455:     oxidativeStress: {
456:         genes: ["SOD2"],
457:         genotype: "T/C",
458:         impact: "Reduced antioxidant defense",
459:         advice: "Increase antioxidant-rich foods and supplements.",
460:     },
461:     detoxification: {
462:         genes: ["GSTP1"],
463:         genotype: "A/G",
464:         impact: "Impaired detoxification capacity",
465:         advice: "Support liver health with cruciferous vegetables.",
466:     },
467:     vitaminDMetabolism: {
468:         genes: ["VDR"],
469:         genotype: "C/C",
470:         impact: "Reduced Vitamin D absorption",
471:         advice: "Ensure adequate Vitamin D intake and sun exposure.",
472:     },
473:     lipidMetabolism: {
474:         genes: ["APOE"],
475:         genotype: "E3/E4",
476:         impact: "Increased LDL cholesterol tendency",
477:         advice: "Monitor lipid levels and follow a heart-healthy diet.",
478:     },
479:     insulinResistance: {
480:         genes: ["TCF7L2"],
481:         genotype: "T/T",
482:         impact: "Higher risk of insulin resistance",
483:         advice: "Focus on low glycemic index foods and regular exercise.",
484:     },
485:     circadianRhythm: {
486:         genes: ["CLOCK"],
487:         genotype: "C/G",
488:         impact: "Altered sleep-wake cycle",
489:         advice:
490:             "Maintain consistent sleep schedule and optimize sleep environment.",
491:     },
492:     exerciseResponsiveness: {
493:         genes: ["ACTN3"],
494:         genotype: "R/X",
495:         impact: "Enhanced power and strength response to training",
496:         advice: "Incorporate strength and power training into routine.",
497:     },
498:     snackingTendency: {
499:         genes: ["FTO"],

```

```

500:     genotype: "A/A",
501:     impact: "Increased tendency for snacking and higher BMI",
502:     advice: "Practice mindful eating and portion control.",
503: },
504: oestrogenElimination: {
505:     genes: ["COMT"],
506:     genotype: "G/A",
507:     impact: "Slower estrogen elimination",
508:     advice: "Support liver detoxification pathways.",
509: },
510: powerVsEndurance: {
511:     genes: ["ACE"],
512:     genotype: "I/D",
513:     impact: "Balanced power and endurance potential",
514:     advice: "Vary training to include both power and endurance exercises.",
515: },
516: injuryRecovery: {
517:     genes: ["COL1A1"],
518:     genotype: "G/T",
519:     impact: "Slower collagen repair, increased injury risk",
520:     advice: "Focus on adequate protein and Vitamin C for tissue repair.",
521: },
522: fluidLossTendency: {
523:     genes: ["AGT"],
524:     genotype: "T/C",
525:     impact: "Increased fluid and electrolyte loss during exercise",
526:     advice: "Ensure proper hydration and electrolyte replenishment.",
527: },
528: vo2Max: {
529:     genes: ["PPARGC1A"],
530:     genotype: "G/G",
531:     impact: "High VO2 Max potential",
532:     advice: "Optimize cardiovascular training for peak performance.",
533: },
534: },
535: digestiveHealth: {
536:     gutHealth: {
537:         level: "NORMAL GUT HEALTH",
538:         description: "Good overall digestive function",
539:     },
540:     glutenIntolerance: {
541:         level: "LOW INTOLERANCE",
542:         description: "CONDITIONAL GLUTEN ELIMINATION NEEDED",
543:     },
544:     celiacDiseaseTendency: {
545:         level: "LOW TENDENCY",
546:         description: "Low genetic risk for celiac disease",
547:     },
548:     impactOnBonesJoints: {
549:         level: "NO IMPACT ON BONE HEALTH",
550:         description: "Digestive health does not affect bone health",
551:     },
552:     lactoseIntolerance: {
553:         level: "HIGH INTOLERANCE",
554:         description: "LACTOSE ELIMINATION NEEDED",
555:     },
556:     probioticsRequirement: {
557:         level: "PROBIOTICS NEEDED",
558:         description: "FOOD & SUPPLEMENTS",
559:     },
560: },
561: genesAndAddiction: {
562:     alcoholAddiction: {
563:         tendency: "LOW",
564:         description: "Low genetic predisposition to alcohol dependency",
565:     },
566:     smokingAddiction: {
567:         tendency: "NORMAL",
568:         description: "Average nicotine addiction susceptibility",
569:     },
570:     snackingHabits: {
571:         tendency: "HIGH",
572:         description: "Genetic tendency toward frequent snacking",

```

```

573:     },
574:     altruism: {
575:         tendency: "HIGH",
576:         description: "Strong genetic predisposition toward helping others",
577:     },
578: },
579: sleepAndRest: {
580:     sleepCycle: {
581:         impact: "DELAYED SLEEP CYCLE",
582:         intervention: "EXERCISE IN THE EVENING AVOID CARBS DURING BREAKFAST",
583:     },
584:     sleepApnea: {
585:         impact: "INCREASED CHANCES OF SLEEP APNEA",
586:         intervention: "EXERCISE REGULARLY SEEK MEDICAL ADVICE IF ISSUE PERSISTS",
587:     },
588:     sleepDepth: {
589:         impact: "LOW SLEEP DEPTH",
590:         intervention: "CONSUME CURD RICE IN DINNER FOR BETTER SLEEP",
591:     },
592:     stressSensitivity: {
593:         impact: "LOW STRESS SENSITIVITY",
594:         intervention: "EXERCISE REGULARLY FOR A BETTER SLEEP CYCLE",
595:     },
596: },
597: allergiesAndSensitivity: {
598:     dustAllergy: { tendency: "NORMAL" },
599:     pesticideSensitivity: { tendency: "HIGH" },
600:     smokeSensitivity: { tendency: "HIGH" },
601:     automobileSmokeSensitivity: { tendency: "HIGH" },
602:     generalAdvice:
603:         "THE BEST WAY TO REDUCE ALLERGIES IS TO STAY AWAY FROM THE SOURCE, NO ROCKET SCIENCE HOWEVER THE FOLLOWING",
604: },
605: geneTestResults: [
606:     {
607:         geneName: "ALDH2 Aldehyde dehydrogenase 2",
608:         variation: "G>A",
609:         result: "GG",
610:     },
611:     {
612:         geneName: "MTHFR Methylenetetrahydrofolate reductase",
613:         variation: "C>T",
614:         result: "CT",
615:     },
616: ],
617: metabolicSummary: {
618:     strengths: ["Efficient detoxification", "Normal lipid metabolism"],
619:     weaknesses: [
620:         "Increased oxidative stress",
621:         "Higher risk of insulin resistance",
622:     ],
623: },
624: preventiveHealth: {
625:     diagnosticTests: {
626:         halfYearly: ["PLASMA SUGAR (FASTING & PP)", "VITAMIN D"],
627:         yearly: [
628:             "HbA1c",
629:             "LIPID PROFILE",
630:             "SERUM HOMOCYSTEINE",
631:             "VITAMIN B12",
632:             "LIVER PROFILE",
633:         ],
634:     },
635:     nutritionalSupplements: [
636:         { supplement: "VITAMIN D", needed: true },
637:         { supplement: "METHYL FOLATE + METHYL COBALAMINE + B6", needed: true },
638:         { supplement: "OMEGA FATTY ACIDS (FISH OIL)", needed: true },
639:         { supplement: "FOLIC ACID + COBALAMINE + B6", needed: true },
640:         { supplement: "OMEGA FATTY ACIDS (VEG)", needed: true },
641:         { supplement: "VITAMIN E", needed: true },
642:         { supplement: "PROBIOTICS", needed: true },
643:     ],
644: },
645: familyGeneticImpact: [

```

```

646:     {
647:       gene: "MnSOD",
648:       normalAlleles: "AA",
649:       yourResult: "AG",
650:       healthImpact:
651:         "INCREASED TENDENCY OF OXIDATIVE STRESS NEED OF ANTI-OXIDANT SUPPLEMENTS",
652:     },
653:     {
654:       gene: "FTO",
655:       normalAlleles: "TT",
656:       yourResult: "AT",
657:       healthImpact:
658:         "INCREASED TENDENCY OF INSULIN RESISTANCE WEIGHT AND CALORIE MANAGEMENT IMPORTANT TO MANAGE AND REDU",
659:     },
660:   ],
661:   summaries: {
662:     nutrigenomicsSummary:
663:       "Based on the overall nutrigenomics analysis, provide expert summary for personalized diet and lifestyle",
664:     exerciseGenomicsSummary:
665:       "Based on the overall genomics analysis, provide expert summary for personalized exercise plan...",
666:   },
667: });
668:
669: const AdminPage = () => {
670:   const [reportData, setReportData] =
671:     useState<ComprehensiveReportData>(getDefaultReportData);
672:   const { toast } = useToast();
673:
674:   // Load data from API on component mount
675:   useEffect(() => {
676:     const fetchData = async () => {
677:       try {
678:         const response = await fetch("/api/report-data");
679:         if (response.ok) {
680:           const data = await response.json();
681:           setReportData(data);
682:           toast({
683:             title: "Data loaded from server!",
684:             description: "Report data has been loaded from the backend.",
685:           });
686:         } else {
687:           toast({
688:             title: "Failed to load data",
689:             description:
690:               "Could not load report data from the server. Using default data.",
691:             variant: "destructive",
692:           });
693:         }
694:       } catch (error) {
695:         console.error("Error fetching report data:", error);
696:         toast({
697:           title: "Error fetching data",
698:           description:
699:             "An error occurred while fetching data from the server. Using default data.",
700:           variant: "destructive",
701:         });
702:       }
703:     };
704:     fetchData();
705:   }, []);
706:
707:   const saveReportData = async () => {
708:     try {
709:       const response = await fetch("/api/report-data", {
710:         method: "POST",
711:         headers: {
712:           "Content-Type": "application/json",
713:         },
714:         body: JSON.stringify(reportData),
715:       });
716:
717:       if (response.ok) {
718:         toast({

```

```

719:         title: "Report data saved!",
720:         description: "Your progress has been saved to the backend.",
721:     });
722:     } else {
723:         const errorData = await response.json();
724:         toast({
725:             title: "Failed to save data",
726:             description:
727:                 errorData.error || "An error occurred while saving data.",
728:             variant: "destructive",
729:         });
730:     }
731: } catch (error) {
732:     console.error("Error saving report data:", error);
733:     toast({
734:         title: "Error saving data",
735:         description: "An unexpected error occurred while saving data.",
736:         variant: "destructive",
737:     });
738: }
739: };
740:
741: const resetSection = (section: keyof ComprehensiveReportData) => {
742:     setReportData((prev) => {
743:         const defaultData = getDefaultReportData();
744:         return { ...prev, [section]: defaultData[section] };
745:     });
746:     toast({
747:         title: `${section} reset!`,
748:         description: `The ${section} section has been reset to default values.`,
749:     });
750: };
751:
752: const updatePatientInfo = (field: keyof PatientInfo, value: string) => {
753:     setReportData((prev) => ({
754:         ...prev,
755:         patientInfo: { ...prev.patientInfo, [field]: value },
756:     }));
757: };
758:
759: const updateReportContent = (field: keyof ReportContent, value: string) => {
760:     setReportData((prev) => ({
761:         ...prev,
762:         content: { ...prev.content, [field]: value },
763:     }));
764: };
765:
766: const updateReportSettings = (field: keyof ReportSettings, value: string) => {
767:     setReportData((prev) => ({
768:         ...prev,
769:         settings: { ...prev.settings, [field]: value },
770:     }));
771: };
772:
773: // NEW: Update function for dynamicDietFieldDefinitions
774: const updateDynamicDietFieldDefinitions = (
775:     definitions: DynamicDietFieldDefinition[]
776: ) => {
777:     setReportData((prev) => ({
778:         ...prev,
779:         dynamicDietFieldDefinitions: definitions,
780:     }));
781: };
782:
783: // NEW: Update function for patientDietAnalysisResults
784: const updatePatientDietAnalysisResult = (
785:     fieldId: string,
786:     data: Partial<PatientDietAnalysisResult>
787: ) => {
788:     setReportData((prev) => {
789:         const existingIndex = prev.patientDietAnalysisResults.findIndex(
790:             (r) => r.fieldId === fieldId
791:         );

```

```

792:     const newResults = [...prev.patientDietAnalysisResults];
793:
794:     if (existingIndex > -1) {
795:         newResults[existingIndex] = { ...newResults[existingIndex], ...data };
796:     } else {
797:         // If result for this fieldId doesn't exist, create a new one
798:         newResults.push({
799:             fieldId,
800:             score: 0,
801:             level: "NORMAL",
802:             recommendation: "",
803:             ...data,
804:         });
805:     }
806:     return { ...prev, patientDietAnalysisResults: newResults };
807: });
808: };
809:
810: const updateNutritionData = (
811:     category: keyof NutritionData,
812:     nutrient: string,
813:     field: keyof NutrientData,
814:     value: string | number
815: ) => {
816:     setReportData((prev) => ({
817:         ...prev,
818:         nutritionData: {
819:             ...prev.nutritionData,
820:             [category]: {
821:                 ...prev.nutritionData[category],
822:                 [nutrient]: {
823:                     ...prev.nutritionData[category][
824:                         nutrient as keyof (typeof prev.nutritionData)[typeof category]
825:                     ],
826:                     [field]: value,
827:                 },
828:             },
829:         },
830:     }));
831: };
832:
833: const updateSportsAndFitness = (
834:     section: keyof SportsAndFitness,
835:     field: string,
836:     data: Partial<ExerciseData>
837: ) => {
838:     setReportData((prev) => ({
839:         ...prev,
840:         sportsAndFitness: {
841:             ...prev.sportsAndFitness,
842:             [section]: {
843:                 ...prev.sportsAndFitness[section],
844:                 [field]: {
845:                     ...prev.sportsAndFitness[section][
846:                         field as keyof (typeof prev.sportsAndFitness)[typeof section]
847:                     ],
848:                     ...data,
849:                 },
850:             },
851:         },
852:     }));
853: };
854:
855: const updateLifestyleCondition = (
856:     section: keyof LifestyleConditions,
857:     field: string,
858:     status: HealthConditionStatus["status"]
859: ) => {
860:     setReportData((prev) => ({
861:         ...prev,
862:         lifestyleConditions: {
863:             ...prev.lifestyleConditions,
864:             [section]: {

```



```

865:         ...prev.lifestyleConditions[section],
866:         [field]: {
867:             ...prev.lifestyleConditions[section][
868:                 field as keyof (typeof prev.lifestyleConditions)[typeof section]
869:             ],
870:             status: status,
871:         },
872:     },
873: },
874: }));
875: };
876:
877: const updateMetabolicCore = (
878:     field: keyof MetabolicCore,
879:     data: Partial<MetabolicGeneData>
880: ) => {
881:     setReportData((prev) => ({
882:         ...prev,
883:         metabolicCore: {
884:             ...prev.metabolicCore,
885:             [field]: {
886:                 ...prev.metabolicCore[field],
887:                 ...data,
888:             },
889:         },
890:     }));
891: };
892:
893: const updateDigestiveHealth = (
894:     field: keyof DigestiveHealth,
895:     data: Partial<DigestiveHealthData>
896: ) => {
897:     setReportData((prev) => ({
898:         ...prev,
899:         digestiveHealth: {
900:             ...prev.digestiveHealth,
901:             [field]: {
902:                 ...prev.digestiveHealth[field],
903:                 ...data,
904:             },
905:         },
906:     }));
907: };
908:
909: const updateGenesAndAddiction = (
910:     field: keyof GenesAndAddiction,
911:     data: Partial<AddictionData>
912: ) => {
913:     setReportData((prev) => ({
914:         ...prev,
915:         genesAndAddiction: {
916:             ...prev.genesAndAddiction,
917:             [field]: {
918:                 ...prev.genesAndAddiction[field],
919:                 ...data,
920:             },
921:         },
922:     }));
923: };
924:
925: const updateSleepAndRest = (
926:     field: keyof SleepAndRest,
927:     data: Partial<SleepData>
928: ) => {
929:     setReportData((prev) => ({
930:         ...prev,
931:         sleepAndRest: {
932:             ...prev.sleepAndRest,
933:             [field]: {
934:                 ...prev.sleepAndRest[field],
935:                 ...data,
936:             },
937:         },

```

```

938:     }));
939: };
940:
941: const updateAllergiesAndSensitivity = (
942:   field: keyof AllergiesAndSensitivity,
943:   data: Partial<AllergyData> | string
944: ) => {
945:   setReportData((prev) => ({
946:     ...prev,
947:     allergiesAndSensitivity: {
948:       ...prev.allergiesAndSensitivity,
949:       [field]:
950:         typeof data === "string"
951:           ? data
952:           : {
953:               ...(prev.allergiesAndSensitivity[
954:                 field as Exclude<
955:                   keyof AllergiesAndSensitivity,
956:                   "generalAdvice"
957:                 >
958:               ] as AllergyData),
959:               ...data,
960:             },
961:     },
962:   }));
963: };
964:
965: const updatePreventiveHealth = (
966:   section: keyof PreventiveHealth,
967:   data: any
968: ) => {
969:   setReportData((prev) => ({
970:     ...prev,
971:     [section]: data,
972:   }));
973: };
974:
975: const addFamilyGeneticImpact = () => {
976:   setReportData((prev) => ({
977:     ...prev,
978:     familyGeneticImpact: [
979:       ...prev.familyGeneticImpact,
980:       { gene: "", normalAlleles: "", yourResult: "", healthImpact: "" },
981:     ],
982:   }));
983: };
984:
985: const updateFamilyGeneticImpact = (
986:   index: number,
987:   field: keyof FamilyGeneticImpact,
988:   value: string
989: ) => {
990:   setReportData((prev) => {
991:     const newImpacts = [...prev.familyGeneticImpact];
992:     newImpacts[index] = { ...newImpacts[index], [field]: value };
993:     return { ...prev, familyGeneticImpact: newImpacts };
994:   });
995: };
996:
997: const removeFamilyGeneticImpact = (index: number) => {
998:   setReportData((prev) => ({
999:     ...prev,
1000:     familyGeneticImpact: prev.familyGeneticImpact.filter(
1001:       (_, i) => i !== index
1002:     ),
1003:   }));
1004: };
1005:
1006: const updateGeneTestResult = (
1007:   index: number,
1008:   field: keyof GeneTestResult,
1009:   value: string
1010: ) => {

```

```

1011:     setReportData((prev) => {
1012:         const newGeneTests = [...prev.geneTestResults];
1013:         newGeneTests[index] = { ...newGeneTests[index], [field]: value };
1014:         return { ...prev, geneTestResults: newGeneTests };
1015:     });
1016: };
1017:
1018: const addGeneTestResult = () => {
1019:     setReportData((prev) => ({
1020:         ...prev,
1021:         geneTestResults: [
1022:             ...prev.geneTestResults,
1023:             { geneName: "", variation: "", result: "" },
1024:         ],
1025:     }));
1026: };
1027:
1028: const removeGeneTestResult = (index: number) => {
1029:     setReportData((prev) => ({
1030:         ...prev,
1031:         geneTestResults: prev.geneTestResults.filter((_, i) => i !== index),
1032:     }));
1033: };
1034:
1035: const updateReportSummaries = (
1036:     field: keyof ReportSummaries,
1037:     value: string
1038: ) => {
1039:     setReportData((prev) => ({
1040:         ...prev,
1041:         summaries: { ...prev.summaries, [field]: value },
1042:     }));
1043: };
1044:
1045: return (
1046:     <div className="container mx-auto py-10">
1047:         <h1 className="text-3xl font-bold mb-5">Admin Dashboard</h1>
1048:
1049:         <div className="mb-5 flex gap-2 no-print">
1050:             <Button onClick={saveReportData}>Save Data</Button>
1051:             <Button onClick={() => setReportData(getDefaultReportData())}>
1052:                 Reset All
1053:             </Button>
1054:         </div>
1055:
1056:         <Tabs defaultValue="patient-info" className="w-[100%]">
1057:             <div className="w-full overflow-x-auto scrollbar-hide">
1058:                 <TabList className="flex gap-2 min-w-max bg-white shadow-sm px-4 py-2 rounded-md border border-
1059:                     <TabTrigger value="patient-info">? Patient Info</TabTrigger>
1060:                     <TabTrigger value="content">? Content</TabTrigger>
1061:                     <TabTrigger value="settings">?? Settings</TabTrigger>
1062:                     <TabTrigger value="diet-fields">? Diet Fields</TabTrigger>
1063:                     <TabTrigger value="nutrition">? Nutrition</TabTrigger>
1064:                     <TabTrigger value="sports-fitness">
1065:                         ????? Sports & Fitness
1066:                     </TabTrigger>
1067:                     <TabTrigger value="lifestyle-conditions">? Lifestyle</TabTrigger>
1068:                     <TabTrigger value="metabolic-core">? Metabolic Core</TabTrigger>
1069:                     <TabTrigger value="digestive-health">
1070:                         ? Digestive Health
1071:                     </TabTrigger>
1072:                     <TabTrigger value="genes-addiction">
1073:                         ? Genes & Addiction
1074:                     </TabTrigger>
1075:                     <TabTrigger value="sleep-rest">? Sleep & Rest</TabTrigger>
1076:                     <TabTrigger value="allergies-sensitivity">
1077:                         ? Allergies
1078:                     </TabTrigger>
1079:                     <TabTrigger value="preventive-health">
1080:                         ?? Preventive Health
1081:                     </TabTrigger>
1082:                     <TabTrigger value="family-genetic-impact">
1083:                         ? Family Impact

```

```

1084:         </TabsTrigger>
1085:         <TabsTrigger value="genes">? Gene Results</TabsTrigger>
1086:         <TabsTrigger value="summaries">? Summaries</TabsTrigger>
1087:         <TabsTrigger value="pdf-export">? PDF Export</TabsTrigger>
1088:         <TabsTrigger value="import-export">? Import/Export</TabsTrigger>
1089:     </TabsList>
1090: </div>
1091:
1092: <TabsContent value="patient-info">
1093:     <PatientInfoAdmin
1094:         patientInfo={reportData.patientInfo}
1095:         updatePatientInfo={updatePatientInfo}
1096:         onSave={() => saveReportData()}
1097:         onReset={() => resetSection("patientInfo")}
1098:     />
1099: </TabsContent>
1100: <TabsContent value="content">
1101:     <ContentAdmin
1102:         content={reportData.content}
1103:         updateContent={updateReportContent}
1104:         onSave={() => saveReportData()}
1105:         onReset={() => resetSection("content")}
1106:     />
1107: </TabsContent>
1108: <TabsContent value="settings">
1109:     <SettingsAdmin
1110:         settings={reportData.settings}
1111:         updateSettings={updateReportSettings}
1112:         onSave={() => saveReportData()}
1113:         onReset={() => resetSection("settings")}
1114:     />
1115: </TabsContent>
1116: <TabsContent value="diet-fields">
1117:     { " " }
1118:     { /* Changed tab content value */ }
1119:     <DynamicDietFieldAdmin
1120:         dynamicDietFieldDefinitions={reportData.dynamicDietFieldDefinitions}
1121:         updateDynamicDietFieldDefinitions={
1122:             updateDynamicDietFieldDefinitions
1123:         }
1124:         onSave={() => saveReportData()}
1125:         onReset={() => resetSection("dynamicDietFieldDefinitions")}
1126:     />
1127: </TabsContent>
1128: <TabsContent value="nutrition">
1129:     <NutritionAdmin
1130:         nutritionData={reportData.nutritionData}
1131:         updateNutritionData={updateNutritionData}
1132:         onSave={() => saveReportData()}
1133:         onReset={() => resetSection("nutritionData")}
1134:     />
1135: </TabsContent>
1136: <TabsContent value="sports-fitness">
1137:     <SportsFitnessAdmin
1138:         sportsAndFitness={reportData.sportsAndFitness}
1139:         updateSportsAndFitness={updateSportsAndFitness}
1140:         onSave={() => saveReportData()}
1141:         onReset={() => resetSection("sportsAndFitness")}
1142:     />
1143: </TabsContent>
1144: <TabsContent value="lifestyle-conditions">
1145:     <LifestyleConditionsAdmin
1146:         lifestyleConditions={reportData.lifestyleConditions}
1147:         updateLifestyleCondition={updateLifestyleCondition}
1148:         onSave={() => saveReportData()}
1149:         onReset={() => resetSection("lifestyleConditions")}
1150:     />
1151: </TabsContent>
1152: <TabsContent value="metabolic-core">
1153:     <MetabolicCoreAdmin
1154:         metabolicCore={reportData.metabolicCore}
1155:         updateMetabolicCore={updateMetabolicCore}
1156:         onSave={() => saveReportData()}

```

```

1157:         onReset={() => resetSection("metabolicCore")}
1158:     />
1159: </TabsContent>
1160: <TabsContent value="digestive-health">
1161:     <DigestiveHealthAdmin
1162:         digestiveHealth={reportData.digestiveHealth}
1163:         updateDigestiveHealth={updateDigestiveHealth}
1164:         onSave={() => saveReportData()}
1165:         onReset={() => resetSection("digestiveHealth")}
1166:     />
1167: </TabsContent>
1168: <TabsContent value="genes-addiction">
1169:     <GenesAddictionAdmin
1170:         genesAndAddiction={reportData.genesAndAddiction}
1171:         updateGenesAndAddiction={updateGenesAndAddiction}
1172:         onSave={() => saveReportData()}
1173:         onReset={() => resetSection("genesAndAddiction")}
1174:     />
1175: </TabsContent>
1176: <TabsContent value="sleep-rest">
1177:     <SleepRestAdmin
1178:         sleepAndRest={reportData.sleepAndRest}
1179:         updateSleepAndRest={updateSleepAndRest}
1180:         onSave={() => saveReportData()}
1181:         onReset={() => resetSection("sleepAndRest")}
1182:     />
1183: </TabsContent>
1184: <TabsContent value="allergies-sensitivity">
1185:     <AllergiesSensitivityAdmin
1186:         allergiesAndSensitivity={reportData.allergiesAndSensitivity}
1187:         updateAllergiesAndSensitivity={updateAllergiesAndSensitivity}
1188:         onSave={() => saveReportData()}
1189:         onReset={() => resetSection("allergiesAndSensitivity")}
1190:     />
1191: </TabsContent>
1192: <TabsContent value="preventive-health">
1193:     <PreventiveHealthAdmin
1194:         preventiveHealth={reportData.preventiveHealth}
1195:         updatePreventiveHealth={updatePreventiveHealth}
1196:         onSave={() => saveReportData()}
1197:         onReset={() => resetSection("preventiveHealth")}
1198:     />
1199: </TabsContent>
1200: <TabsContent value="family-genetic-impact">
1201:     <FamilyGeneticImpactAdmin
1202:         familyGeneticImpact={reportData.familyGeneticImpact}
1203:         addFamilyGeneticImpact={addFamilyGeneticImpact}
1204:         updateFamilyGeneticImpact={updateFamilyGeneticImpact}
1205:         removeFamilyGeneticImpact={removeFamilyGeneticImpact}
1206:         onSave={() => saveReportData()}
1207:         onReset={() => resetSection("familyGeneticImpact")}
1208:     />
1209: </TabsContent>
1210: <TabsContent value="genes">
1211:     <GeneResultsAdmin
1212:         geneTestResults={reportData.geneTestResults}
1213:         updateGeneTestResult={updateGeneTestResult}
1214:         addGeneTestResult={addGeneTestResult}
1215:         removeGeneTestResult={removeGeneTestResult}
1216:         onSave={() => saveReportData()}
1217:         onReset={() => resetSection("geneTestResults")}
1218:     />
1219: </TabsContent>
1220: <TabsContent value="summaries">
1221:     <SummariesAdmin
1222:         summaries={reportData.summaries}
1223:         updateSummaries={updateReportSummaries}
1224:         onSave={() => saveReportData()}
1225:         onReset={() => resetSection("summaries")}
1226:     />
1227: </TabsContent>
1228: <TabsContent value="pdf-export">
1229:     <PDFGenerator reportData={reportData} />

```

```

1230:         </TabsContent>
1231:         <TabsContent value="import-export">
1232:             <ImportExportAdmin
1233:                 reportData={reportData}
1234:                 setReportData={setReportData}
1235:             />
1236:         </TabsContent>
1237:     </Tabs>
1238: </div>
1239: );
1240: };
1241:
1242: export default AdminPage;

```

■ 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\report-data\route.ts

```

=====
1: import { promises as fs } from "fs"
2: import path from "path"
3: import { NextResponse } from "next/server"
4: import { v4 as uuidv4 } from "uuid" // Import uuid
5: import type { ComprehensiveReportData } from "@/types/report-types"
6:
7: const dataFilePath = path.join(process.cwd(), "data", "report-data.json")
8:
9: // Default data to initialize if the file doesn't exist
10: const defaultReportData: ComprehensiveReportData = {
11:   patientInfo: {
12:     name: "John Doe",
13:     gender: "MALE",
14:     birthDate: "1990-01-01",
15:     sampleCode: "DNL1000000110",
16:     sampleDate: "2023-01-15",
17:     reportDate: "2023-02-01",
18:     reportAuth: "Dr. Jane Smith, PhD",
19:     genomicAnalytics: "Genomics Lab Inc.",
20:     checkedBy: "Dr. Alice Brown",
21:     scientificContent: "Dr. Robert White",
22:   },
23:   content: {
24:     disclaimer:
25:       "This report is for informational purposes only and should not be used as a substitute for professional medical advice.",
26:     introduction:
27:       "Welcome to your personalized nutrigenomics report. This document provides insights into how your genetic data can inform your health and wellness choices."

```

```

28:     genomicsExplanation:
29:         "Genomics is the study of an organism's complete set of DNA, including all of its genes. It helps us
30:     genesHealthImpact:
31:         "Your genes play a significant role in determining your health predispositions. This report highlights
32:     fundamentalsPRS:
33:         "Polygenic Risk Scores (PRS) combine the effects of many genetic variants to estimate an individual's
34:     utilityDoctors:
35:         "This report is designed to be a valuable tool for doctors and dietitians, providing actionable genet
36:     microarrayExplanation:
37:         "Microarray technology allows for the simultaneous measurement of the expression levels of thousands
38:     microarrayData:
39:         "Your microarray data has been analyzed to identify specific genetic variations relevant to your heal
40: },
41: settings: {
42:     title: "Comprehensive Health Report",
43:     subtitle: "A Detailed Analysis of Your Genetics",
44:     companyName: "Health Solutions Inc.",
45:     headerColor: "#2C3E50",
46:     accentColor: "#E74C3C",
47:     fonts: {
48:         primary: "Helvetica, Arial, sans-serif",
49:         secondary: "Georgia, serif",
50:         mono: "Courier New, monospace",
51:     },
52: },
53: // NEW: Default dynamic diet field definitions
54: dynamicDietFieldDefinitions: [
55:     {
56:         _uuid: uuidv4(),
57:         id: "carbohydrate_sensitivity",
58:         label: "Carbohydrate Sensitivity",
59:         category: "Macronutrients",
60:         min: 1,
61:         max: 10,
62:         highRecommendation: "Reduce carbohydrate intake, focus on complex carbs.",
63:         normalRecommendation: "Maintain balanced carbohydrate intake.",
64:         lowRecommendation: "Ensure adequate carbohydrate intake.",
65:     },
66:     {
67:         _uuid: uuidv4(),
68:         id: "fat_sensitivity",
69:         label: "Fat Sensitivity",
70:         category: "Macronutrients",
71:         min: 1,
72:         max: 10,
73:         highRecommendation: "Reduce saturated and trans fats; focus on healthy fats.",
74:         normalRecommendation: "Maintain balanced fat intake.",
75:         lowRecommendation: "Ensure adequate healthy fat intake.",
76:     },
77:     {
78:         _uuid: uuidv4(),
79:         id: "protein_requirement",
80:         label: "Protein Requirement",
81:         category: "Macronutrients",
82:         min: 1,
83:         max: 10,
84:         highRecommendation: "Increased protein intake recommended.",
85:         normalRecommendation: "Ensure adequate protein intake.",
86:         lowRecommendation: "Consider increasing protein intake for muscle maintenance.",
87:     },
88:     {
89:         _uuid: uuidv4(),
90:         id: "meal_frequency",
91:         label: "Meal Frequency",
92:         category: "Meal Pattern",
93:         min: 1,
94:         max: 10,
95:         highRecommendation: "Consider smaller, more frequent meals.",
96:         normalRecommendation: "Maintain regular meal frequency.",
97:         lowRecommendation: "Consider fewer, larger meals.",
98:     },
99:     {
100:         _uuid: uuidv4(),

```

```

101:         id: "alcohol_sensitivity",
102:         label: "Alcohol Sensitivity",
103:         category: "Food Sensitivities",
104:         min: 1,
105:         max: 10,
106:         highRecommendation: "Limit alcohol consumption due to genetic sensitivity.",
107:         normalRecommendation: "Moderate alcohol intake is fine.",
108:         lowRecommendation: "Normal alcohol metabolism.",
109:     },
110:     {
111:         _uuid: uuidv4(),
112:         id: "caffeine_sensitivity",
113:         label: "Caffeine Sensitivity",
114:         category: "Food Sensitivities",
115:         min: 1,
116:         max: 10,
117:         highRecommendation: "Limit caffeine intake due to slow metabolism.",
118:         normalRecommendation: "Normal caffeine metabolism; moderate intake is fine.",
119:         lowRecommendation: "Fast caffeine metabolism; higher intake may be tolerated.",
120:     },
121: ],
122: // NEW: Default patient-specific results for dynamic diet fields
123: patientDietAnalysisResults: [
124:     {
125:         fieldId: "carbohydrate_sensitivity",
126:         score: 5,
127:         level: "NORMAL",
128:         recommendation: "Maintain balanced carbohydrate intake.",
129:     },
130:     {
131:         fieldId: "fat_sensitivity",
132:         score: 7,
133:         level: "HIGH",
134:         recommendation: "Reduce saturated and trans fats; focus on healthy fats.",
135:     },
136:     {
137:         fieldId: "protein_requirement",
138:         score: 3,
139:         level: "LOW",
140:         recommendation: "Ensure adequate protein intake for muscle maintenance.",
141:     },
142:     {
143:         fieldId: "meal_frequency",
144:         score: 6,
145:         level: "FREQUENT",
146:         recommendation: "Consider smaller, more frequent meals.",
147:     },
148:     {
149:         fieldId: "alcohol_sensitivity",
150:         score: 9,
151:         level: "HIGH",
152:         recommendation: "Limit alcohol consumption due to genetic sensitivity.",
153:     },
154:     {
155:         fieldId: "caffeine_sensitivity",
156:         score: 2,
157:         level: "LOW",
158:         recommendation: "Normal caffeine metabolism; moderate intake is fine.",
159:     },
160: ],
161: nutritionData: {
162:     vitamins: {
163:         vitaminA: { score: 3, healthImpact: "Skin & Vision", intakeLevel: "ENHANCED INTAKE", source: "DIET" },
164:         vitaminB12: {
165:             score: 6,
166:             healthImpact: "Nervous System & Bones",
167:             intakeLevel: "ENHANCED INTAKE (METHYLCOBALAMIN)",
168:             source: "DIET",
169:         },
170:         vitaminB2: { score: 3, healthImpact: "Energy Metabolism", intakeLevel: "ENHANCED INTAKE", source: "DIET" },
171:         vitaminB6: { score: 6, healthImpact: "Tissue growth repair", intakeLevel: "ENHANCED INTAKE", source: "DIET" },
172:         vitaminB9: {
173:             score: 3,

```



```

174:         healthImpact: "RBC development & Heart health",
175:         intakeLevel: "ENHANCED INTAKE (L METHYLFOLATE)",
176:         source: "DIET",
177:     },
178:     vitaminC: { score: 3, healthImpact: "Skin & Immunity", intakeLevel: "ENHANCED INTAKE", source: "DIET" },
179:     vitaminD: {
180:         score: 6,
181:         healthImpact: "Bones & Diabetes",
182:         intakeLevel: "ENHANCED INTAKE & SUN EXPOSURE",
183:         source: "DIET & SUN EXPOSURE",
184:     },
185:     vitaminE: { score: 5, healthImpact: "Skin & Immune system", intakeLevel: "ENHANCED INTAKE", source: "DIET" },
186: },
187: fattyAcids: {
188:     omegaFattyAcids: {
189:         score: 5,
190:         healthImpact: "Cardiac health & Triglycerides",
191:         intakeLevel: "ENHANCED INTAKE (VEG SOURCES)",
192:         source: "ENHANCED INTAKE (VEG/FISH OIL) & DIET",
193:     },
194: },
195: elements: {
196:     calcium: {
197:         score: 3,
198:         healthImpact: "Bone health & Nerve function",
199:         intakeLevel: "ENHANCED INTAKE",
200:         source: "DIET",
201:     },
202:     iodine: {
203:         score: 3,
204:         healthImpact: "Detoxification & Thyroid hormones",
205:         intakeLevel: "ENHANCED INTAKE",
206:         source: "DIET",
207:     },
208:     iron: { score: 3, healthImpact: "Haemoglobin & Energy", intakeLevel: "RESTRICTED INTAKE", source: "DIET" },
209:     magnesium: {
210:         score: 3,
211:         healthImpact: "Heart health & Immunity",
212:         intakeLevel: "ENHANCED INTAKE",
213:         source: "DIET",
214:     },
215:     phosphorus: {
216:         score: 3,
217:         healthImpact: "Bone health & Kidney function",
218:         intakeLevel: "ENHANCED INTAKE",
219:         source: "DIET",
220:     },
221:     selenium: {
222:         score: 3,
223:         healthImpact: "Cognitive & Muscle function",
224:         intakeLevel: "ENHANCED INTAKE",
225:         source: "DIET",
226:     },
227:     zinc: { score: 3, healthImpact: "Fertility & Skin", intakeLevel: "ENHANCED INTAKE", source: "DIET" },
228: },
229: complexNutrients: {
230:     antiInflammatory: {
231:         score: 5,
232:         healthImpact: "Pain control & Organ health",
233:         intakeLevel: "ENHANCED INTAKE",
234:         source: "DIET",
235:     },
236:     antioxidants: {
237:         score: 6,
238:         healthImpact: "Skin & Immune system",
239:         intakeLevel: "ENHANCED INTAKE",
240:         source: "DIET",
241:     },
242: },
243: },
244: sportsAndFitness: {
245:     exerciseType: {
246:         endurancePotential: {

```

```

247:         level: "High Endurance Potential",
248:         description: "Excellent genetic predisposition for endurance activities",
249:     },
250:     powerPotential: { level: "Average Power Potential", description: "Moderate power output capacity" },
251:     strengthProfile: { level: "High Strength Profile", description: "Good potential for strength training" },
252:     exerciseTime: { level: "Evening time is more beneficial than morning", description: "MORNING Vs. EVENING" },
253: },
254: performance: {
255:     enduranceCapacity: { level: "High Endurance Potential", description: "Superior aerobic capacity" },
256:     fatigueResistance: {
257:         level: "Normal fatigue resistance. Average endurance capacity",
258:         description: "Average endurance capacity",
259:     },
260:     bloodFlow: {
261:         level: "Reduced vascular efficiency, affects endurance sports.",
262:         description: "Affects endurance sports",
263:     },
264:     oxygenEfficiency: {
265:         level: "Normal VO2 Max. Need practice to excel in endurance sports",
266:         description: "Need practice to excel",
267:     },
268:     waterLoss: {
269:         level:
270:             "Increased water and electrolyte loss tendency. Focus on Hydration and electrolyte management during sports",
271:         description: "Focus on hydration",
272:     },
273:     lactateClearing: {
274:         level: "High lactate clearance. Beneficial for Endurance sports.",
275:         description: "Beneficial for endurance",
276:     },
277:     recoveryEfficiency: {
278:         level: "High recovery efficiency. Enhanced sports potential & reduced injury risk",
279:         description: "Enhanced sports potential",
280:     },
281:     bodyComposition: {
282:         level: "Low fat accumulation. Increased muscle tendency.",
283:         description: "Increased muscle tendency",
284:     },
285:     injuryRisk: {
286:         level: "High injury tendency. Thorough injury management steps to be taken.",
287:         description: "Thorough injury management needed",
288:     },
289:     achillesTendonInjury: {
290:         level: "Low injury tendency. Ensure adequate warmup before exercises",
291:         description: "Ensure adequate warmup",
292:     },
293:     ligamentInjury: {
294:         level: "Low ACL injury tendency. Ensure adequate warmup before exercises",
295:         description: "Ensure adequate warmup",
296:     },
297:     rupture: {
298:         level: "Low rupture tendency. Ensure adequate warmup before exercises",
299:         description: "Ensure adequate warmup",
300:     },
301:     muscleCramps: {
302:         level: "Low risk of muscle cramps. Ensure adequate warmup before exercises",
303:         description: "Ensure adequate warmup",
304:     },
305: },
306: },
307: lifestyleConditions: {
308:     heartVascularHealth: {
309:         hypertension: { status: "strength" },
310:         leftVentricularHypertrophy: { status: "strength" },
311:         atherosclerosis: { status: "improvement" },
312:         familialHypercholesterolemia: { status: "strength" },
313:         cardiomyopathy: { status: "strength" },
314:         cholesterolImbalance: { status: "improvement" },
315:         stroke: { status: "improvement" },
316:     },
317:     diabetes: {
318:         obesity: { status: "improvement" },
319:         diabetesType2: { status: "improvement" },

```

```

320:     },
321:     liverHealth: {
322:         nafld: { status: "improvement" },
323:         liverCirrhosis: { status: "strength" },
324:     },
325:     boneHealth: {
326:         osteoporosis: { status: "improvement" },
327:         osteoarthritis: { status: "improvement" },
328:     },
329:     gutHealth: {
330:         irritableBowelSyndrome: { status: "improvement" },
331:         celiacDisease: { status: "strength" },
332:     },
333:     hormoneSystem: {
334:         hypothyroidism: { status: "improvement" },
335:     },
336: },
337: metabolicCore: {
338:     methylation: {
339:         genes: ["MTHFR"],
340:         genotype: "C/T",
341:         impact: "Reduced folate metabolism",
342:         advice: "Consider L-methylfolate supplementation.",
343:     },
344:     inflammation: {
345:         genes: ["CRP"],
346:         genotype: "G/A",
347:         impact: "Increased inflammatory response",
348:         advice: "Incorporate anti-inflammatory foods.",
349:     },
350:     oxidativeStress: {
351:         genes: ["SOD2"],
352:         genotype: "T/C",
353:         impact: "Reduced antioxidant defense",
354:         advice: "Increase antioxidant-rich foods and supplements.",
355:     },
356:     detoxification: {
357:         genes: ["GSTP1"],
358:         genotype: "A/G",
359:         impact: "Impaired detoxification capacity",
360:         advice: "Support liver health with cruciferous vegetables.",
361:     },
362:     vitaminDMetabolism: {
363:         genes: ["VDR"],
364:         genotype: "C/C",
365:         impact: "Reduced Vitamin D absorption",
366:         advice: "Ensure adequate Vitamin D intake and sun exposure.",
367:     },
368:     lipidMetabolism: {
369:         genes: ["APOE"],
370:         genotype: "E3/E4",
371:         impact: "Increased LDL cholesterol tendency",
372:         advice: "Monitor lipid levels and follow a heart-healthy diet.",
373:     },
374:     insulinResistance: {
375:         genes: ["TCF7L2"],
376:         genotype: "T/T",
377:         impact: "Higher risk of insulin resistance",
378:         advice: "Focus on low glycemic index foods and regular exercise.",
379:     },
380:     circadianRhythm: {
381:         genes: ["CLOCK"],
382:         genotype: "C/G",
383:         impact: "Altered sleep-wake cycle",
384:         advice: "Maintain consistent sleep schedule and optimize sleep environment.",
385:     },
386:     exerciseResponsiveness: {
387:         genes: ["ACTN3"],
388:         genotype: "R/X",
389:         impact: "Enhanced power and strength response to training",
390:         advice: "Incorporate strength and power training into routine.",
391:     },
392:     snackingTendency: {

```

```

393:     genes: ["FTO"],
394:     genotype: "A/A",
395:     impact: "Increased tendency for snacking and higher BMI",
396:     advice: "Practice mindful eating and portion control.",
397: },
398: oestrogenElimination: {
399:     genes: ["COMT"],
400:     genotype: "G/A",
401:     impact: "Slower estrogen elimination",
402:     advice: "Support liver detoxification pathways.",
403: },
404: powerVsEndurance: {
405:     genes: ["ACE"],
406:     genotype: "I/D",
407:     impact: "Balanced power and endurance potential",
408:     advice: "Vary training to include both power and endurance exercises.",
409: },
410: injuryRecovery: {
411:     genes: ["COL1A1"],
412:     genotype: "G/T",
413:     impact: "Slower collagen repair, increased injury risk",
414:     advice: "Focus on adequate protein and Vitamin C for tissue repair.",
415: },
416: fluidLossTendency: {
417:     genes: ["AGT"],
418:     genotype: "T/C",
419:     impact: "Increased fluid and electrolyte loss during exercise",
420:     advice: "Ensure proper hydration and electrolyte replenishment.",
421: },
422: vo2Max: {
423:     genes: ["PPARGC1A"],
424:     genotype: "G/G",
425:     impact: "High VO2 Max potential",
426:     advice: "Optimize cardiovascular training for peak performance.",
427: },
428: },
429: digestiveHealth: {
430:     gutHealth: { level: "NORMAL GUT HEALTH", description: "Good overall digestive function" },
431:     glutenIntolerance: { level: "LOW INTOLERANCE", description: "CONDITIONAL GLUTEN ELIMINATION NEEDED" },
432:     celiacDiseaseTendency: { level: "LOW TENDENCY", description: "Low genetic risk for celiac disease" },
433:     impactOnBonesJoints: {
434:         level: "NO IMPACT ON BONE HEALTH",
435:         description: "Digestive health does not affect bone health",
436:     },
437:     lactoseIntolerance: { level: "HIGH INTOLERANCE", description: "LACTOSE ELIMINATION NEEDED" },
438:     probioticsRequirement: { level: "PROBIOTICS NEEDED", description: "FOOD & SUPPLEMENTS" },
439: },
440: genesAndAddiction: {
441:     alcoholAddiction: { tendency: "LOW", description: "Low genetic predisposition to alcohol dependency" },
442:     smokingAddiction: { tendency: "NORMAL", description: "Average nicotine addiction susceptibility" },
443:     snackingHabits: { tendency: "HIGH", description: "Genetic tendency toward frequent snacking" },
444:     altruism: { tendency: "HIGH", description: "Strong genetic predisposition toward helping others" },
445: },
446: sleepAndRest: {
447:     sleepCycle: {
448:         impact: "DELAYED SLEEP CYCLE",
449:         intervention: "EXERCISE IN THE EVENING AVOID CARBS DURING BREAKFAST",
450:     },
451:     sleepApnea: {
452:         impact: "INCREASED CHANCES OF SLEEP APNEA",
453:         intervention: "EXERCISE REGULARLY SEEK MEDICAL ADVICE IF ISSUE PERSISTS",
454:     },
455:     sleepDepth: { impact: "LOW SLEEP DEPTH", intervention: "CONSUME CURD RICE IN DINNER FOR BETTER SLEEP" },
456:     stressSensitivity: {
457:         impact: "LOW STRESS SENSITIVITY",
458:         intervention: "EXERCISE REGULARLY FOR A BETTER SLEEP CYCLE",
459:     },
460: },
461: allergiesAndSensitivity: {
462:     dustAllergy: { tendency: "NORMAL" },
463:     pesticideSensitivity: { tendency: "HIGH" },
464:     smokeSensitivity: { tendency: "HIGH" },
465:     automobileSmokeSensitivity: { tendency: "HIGH" },

```

```

466:     generalAdvice:
467:       "THE BEST WAY TO REDUCE ALLERGIES IS TO STAY AWAY FROM THE SOURCE, NO ROCKET SCIENCE HOWEVER THE FOLL
468:     },
469:     geneTestResults: [
470:       {
471:         geneName: "ALDH2 Aldehyde dehydrogenase 2",
472:         variation: "G>A",
473:         result: "GG",
474:       },
475:       {
476:         geneName: "MTHFR Methylene tetrahydrofolate reductase",
477:         variation: "C>T",
478:         result: "CT",
479:       },
480:     ],
481:     metabolicSummary: {
482:       strengths: ["Efficient detoxification", "Normal lipid metabolism"],
483:       weaknesses: ["Increased oxidative stress", "Higher risk of insulin resistance"],
484:     },
485:     preventiveHealth: {
486:       diagnosticTests: {
487:         halfYearly: ["PLASMA SUGAR (FASTING & PP)", "VITAMIN D"],
488:         yearly: ["HbA1c", "LIPID PROFILE", "SERUM HOMOCYSTEINE", "VITAMIN B12", "LIVER PROFILE"],
489:       },
490:       nutritionalSupplements: [
491:         { supplement: "VITAMIN D", needed: true },
492:         { supplement: "METHYL FOLATE + METHYL COBALAMINE + B6", needed: true },
493:         { supplement: "OMEGA FATTY ACIDS (FISH OIL)", needed: true },
494:         { supplement: "FOLIC ACID + COBALAMINE + B6", needed: true },
495:         { supplement: "OMEGA FATTY ACIDS (VEG)", needed: true },
496:         { supplement: "VITAMIN E", needed: true },
497:         { supplement: "PROBIOTICS", needed: true },
498:       ],
499:     },
500:     familyGeneticImpact: [
501:       {
502:         gene: "MnSOD",
503:         normalAlleles: "AA",
504:         yourResult: "AG",
505:         healthImpact: "INCREASED TENDENCY OF OXIDATIVE STRESS NEED OF ANTI-OXIDANT SUPPLEMENTS",
506:       },
507:       {
508:         gene: "FTO",
509:         normalAlleles: "TT",
510:         yourResult: "AT",
511:         healthImpact:
512:           "INCREASED TENDENCY OF INSULIN RESISTANCE WEIGHT AND CALORIE MANAGEMENT IMPORTANT TO MANAGE AND RED
513:       },
514:     ],
515:     summaries: {
516:       nutrigenomicsSummary:
517:         "Based on the overall nutrigenomics analysis, provide expert summary for personalized diet and lifestyle",
518:       exerciseGenomicsSummary:
519:         "Based on the overall genomics analysis, provide expert summary for personalized exercise plan...",
520:     },
521:   }
522:
523: export async function GET() {
524:   try {
525:     // Ensure the data directory exists
526:     const dataDir = path.dirname(dataFilePath)
527:     await fs.mkdir(dataDir, { recursive: true })
528:
529:     let data: ComprehensiveReportData
530:
531:     try {
532:       const fileContent = await fs.readFile(dataFilePath, "utf-8")
533:       data = JSON.parse(fileContent)
534:     } catch (readError: any) {
535:       if (readError.code === "ENOENT") {
536:         // File does not exist, initialize with default data
537:         data = defaultReportData
538:       }
539:       await fs.writeFile(dataFilePath, JSON.stringify(data, null, 2), "utf-8")

```

```

539:         } else {
540:             // Other read errors
541:             console.error("Error reading report data file:", readError)
542:             return NextResponse.json({ error: "Failed to read report data" }, { status: 500 })
543:         }
544:     }
545:
546:     return NextResponse.json(data)
547: } catch (error) {
548:     console.error("Error in GET /api/report-data:", error)
549:     return NextResponse.json({ error: "Internal Server Error" }, { status: 500 })
550: }
551: }
552:
553: export async function POST(request: Request) {
554:     try {
555:         const data: ComprehensiveReportData = await request.json()
556:
557:         // Basic validation (can be expanded)
558:         if (!data || !data.patientInfo || !data.content) {
559:             return NextResponse.json({ error: "Invalid report data provided" }, { status: 400 })
560:         }
561:
562:         // Ensure the data directory exists
563:         const dataDir = path.dirname(dataFilePath)
564:         await fs.mkdir(dataDir, { recursive: true })
565:
566:         await fs.writeFile(dataFilePath, JSON.stringify(data, null, 2), "utf-8")
567:         return NextResponse.json({ message: "Report data saved successfully" })
568:     } catch (error) {
569:         console.error("Error in POST /api/report-data:", error)
570:         return NextResponse.json({ error: "Internal Server Error" }, { status: 500 })
571:     }
572: }

```

■ 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: 'v0.dev'
14: }
15:
16: export default function RootLayout({
17:     children,
18: }: Readonly<{
19:     children: React.ReactNode
20: }>) {
21:     return (
22:         <html lang="en" suppressHydrationWarning>
23:             <body className={inter.className}>
24:                 <ThemeProvider attribute="class" defaultTheme="system" enableSystem disableTransitionOnChange>
25:                     {children}
26:                 <Toaster />
27:             </ThemeProvider>
28:         </body>
29:     </html>
30: )
31: }

```

■ 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:       {/* Quick Actions */}
25:       <div className="grid md:grid-cols-4 gap-6 mb-12 mt-12">
26:         <Link href="/admin">
27:           <Card className="hover:shadow-xl transition-all duration-300 border-0 bg-gradient-to-br from-blue
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">Admin Panel</h3>
31:               <p className="text-blue-100 text-sm">Manage all data</p>
32:             </CardContent>
33:           </Card>
34:         </Link>
35:
36:         <Link href="/report/preview">
37:           <Card className="hover:shadow-xl transition-all duration-300 border-0 bg-gradient-to-br from-gre
38:             <CardContent className="p-6 text-center">
39:               <Eye className="h-12 w-12 mx-auto mb-4" />
40:               <h3 className="font-bold text-lg">Preview Report</h3>
41:               <p className="text-green-100 text-sm">View live report</p>
42:             </CardContent>
43:           </Card>
44:         </Link>
45:
46:         <Link href="/report/pdf">
47:           <Card className="hover:shadow-xl transition-all duration-300 border-0 bg-gradient-to-br from-purp
48:             <CardContent className="p-6 text-center">
49:               <Download className="h-12 w-12 mx-auto mb-4" />
50:               <h3 className="font-bold text-lg">Generate PDF</h3>
51:               <p className="text-purple-100 text-sm">Export to PDF</p>
52:             </CardContent>
53:           </Card>
54:         </Link>
55:
56:         <Card className="hover:shadow-xl transition-all duration-300 border-0 bg-gradient-to-br from-orange
57:           <CardContent className="p-6 text-center">
58:             <Database className="h-12 w-12 mx-auto mb-4" />
59:             <h3 className="font-bold text-lg">Data Management</h3>
60:             <p className="text-orange-100 text-sm">CRUD operations</p>
61:           </CardContent>
62:         </Card>
63:       </div>
64:
65:       {/* Report Sections */}
66:       <Card className="shadow-lg border-0 bg-white mb-12">
67:         <CardHeader className="bg-gradient-to-r from-indigo-500 to-purple-600 text-white rounded-t-lg">
68:           <CardTitle className="text-2xl">
69:             ? Report Sections Included
70:           </CardTitle>
71:           <CardDescription className="text-indigo-100">
```

```

72:         All sections from the original PDF with complete data management
73:     </CardDescription>
74: </CardHeader>
75: <CardContent className="p-8">
76:     <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-6">
77:         <div className="flex items-center gap-3 p-4 bg-red-50 rounded-lg">
78:             <Utensils className="h-8 w-8 text-red-600" />
79:             <div>
80:                 <h4 className="font-semibold text-red-800">Diet Analysis</h4>
81:                 <p className="text-sm text-red-600">
82:                     Macronutrients, sensitivities, meal patterns
83:                 </p>
84:             </div>
85:         </div>
86:
87:         <div className="flex items-center gap-3 p-4 bg-blue-50 rounded-lg">
88:             <Activity className="h-8 w-8 text-blue-600" />
89:             <div>
90:                 <h4 className="font-semibold text-blue-800">Nutrition</h4>
91:                 <p className="text-sm text-blue-600">
92:                     Vitamins, minerals, supplements
93:                 </p>
94:             </div>
95:         </div>
96:
97:         <div className="flex items-center gap-3 p-4 bg-green-50 rounded-lg">
98:             <Zap className="h-8 w-8 text-green-600" />
99:             <div>
100:                 <h4 className="font-semibold text-green-800">
101:                     Sports & Fitness
102:                 </h4>
103:                 <p className="text-sm text-green-600">
104:                     Exercise types, performance factors
105:                 </p>
106:             </div>
107:         </div>
108:
109:         <div className="flex items-center gap-3 p-4 bg-yellow-50 rounded-lg">
110:             <Heart className="h-8 w-8 text-yellow-600" />
111:             <div>
112:                 <h4 className="font-semibold text-yellow-800">
113:                     Lifestyle Conditions
114:                 </h4>
115:                 <p className="text-sm text-yellow-600">
116:                     Health risks and strengths
117:                 </p>
118:             </div>
119:         </div>
120:
121:         <div className="flex items-center gap-3 p-4 bg-purple-50 rounded-lg">
122:             <Dna className="h-8 w-8 text-purple-600" />
123:             <div>
124:                 <h4 className="font-semibold text-purple-800">
125:                     Metabolic Core
126:                 </h4>
127:                 <p className="text-sm text-purple-600">
128:                     Gene analysis and impacts
129:                 </p>
130:             </div>
131:         </div>
132:
133:         <div className="flex items-center gap-3 p-4 bg-indigo-50 rounded-lg">
134:             <Brain className="h-8 w-8 text-indigo-600" />
135:             <div>
136:                 <h4 className="font-semibold text-indigo-800">
137:                     Digestive Health
138:                 </h4>
139:                 <p className="text-sm text-indigo-600">
140:                     Gut health, intolerances
141:                 </p>
142:             </div>
143:         </div>
144:

```



```

145:         <div className="flex items-center gap-3 p-4 bg-pink-50 rounded-lg">
146:             <Dna className="h-8 w-8 text-pink-600" />
147:             <div>
148:                 <h4 className="font-semibold text-pink-800">
149:                     Genes & Addiction
150:                 </h4>
151:                 <p className="text-sm text-pink-600">Tendencies, advice</p>
152:             </div>
153:         </div>
154:
155:         <div className="flex items-center gap-3 p-4 bg-teal-50 rounded-lg">
156:             <Brain className="h-8 w-8 text-teal-600" />
157:             <div>
158:                 <h4 className="font-semibold text-teal-800">Sleep & Rest</h4>
159:                 <p className="text-sm text-teal-600">Sleep patterns, quality</p>
160:             </div>
161:         </div>
162:
163:         <div className="flex items-center gap-3 p-4 bg-cyan-50 rounded-lg">
164:             <AlertCircle className="h-8 w-8 text-cyan-600" />
165:             <div>
166:                 <h4 className="font-semibold text-cyan-800">
167:                     Allergies & Sensitivity
168:                 </h4>
169:                 <p className="text-sm text-cyan-600">
170:                     Environmental sensitivities
171:                 </p>
172:             </div>
173:         </div>
174:
175:         <div className="flex items-center gap-3 p-4 bg-lime-50 rounded-lg">
176:             <Heart className="h-8 w-8 text-lime-600" />
177:             <div>
178:                 <h4 className="font-semibold text-lime-800">
179:                     Preventive Health
180:                 </h4>
181:                 <p className="text-sm text-lime-600">
182:                     Diagnostic tests, supplements
183:                 </p>
184:             </div>
185:         </div>
186:
187:         <div className="flex items-center gap-3 p-4 bg-amber-50 rounded-lg">
188:             <Users className="h-8 w-8 text-amber-600" />
189:             <div>
190:                 <h4 className="font-semibold text-amber-800">
191:                     Family Genetic Impact
192:                 </h4>
193:                 <p className="text-sm text-amber-600">
194:                     Inherited traits, family recommendations
195:                 </p>
196:             </div>
197:         </div>
198:     </div>
199: </CardContent>
200: </Card>
201: </div>
202: );
203: }

```

■ 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:     // In a real application, you would fetch data based on the 'id'
16:     // For this example, we fetch the single stored report data
17:     const response = await fetch(
18:       `${
19:         process.env.NEXT_PUBLIC_APP_URL || "http://localhost:3000"
20:       }/api/report-data`,
21:       { cache: "no-store" }
22:     );
23:
24:     if (!response.ok) {
25:       console.error(
26:         `Failed to fetch report data: ${response.status} ${response.statusText}`
27:       );
28:       return null;
29:     }
30:
31:     const data = await response.json();
32:     // If you had multiple reports, you'd filter by ID here
33:     return data as ComprehensiveReportData;
34:   } catch (error) {
35:     console.error("Error fetching report data:", error);
36:     return null;
37:   }
38: }
39:
40: export default async function ReportPage({ params }: ReportPageProps) {
41:   const reportData = await getReportData(params.id);
42:
43:   if (!reportData) {
44:     notFound();
45:   }
46:
47:   return (
48:     <div className="min-h-screen bg-gray-100 print:bg-white">
49:       <ComprehensiveReportViewer reportData={reportData} />
50:     </div>
51:   );
52: }

```

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/report-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: 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/report-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 ReportPreviewPage() {
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 preview.
35:       </div>
36:     );
37:   }
38:
39:   return (
40:     <div className="min-h-screen bg-gray-100 print:bg-white">
41:       <ComprehensiveReportViewer reportData={reportData} />
42:     </div>
43:   );
44: }

```

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

```
=====
1: "use client"
2:
3: import { Card, CardContent, CardDescription, 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 { Save, RotateCcw } from "lucide-react"
9: import type { AllergiesAndSensitivity, AllergyData } from "@types/report-types"
10:
11: interface AllergiesSensitivityAdminProps {
12:   allergiesAndSensitivity: AllergiesAndSensitivity
13:   updateAllergiesAndSensitivity: (field: keyof AllergiesAndSensitivity, data: Partial<AllergyData> | string
14:   onSave: () => void
15:   onReset: () => void
16: }
17:
18: export default function AllergiesSensitivityAdmin({
19:   allergiesAndSensitivity,
20:   updateAllergiesAndSensitivity,
21:   onSave,
22:   onReset,
23: }: AllergiesSensitivityAdminProps) {
24:   const renderAllergyField = (
25:     field: Exclude<keyof AllergiesAndSensitivity, "generalAdvice">,
26:     data: AllergyData,
27:     title: string,
28:   ) => (
29:     <Card key={field} className="border-2 border-orange-100 bg-orange-50/30">
30:       <CardHeader className="pb-3">
31:         <CardTitle className="text-lg text-orange-800">{title}</CardTitle>
32:       </CardHeader>
33:       <CardContent className="space-y-4">
34:         <div className="space-y-2">
35:           <Label className="text-sm font-medium">Tendency</Label>
36:           <Input
37:             value={data.tendency}
38:             onChange={(e) => updateAllergiesAndSensitivity(field, { tendency: e.target.value })}
39:             placeholder="e.g., NORMAL, HIGH"
40:             className="border-2 focus:border-orange-500"
41:           />
42:         </div>
43:       </CardContent>
44:     </Card>
45:   )
46:
47:   return (
48:     <Card className="shadow-lg border-0 bg-white">
49:       <CardHeader className="bg-gradient-to-r from-orange-500 to-red-600 text-white rounded-t-lg">
50:         <CardTitle className="text-2xl">? Allergies & Sensitivity</CardTitle>
51:         <CardDescription className="text-orange-100">
52:           Manage genetic predispositions for various allergies and environmental sensitivities
53:         </CardDescription>
54:       </CardHeader>
55:       <CardContent className="space-y-8 p-8">
56:         { /* Action Buttons */ }
57:         <div className="flex justify-end gap-2 mb-6">
58:           <Button onClick={onReset} variant="outline" size="sm">
59:             <RotateCcw className="h-4 w-4 mr-2" />
60:             Reset
61:           </Button>
62:           <Button onClick={onSave} size="sm">
63:             <Save className="h-4 w-4 mr-2" />
64:             Save Changes
65:           </Button>
66:         </div>
67:
68:         { /* Allergy & Sensitivity Fields */ }
69:         <div className="grid md:grid-cols-2 lg:grid-cols-2 gap-6">
70:           {Object.entries(allergiesAndSensitivity)
71:             .filter(([key]) => key !== "generalAdvice")
=====
```

```

72:         .map(([key, item]) =>
73:             renderAllergyField(
74:                 key as Exclude<keyof AllergiesAndSensitivity, "generalAdvice">,
75:                 item as AllergyData,
76:                 key.replace(/([A-Z])/g, " $1").replace(/^../, (str) => str.toUpperCase()),
77:             ),
78:         )}
79:     </div>
80:
81:     { /* General Advice */ }
82:     <div className="space-y-2">
83:         <h3 className="text-xl font-semibold text-gray-800 border-b-2 border-orange-200 pb-2">? General Advice
84:         <Label className="text-sm font-medium">General Advice for Allergies & Sensitivities</Label>
85:         <Textarea
86:             value={allergiesAndSensitivity.generalAdvice}
87:             onChange={(e) => updateAllergiesAndSensitivity("generalAdvice", e.target.value)}
88:             rows={5}
89:             className="border-2 focus:border-orange-500 text-sm"
90:             placeholder="Enter general advice for managing allergies and sensitivities..."
91:         />
92:     </div>
93:
94:     { /* Section Guidelines */ }
95:     <div className="bg-orange-50 border-2 border-orange-200 rounded-lg p-6">
96:         <h4 className="font-semibold text-orange-800 mb-2">? Allergies & Sensitivity Guidelines:</h4>
97:         <div className="text-sm text-orange-700 space-y-2">
98:             <ul className="list-disc list-inside space-y-1">
99:                 <li>
100:                     <strong>Dust Allergy:</strong> Genetic tendency towards allergic reactions to dust.
101:                 </li>
102:                 <li>
103:                     <strong>Pesticide Sensitivity:</strong> Genetic susceptibility to adverse reactions from pesticides.
104:                 </li>
105:                 <li>
106:                     <strong>Smoke Sensitivity:</strong> Genetic predisposition to sensitivity from smoke (e.g.,
107:                     environmental).
108:                 </li>
109:                 <li>
110:                     <strong>Automobile Smoke Sensitivity:</strong> Genetic tendency to react to vehicle exhaust.
111:                 </li>
112:                 <li>Genetic variations can influence immune system responses to various environmental factors.
113:                 <li>Personalized advice can help in reducing hyper-reactivity and managing symptoms.</li>
114:             </ul>
115:         </div>
116:     </div>
117: </CardContent>
118: </Card>
119: )
120: }

```

File: components\admin\content-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" // Import icons
8: import type { ReportContent } from "@/types/report-types"
9:
10: interface ContentAdminProps {
11:     content: ReportContent
12:     updateContent: (field: keyof ReportContent, value: string) => void
13:     onSave: () => void // Added onSave
14:     onReset: () => void // Added onReset
15: }
16:
17: export default function ContentAdmin({ content, updateContent, onSave, onReset }: ContentAdminProps) {
18:     return (
19:         <Card className="shadow-lg border-0 bg-white">

```

```

20:     <CardHeader className="bg-gradient-to-r from-green-500 to-teal-600 text-white rounded-t-lg">
21:         <CardTitle className="text-2xl">? Report Content</CardTitle>
22:         <CardDescription className="text-green-100">Edit the main content sections of the report</CardDescr
23:     </CardHeader>
24:     <CardContent className="space-y-8 p-8">
25:         {/* Action Buttons */}
26:         <div className="flex justify-end gap-2 mb-6">
27:             <Button onClick={onReset} variant="outline" size="sm">
28:                 <RotateCcw className="h-4 w-4 mr-2" />
29:                 Reset
30:             </Button>
31:             <Button onClick={onSave} size="sm">
32:                 <Save className="h-4 w-4 mr-2" />
33:                 Save Changes
34:             </Button>
35:         </div>
36:
37:         <div className="space-y-3">
38:             <Label htmlFor="disclaimer" className="text-lg font-semibold text-gray-800">
39:                 ?? Disclaimer
40:             </Label>
41:             <Textarea
42:                 id="disclaimer"
43:                 value={content.disclaimer}
44:                 onChange={(e) => updateContent("disclaimer", e.target.value)}
45:                 placeholder="Enter disclaimer text"
46:                 rows={10}
47:                 className="border-2 focus:border-green-500 text-sm leading-relaxed"
48:             />
49:         </div>
50:         <div className="space-y-3">
51:             <Label htmlFor="introduction" className="text-lg font-semibold text-gray-800">
52:                 ? Welcome Letter Introduction
53:             </Label>
54:             <Textarea
55:                 id="introduction"
56:                 value={content.introduction}
57:                 onChange={(e) => updateContent("introduction", e.target.value)}
58:                 placeholder="Enter introduction text"
59:                 rows={8}
60:                 className="border-2 focus:border-green-500 text-sm leading-relaxed"
61:             />
62:         </div>
63:         <div className="space-y-3">
64:             <Label htmlFor="genomicsExplanation" className="text-lg font-semibold text-gray-800">
65:                 ? What is Genomics?
66:             </Label>
67:             <Textarea
68:                 id="genomicsExplanation"
69:                 value={content.genomicsExplanation}
70:                 onChange={(e) => updateContent("genomicsExplanation", e.target.value)}
71:                 placeholder="Enter genomics explanation"
72:                 rows={8}
73:                 className="border-2 focus:border-green-500 text-sm leading-relaxed"
74:             />
75:         </div>
76:         <div className="space-y-3">
77:             <Label htmlFor="genesHealthImpact" className="text-lg font-semibold text-gray-800">
78:                 ? Genes and Health Impact
79:             </Label>
80:             <Textarea
81:                 id="genesHealthImpact"
82:                 value={content.genesHealthImpact}
83:                 onChange={(e) => updateContent("genesHealthImpact", e.target.value)}
84:                 placeholder="Enter genes and health impact content"
85:                 rows={8}
86:                 className="border-2 focus:border-green-500 text-sm leading-relaxed"
87:             />
88:         </div>
89:         <div className="space-y-3">
90:             <Label htmlFor="fundamentalsPRS" className="text-lg font-semibold text-gray-800">
91:                 ? Fundamentals and PRS
92:             </Label>

```

```

93:         <Textarea
94:             id="fundamentalsPRS"
95:             value={content.fundamentalsPRS}
96:             onChange={(e) => updateContent("fundamentalsPRS", e.target.value)}
97:             placeholder="Enter fundamentals and PRS content"
98:             rows={8}
99:             className="border-2 focus:border-green-500 text-sm leading-relaxed"
100:         />
101:     </div>
102:     <div className="space-y-3">
103:         <Label htmlFor="utilityDoctors" className="text-lg font-semibold text-gray-800">
104:             ??? Utility for Doctors and Dietitians
105:         </Label>
106:         <Textarea
107:             id="utilityDoctors"
108:             value={content.utilityDoctors}
109:             onChange={(e) => updateContent("utilityDoctors", e.target.value)}
110:             placeholder="Enter utility for doctors content"
111:             rows={8}
112:             className="border-2 focus:border-green-500 text-sm leading-relaxed"
113:         />
114:     </div>
115:     <div className="space-y-3">
116:         <Label htmlFor="microarrayExplanation" className="text-lg font-semibold text-gray-800">
117:             ? Microarray Explanation
118:         </Label>
119:         <Textarea
120:             id="microarrayExplanation"
121:             value={content.microarrayExplanation}
122:             onChange={(e) => updateContent("microarrayExplanation", e.target.value)}
123:             placeholder="Enter microarray explanation"
124:             rows={8}
125:             className="border-2 focus:border-green-500 text-sm leading-relaxed"
126:         />
127:     </div>
128:     <div className="space-y-3">
129:         <Label htmlFor="microarrayData" className="text-lg font-semibold text-gray-800">
130:             ? Your Microarray Data
131:         </Label>
132:         <Textarea
133:             id="microarrayData"
134:             value={content.microarrayData}
135:             onChange={(e) => updateContent("microarrayData", e.target.value)}
136:             placeholder="Enter microarray data content"
137:             rows={8}
138:             className="border-2 focus:border-green-500 text-sm leading-relaxed"
139:         />
140:     </div>
141: </CardContent>
142: </Card>
143: )
144: }

```

■ File: components\admin\diet-analysis-admin.tsx

```

=====
1: "use client"
2:
3: import { Card, CardContent, CardDescription, 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 { Select, SelectContent, SelectItem, SelectTrigger, SelectValue } from "@components/ui/select"
8: import { Button } from "@components/ui/button" // Import Button
9: import { Save, RotateCcw } from "lucide-react" // Import icons
10: import type { DietAnalysis, MacronutrientData } from "@types/report-types"
11:
12: interface DietAnalysisAdminProps {
13:     dietAnalysis: DietAnalysis
14:     updateDietAnalysis: (section: keyof DietAnalysis, field: string, data: Partial<MacronutrientData>) => void
15:     onSave: () => void // Added onSave
16:     onReset: () => void // Added onReset

```

```

17: }
18:
19: export default function DietAnalysisAdmin({
20:   dietAnalysis,
21:   updateDietAnalysis,
22:   onSave,
23:   onReset,
24: }: DietAnalysisAdminProps) {
25:   const renderMacronutrientField = (
26:     section: keyof DietAnalysis,
27:     field: string,
28:     data: MacronutrientData,
29:     title: string,
30:   ) => (
31:     <Card key={field} className="border-2 border-blue-100 bg-blue-50/30">
32:       <CardHeader className="pb-3">
33:         <CardTitle className="text-lg text-blue-800">{title}</CardTitle>
34:       </CardHeader>
35:       <CardContent className="space-y-4">
36:         <div className="grid grid-cols-2 gap-4">
37:           <div className="space-y-2">
38:             <Label className="text-sm font-medium">Score (1-10)</Label>
39:             <Input
40:               type="number"
41:               min="1"
42:               max="10"
43:               value={data.score}
44:               onChange={(e) => updateDietAnalysis(section, field, { score: Number.parseInt(e.target.value) }
45:               className="border-2 focus:border-blue-500"
46:             />
47:           </div>
48:           <div className="space-y-2">
49:             <Label className="text-sm font-medium">Level</Label>
50:             <Select value={data.level} onChange={(value) => updateDietAnalysis(section, field, { level
51:               <SelectTrigger className="border-2 focus:border-blue-500">
52:                 <SelectValue />
53:               </SelectTrigger>
54:               <SelectContent>
55:                 <SelectItem value="HIGH">HIGH</SelectItem>
56:                 <SelectItem value="NORMAL">NORMAL</SelectItem>
57:                 <SelectItem value="LOW">LOW</SelectItem>
58:                 <SelectItem value="FREQUENT">FREQUENT</SelectItem>
59:                 <SelectItem value="LESS FREQUENT">LESS FREQUENT</SelectItem>
60:                 <SelectItem value="SUITABLE">SUITABLE</SelectItem>
61:                 <SelectItem value="LESS SUITABLE">LESS SUITABLE</SelectItem>
62:               </SelectContent>
63:             </Select>
64:           </div>
65:         </div>
66:         <div className="space-y-2">
67:           <Label className="text-sm font-medium">Recommendation</Label>
68:           <Textarea
69:             value={data.recommendation}
70:             onChange={(e) => updateDietAnalysis(section, field, { recommendation: e.target.value }}
71:             rows={3}
72:             className="border-2 focus:border-blue-500 text-sm"
73:           />
74:         </div>
75:       </CardContent>
76:     </Card>
77:   )
78:
79:   return (
80:     <Card className="shadow-lg border-0 bg-white">
81:       <CardHeader className="bg-gradient-to-r from-amber-500 to-orange-600 text-white rounded-t-lg">
82:         <CardTitle className="text-2xl">? Diet Analysis</CardTitle>
83:         <CardDescription className="text-amber-100">
84:           Configure macronutrients, meal patterns, and food sensitivities
85:         </CardDescription>
86:       </CardHeader>
87:       <CardContent className="space-y-8 p-8">
88:         { /* Action Buttons */ }
89:         <div className="flex justify-end gap-2 mb-6">

```



```

90:         <Button onClick={onReset} variant="outline" size="sm">
91:             <RotateCcw className="h-4 w-4 mr-2" />
92:             Reset
93:         </Button>
94:         <Button onClick={onSave} size="sm">
95:             <Save className="h-4 w-4 mr-2" />
96:             Save Changes
97:         </Button>
98:     </div>
99:
100:     { /* Macronutrients Section */}
101:     <div className="space-y-4">
102:         <h3 className="text-xl font-semibold text-gray-800 border-b-2 border-amber-200 pb-2">? Macronutri
103:         <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-6">
104:             {renderMacronutrientField(
105:                 "macronutrients",
106:                 "carbohydrateSensitivity",
107:                 dietAnalysis.macronutrients.carbohydrateSensitivity,
108:                 "Carbohydrate Sensitivity",
109:             )}
110:             {renderMacronutrientField(
111:                 "macronutrients",
112:                 "fatSensitivity",
113:                 dietAnalysis.macronutrients.fatSensitivity,
114:                 "Fat Sensitivity",
115:             )}
116:             {renderMacronutrientField(
117:                 "macronutrients",
118:                 "proteinRequirement",
119:                 dietAnalysis.macronutrients.proteinRequirement,
120:                 "Protein Requirement",
121:             )}
122:         </div>
123:     </div>
124:
125:     { /* Meal Pattern 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">?? Meal Patt
128:         <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-6">
129:             {renderMacronutrientField(
130:                 "mealPattern",
131:                 "mealFrequency",
132:                 dietAnalysis.mealPattern.mealFrequency,
133:                 "Meal Frequency",
134:             )}
135:             {renderMacronutrientField(
136:                 "mealPattern",
137:                 "mealReplacement",
138:                 dietAnalysis.mealPattern.mealReplacement,
139:                 "Meal Replacement",
140:             )}
141:             {renderMacronutrientField(
142:                 "mealPattern",
143:                 "weightMaintenance",
144:                 dietAnalysis.mealPattern.weightMaintenance,
145:                 "Weight Maintenance",
146:             )}
147:         </div>
148:     </div>
149:
150:     { /* Food Sensitivities Section */}
151:     <div className="space-y-4">
152:         <h3 className="text-xl font-semibold text-gray-800 border-b-2 border-amber-200 pb-2">
153:             ? Food Sensitivities
154:         </h3>
155:         <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-6">
156:             {renderMacronutrientField(
157:                 "foodSensitivities",
158:                 "alcoholSensitivity",
159:                 dietAnalysis.foodSensitivities.alcoholSensitivity,
160:                 "Alcohol Sensitivity",
161:             )}
162:             {renderMacronutrientField(

```

```

163:         "foodSensitivities",
164:         "caffeineSensitivity",
165:         dietAnalysis.foodSensitivities.caffeineSensitivity,
166:         "Caffeine Sensitivity",
167:     )}
168:     {renderMacronutrientField(
169:         "foodSensitivities",
170:         "glutenSensitivity",
171:         dietAnalysis.foodSensitivities.glutenSensitivity,
172:         "Gluten Sensitivity",
173:     )}
174:     {renderMacronutrientField(
175:         "foodSensitivities",
176:         "lactoseSensitivity",
177:         dietAnalysis.foodSensitivities.lactoseSensitivity,
178:         "Lactose Sensitivity",
179:     )}
180:     {renderMacronutrientField(
181:         "foodSensitivities",
182:         "saltSensitivity",
183:         dietAnalysis.foodSensitivities.saltSensitivity,
184:         "Salt Sensitivity",
185:     )}
186: </div>
187: </div>
188:
189: {/* Taste Sensitivities Section */}
190: <div className="space-y-4">
191:   <h3 className="text-xl font-semibold text-gray-800 border-b-2 border-amber-200 pb-2">
192:     ? Taste Sensitivities
193:   </h3>
194:   <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-6">
195:     {renderMacronutrientField(
196:       "tasteSensitivities",
197:       "spiceTolerance",
198:       dietAnalysis.tasteSensitivities.spiceTolerance,
199:       "Spice Tolerance",
200:     )}
201:     {renderMacronutrientField(
202:       "tasteSensitivities",
203:       "sweetTaste",
204:       dietAnalysis.tasteSensitivities.sweetTaste,
205:       "Sweet Taste",
206:     )}
207:     {renderMacronutrientField(
208:       "tasteSensitivities",
209:       "tasteSensitivity",
210:       dietAnalysis.tasteSensitivities.tasteSensitivity,
211:       "Taste Sensitivity",
212:     )}
213:   </div>
214: </div>
215: </CardContent>
216: </Card>
217: )
218: }

```

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

```

1: "use client"
2:
3: import { Card, CardContent, CardDescription, 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 { Save, RotateCcw } from "lucide-react"
9: import type { DigestiveHealth, DigestiveHealthData } from "@/types/report-types"
10:
11: interface DigestiveHealthAdminProps {
12:   digestiveHealth: DigestiveHealth

```

```

13:   updateDigestiveHealth: (field: keyof DigestiveHealth, data: Partial<DigestiveHealthData>) => void
14:   onSave: () => void
15:   onReset: () => void
16: }
17:
18: export default function DigestiveHealthAdmin({
19:   digestiveHealth,
20:   updateDigestiveHealth,
21:   onSave,
22:   onReset,
23: }: DigestiveHealthAdminProps) {
24:   const renderDigestiveHealthField = (field: keyof DigestiveHealth, data: DigestiveHealthData, title: string) => {
25:     <Card key={field} className="border-2 border-red-100 bg-red-50/30">
26:       <CardHeader className="pb-3">
27:         <CardTitle className="text-lg text-red-800">{title}</CardTitle>
28:       </CardHeader>
29:       <CardContent className="space-y-4">
30:         <div className="space-y-2">
31:           <Label className="text-sm font-medium">Level</Label>
32:           <Input
33:             value={data.level}
34:             onChange={(e) => updateDigestiveHealth(field, { level: e.target.value })}
35:             placeholder="e.g., NORMAL GUT HEALTH"
36:             className="border-2 focus:border-red-500"
37:           />
38:         </div>
39:         <div className="space-y-2">
40:           <Label className="text-sm font-medium">Description</Label>
41:           <Textarea
42:             value={data.description}
43:             onChange={(e) => updateDigestiveHealth(field, { description: e.target.value })}
44:             rows={3}
45:             className="border-2 focus:border-red-500 text-sm"
46:           />
47:         </div>
48:       </CardContent>
49:     </Card>
50:   )
51:
52:   return (
53:     <Card className="shadow-lg border-0 bg-white">
54:       <CardHeader className="bg-gradient-to-r from-red-500 to-orange-600 text-white rounded-t-lg">
55:         <CardTitle className="text-2xl">? Digestive Health</CardTitle>
56:         <CardDescription className="text-red-100">
57:           Manage genetic influences on gut health, intolerances, and sensitivities
58:         </CardDescription>
59:       </CardHeader>
60:       <CardContent className="space-y-8 p-8">
61:         { /* Action Buttons */ }
62:         <div className="flex justify-end gap-2 mb-6">
63:           <Button onClick={onReset} variant="outline" size="sm">
64:             <RotateCcw className="h-4 w-4 mr-2" />
65:             Reset
66:           </Button>
67:           <Button onClick={onSave} size="sm">
68:             <Save className="h-4 w-4 mr-2" />
69:             Save Changes
70:           </Button>
71:         </div>
72:
73:         { /* Digestive Health Fields */ }
74:         <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-6">
75:           {Object.entries(digestiveHealth).map(([key, item]) =>
76:             renderDigestiveHealthField(
77:               key as keyof DigestiveHealth,
78:               item as DigestiveHealthData,
79:               key.replace(/[A-Z]/g, " $1").replace(/^./, (str) => str.toUpperCase()),
80:             ),
81:           )}
82:         </div>
83:
84:         { /* Section Guidelines */ }
85:         <div className="bg-red-50 border-2 border-red-200 rounded-lg p-6">

```

```

86:         <h4 className="font-semibold text-red-800 mb-2">? Digestive Health Guidelines:</h4>
87:         <div className="text-sm text-red-700 space-y-2">
88:             <ul className="list-disc list-inside space-y-1">
89:                 <li>
90:                     <strong>Gut Health:</strong> Overall genetic predisposition for gut function.
91:                 </li>
92:                 <li>
93:                     <strong>Gluten Intolerance:</strong> Genetic tendency towards gluten sensitivity.
94:                 </li>
95:                 <li>
96:                     <strong>Celiac Disease Tendency:</strong> Genetic risk for celiac disease.
97:                 </li>
98:                 <li>
99:                     <strong>Impact on Bones/Joints:</strong> How digestive health genetics might influence musc
100:                    health.
101:                 </li>
102:                 <li>
103:                     <strong>Lactose Intolerance:</strong> Genetic ability to digest lactose.
104:                 </li>
105:                 <li>
106:                     <strong>Probiotics Requirement:</strong> Indication for probiotic supplementation based on g
107:                 </li>
108:             </ul>
109:         </div>
110:     </div>
111: </CardContent>
112: </Card>
113: )
114: }

```

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

```

1: "use client"
2:
3: import { useState, useEffect } from "react"
4: import { DragDropContext, Droppable, Draggable } from "@hello-pangea/dnd"
5: import { v4 as uuidv4 } from "uuid"
6: import { Card, CardContent, CardDescription, CardHeader, CardTitle } from "@components/ui/card"
7: import { Button } from "@components/ui/button"
8: import { Plus, Save, RotateCcw, Trash2, GripVertical } from "lucide-react"
9: import { useToast } from "@components/ui/use-toast"
10: import { FieldInput, FieldTextarea, FieldSelect } from "@components/admin/dynamic-field-helpers"
11: import type { DynamicDietFieldDefinition } from "@types/report-types"
12:
13: interface DynamicDietFieldAdminProps {
14:     dynamicDietFieldDefinitions: DynamicDietFieldDefinition[]
15:     updateDynamicDietFieldDefinitions: (definitions: DynamicDietFieldDefinition[]) => void
16:     onSave: () => void
17:     onReset: () => void
18: }
19:
20: export default function DynamicDietFieldAdmin({
21:     dynamicDietFieldDefinitions,
22:     updateDynamicDietFieldDefinitions,
23:     onSave,
24:     onReset,
25: }: DynamicDietFieldAdminProps) {
26:     const [localFields, setLocalFields] = useState<DynamicDietFieldDefinition[]>(dynamicDietFieldDefinitions)
27:     const [categories, setCategories] = useState<string[]>([
28:         "Macronutrients",
29:         "Meal Pattern",
30:         "Food Sensitivities",
31:         "Taste Sensitivities",
32:     ]) // Predefined categories for diet analysis
33:     const [showCreateModal, setShowCreateModal] = useState(false)
34:     const [newFieldData, setNewFieldData] = useState<Omit<DynamicDietFieldDefinition, "_uuid" | "id">>({
35:         label: "",
36:         category: categories[0] || "", // Default to first category
37:         min: 1,
38:         max: 10,
39:         highRecommendation: "",

```

```

40:     normalRecommendation: "",
41:     lowRecommendation: "",
42: })
43: const { toast } = useToast()
44:
45: useEffect(() => {
46:     setLocalFields(dynamicDietFieldDefinitions)
47: }, [dynamicDietFieldDefinitions])
48:
49: const addNewField = () => {
50:     if (!newFieldData.label.trim()) {
51:         toast({
52:             title: "Validation Error",
53:             description: "Field label is required.",
54:             variant: "destructive",
55:         })
56:         return
57:     }
58:
59:     const fieldId = newFieldData.label
60:         .toLowerCase()
61:         .replace(/\s+/g, "_")
62:         .replace(/[^a-z0-9_]/g, "")
63:     const isDuplicate = localFields.some((field) => field.id === fieldId)
64:
65:     if (isDuplicate) {
66:         toast({
67:             title: "Validation Error",
68:             description: "A field with this label (ID) already exists.",
69:             variant: "destructive",
70:         })
71:         return
72:     }
73:
74:     const newField: DynamicDietFieldDefinition = {
75:         _uuid: uuidv4(),
76:         id: fieldId,
77:         ...newFieldData,
78:     }
79:
80:     const updatedFields = [...localFields, newField]
81:     setLocalFields(updatedFields)
82:     updateDynamicDietFieldDefinitions(updatedFields)
83:     setShowCreateModal(false)
84:     setNewFieldData({
85:         label: "",
86:         category: categories[0] || "",
87:         min: 1,
88:         max: 10,
89:         highRecommendation: "",
90:         normalRecommendation: "",
91:         lowRecommendation: "",
92:     })
93:     toast({
94:         title: "Field Added",
95:         description: `${newField.label} has been added. Remember to Save Changes.`,
96:     })
97: }
98:
99: const handleCreateModalClose = () => {
100:     setShowCreateModal(false)
101:     setNewFieldData({
102:         label: "",
103:         category: categories[0] || "",
104:         min: 1,
105:         max: 10,
106:         highRecommendation: "",
107:         normalRecommendation: "",
108:         lowRecommendation: "",
109:     })
110: }
111:
112: const updateLocalField = (index: number, key: keyof DynamicDietFieldDefinition, value: any) => {

```

```

113:     const updated = [...localFields]
114:     if (key === "id") {
115:         const newId = String(value)
116:         .toLowerCase()
117:         .replace(/s+/g, "_")
118:         .replace(/^[a-z0-9_]/g, "")
119:         const isDuplicate = localFields.some((f, i) => i !== index && f.id === newId)
120:         if (isDuplicate) {
121:             toast({
122:                 title: "Validation Error",
123:                 description: "Field ID must be unique.",
124:                 variant: "destructive",
125:             })
126:             return
127:         }
128:         updated[index] = { ...updated[index], [key]: newId }
129:     } else {
130:         updated[index] = { ...updated[index], [key]: value }
131:     }
132:     setLocalFields(updated)
133:     updateDynamicDietFieldDefinitions(updated) // Update parent state immediately
134: }
135:
136: const deleteField = (idToDelete: string) => {
137:     if (!window.confirm("Are you sure you want to delete this field? This action cannot be undone.")) {
138:         return
139:     }
140:     const updatedFields = localFields.filter((field) => field.id !== idToDelete)
141:     setLocalFields(updatedFields)
142:     updateDynamicDietFieldDefinitions(updatedFields)
143:     toast({
144:         title: "Field Deleted",
145:         description: "Field removed. Remember to Save Changes.",
146:         variant: "destructive",
147:     })
148: }
149:
150: const onDragEnd = (result: any) => {
151:     if (!result.destination) return
152:
153:     const reordered = Array.from(localFields)
154:     const [moved] = reordered.splice(result.source.index, 1)
155:     reordered.splice(result.destination.index, 0, moved)
156:
157:     setLocalFields(reordered)
158:     updateDynamicDietFieldDefinitions(reordered) // Update parent state
159:     toast({
160:         title: "Fields Reordered",
161:         description: "Order updated. Remember to Save Changes.",
162:     })
163: }
164:
165: return (
166:     <Card className="shadow-lg border-0 bg-white">
167:         <CardHeader className="bg-gradient-to-r from-amber-500 to-orange-600 text-white rounded-t-lg">
168:             <CardTitle className="text-2xl">? Dynamic Diet Field Management</CardTitle>
169:             <CardDescription className="text-amber-100">
170:                 Define and manage the types of diet analysis fields for your reports.
171:             </CardDescription>
172:         </CardHeader>
173:         <CardContent className="space-y-8 p-8">
174:             <div className="flex justify-end gap-2 mb-6">
175:                 <Button onClick={onReset} variant="outline" size="sm">
176:                     <RotateCcw className="h-4 w-4 mr-2" />
177:                     Reset Section
178:                 </Button>
179:                 <Button onClick={onSave} size="sm">
180:                     <Save className="h-4 w-4 mr-2" />
181:                     Save All Changes
182:                 </Button>
183:             </div>
184:
185:             <div className="flex justify-between items-center mb-6">

```

```

186:     <h3 className="text-xl font-semibold text-gray-800">Configured Fields</h3>
187:     <Button onClick={() => setShowCreateModal(true)} size="sm">
188:       <Plus className="h-4 w-4 mr-2" />
189:       Add New Field
190:     </Button>
191: </div>
192:
193: {localFields.length === 0 ? (
194:   <div className="text-center py-12 text-gray-500 italic">No dynamic diet fields defined yet.</div>
195: ) : (
196:   <DragDropContext onDragEnd={onDragEnd}>
197:     <Droppable droppableId="fields">
198:       {(provided) => (
199:         <div {...provided.droppableProps} ref={provided.innerRef} className="space-y-4">
200:           {localFields.map((field, index) => (
201:             <Draggable key={field._uuid} draggableId={field._uuid} index={index}>
202:               {(provided, snapshot) => (
203:                 <div
204:                   ref={provided.innerRef}
205:                   {...provided.draggableProps}
206:                   className={`border rounded-lg p-4 bg-white shadow-sm transition-all duration-200
207:                     ${snapshot.isDragging ? "bg-gray-100 ring-2 ring-blue-500 shadow-lg" : "hover:sha
208:                   `}
209:                 >
210:                   <div className="flex items-center justify-between mb-4">
211:                     <div className="flex items-center gap-2">
212:                       <span {...provided.dragHandleProps} className="cursor-grab text-gray-400">
213:                         <GripVertical className="h-5 w-5" />
214:                       </span>
215:                       <h4 className="text-lg font-semibold text-gray-900">
216:                         {index + 1}. {field.label} (ID: {field.id})
217:                       </h4>
218:                     </div>
219:                     <Button
220:                       variant="destructive"
221:                       size="sm"
222:                       onClick={() => deleteField(field.id)}
223:                       className="ml-4"
224:                     >
225:                       <Trash2 className="h-4 w-4" />
226:                     </Button>
227:                   </div>
228:
229:                   <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
230:                     <FieldInput
231:                       label="Field ID (Auto-generated from Label)"
232:                       value={field.id}
233:                       onChange={(val) => updateLocalField(index, "id", val)}
234:                       disabled // ID is derived from label, not directly editable
235:                     />
236:                     <FieldInput
237:                       label="Label"
238:                       value={field.label}
239:                       onChange={(val) => updateLocalField(index, "label", val)}
240:                     />
241:                     <FieldSelect
242:                       label="Category"
243:                       value={field.category}
244:                       options={categories}
245:                       onChange={(val) => updateLocalField(index, "category", val)}
246:                     />
247:                     <FieldInput
248:                       label="Min Score"
249:                       type="number"
250:                       value={field.min}
251:                       onChange={(val) => updateLocalField(index, "min", val)}
252:                       min={0}
253:                       max={field.max}
254:                     />
255:                     <FieldInput
256:                       label="Max Score"
257:                       type="number"
258:                       value={field.max}

```

```

259:             onChange={ (val) => updateLocalField(index, "max", val)}
260:             min={field.min}
261:         />
262:         <FieldTextarea
263:             label="High Recommendation"
264:             value={field.highRecommendation}
265:             onChange={ (val) => updateLocalField(index, "highRecommendation", val)}
266:         />
267:         <FieldTextarea
268:             label="Normal Recommendation"
269:             value={field.normalRecommendation}
270:             onChange={ (val) => updateLocalField(index, "normalRecommendation", val)}
271:         />
272:         <FieldTextarea
273:             label="Low Recommendation"
274:             value={field.lowRecommendation}
275:             onChange={ (val) => updateLocalField(index, "lowRecommendation", val)}
276:         />
277:     </div>
278: </div>
279:     )}
280: </Draggable>
281:   )}
282:   {provided.placeholder}
283: </div>
284: )}
285: </Droppable>
286: </DragDropContext>
287: )}
288: </CardContent>
289:
290: { /* Create New Field Modal */}
291: {showCreateModal && (
292:   <div className="fixed inset-0 bg-black bg-opacity-50 flex items-center justify-center z-50 p-4">
293:     <div className="bg-white rounded-lg p-6 w-full max-w-md max-h-[90vh] overflow-y-auto shadow-xl">
294:       <h3 className="text-xl font-bold mb-4">Add New Diet Field Definition</h3>
295:       <div className="space-y-4">
296:         <FieldInput
297:           label="Field Label"
298:           value={newFieldData.label}
299:           onChange={ (val) => setNewFieldData({ ...newFieldData, label: val })}
300:           placeholder="e.g., Gluten Sensitivity"
301:           required
302:         />
303:         <FieldSelect
304:           label="Category"
305:           value={newFieldData.category}
306:           options={categories}
307:           onChange={ (val) => setNewFieldData({ ...newFieldData, category: val })}
308:         />
309:         <div className="grid grid-cols-2 gap-4">
310:           <FieldInput
311:             label="Min Score"
312:             type="number"
313:             value={newFieldData.min}
314:             onChange={ (val) => setNewFieldData({ ...newFieldData, min: Number(val) })}
315:             min={0}
316:             max={newFieldData.max}
317:           />
318:           <FieldInput
319:             label="Max Score"
320:             type="number"
321:             value={newFieldData.max}
322:             onChange={ (val) => setNewFieldData({ ...newFieldData, max: Number(val) })}
323:             min={newFieldData.min}
324:           />
325:         </div>
326:         <FieldTextarea
327:           label="High Recommendation"
328:           value={newFieldData.highRecommendation}
329:           onChange={ (val) => setNewFieldData({ ...newFieldData, highRecommendation: val })}
330:           placeholder="Recommendation for high score"
331:         />

```



```

332:         <FieldTextarea
333:             label="Normal Recommendation"
334:             value={newFieldData.normalRecommendation}
335:             onChange={(val) => setNewFieldData({ ...newFieldData, normalRecommendation: val })}
336:             placeholder="Recommendation for normal score"
337:         />
338:         <FieldTextarea
339:             label="Low Recommendation"
340:             value={newFieldData.lowRecommendation}
341:             onChange={(val) => setNewFieldData({ ...newFieldData, lowRecommendation: val })}
342:             placeholder="Recommendation for low score"
343:         />
344:     </div>
345:     <div className="flex justify-end gap-3 mt-6">
346:         <Button variant="outline" onClick={handleCreateModalClose}>
347:             Cancel
348:         </Button>
349:         <Button onClick={addNewField}>Add Field</Button>
350:     </div>
351: </div>
352: </div>
353:     )}
354: </Card>
355: )
356: }

```

■ 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 ?? ""}
41:     </input>
42:   </div>
43: )

```

```

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
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
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 { Card, CardContent, CardDescription, 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 { Save, RotateCcw, Plus, Trash2 } from "lucide-react"
9: import type { FamilyGeneticImpact } from "@types/report-types"
10:
11: interface FamilyGeneticImpactAdminProps {
12:   familyGeneticImpact: FamilyGeneticImpact[]
13:   addFamilyGeneticImpact: () => void
14:   updateFamilyGeneticImpact: (index: number, field: keyof FamilyGeneticImpact, value: string) => void
15:   removeFamilyGeneticImpact: (index: number) => void
16:   onSave: () => void
17:   onReset: () => void
18: }
19:
20: export default function FamilyGeneticImpactAdmin({
21:   familyGeneticImpact,
22:   addFamilyGeneticImpact,
23:   updateFamilyGeneticImpact,
24:   removeFamilyGeneticImpact,

```

```

25:   onSave,
26:   onReset,
27: }: FamilyGeneticImpactAdminProps) {
28:   return (
29:     <Card className="shadow-lg border-0 bg-white">
30:       <CardHeader className="bg-gradient-to-r from-blue-500 to-cyan-600 text-white rounded-t-lg">
31:         <CardTitle className="text-2xl">?????? Family Genetic Impact</CardTitle>
32:         <CardDescription className="text-blue-100">
33:           Manage genetic variations that may impact family members
34:         </CardDescription>
35:       </CardHeader>
36:       <CardContent className="space-y-8 p-8">
37:         { /* Action Buttons */ }
38:         <div className="flex justify-between items-center mb-6">
39:           <Button onClick={addFamilyGeneticImpact} size="sm">
40:             <Plus className="h-4 w-4 mr-2" />
41:             Add Genetic Impact
42:           </Button>
43:           <div className="flex gap-2">
44:             <Button onClick={onReset} variant="outline" size="sm">
45:               <RotateCcw className="h-4 w-4 mr-2" />
46:               Reset
47:             </Button>
48:             <Button onClick={onSave} size="sm">
49:               <Save className="h-4 w-4 mr-2" />
50:               Save Changes
51:             </Button>
52:           </div>
53:         </div>
54:
55:         { /* Family Genetic Impact List */ }
56:         <div className="space-y-6">
57:           {familyGeneticImpact.length === 0 && (
58:             <div className="text-center py-12 text-gray-500">
59:               <div className="text-6xl mb-4">?</div>
60:               <p className="text-lg mb-4">No family genetic impacts added yet</p>
61:               <Button onClick={addFamilyGeneticImpact} className="bg-blue-600 hover:bg-blue-700">
62:                 <Plus className="h-4 w-4 mr-2" />
63:                 Add First Impact
64:               </Button>
65:             </div>
66:           )}
67:
68:           {familyGeneticImpact.map((impact, index) => (
69:             <Card key={index} className="border-2 border-blue-100 bg-blue-50/30">
70:               <CardHeader className="pb-3">
71:                 <CardTitle className="text-lg text-blue-800 flex justify-between items-center">
72:                   Genetic Impact #{index + 1}
73:                   <Button onClick={() => removeFamilyGeneticImpact(index)} size="sm" variant="destructive">
74:                     <Trash2 className="h-4 w-4" />
75:                   </Button>
76:                 </CardTitle>
77:               </CardHeader>
78:               <CardContent className="space-y-4">
79:                 <div className="space-y-2">
80:                   <Label className="text-sm font-medium">Gene</Label>
81:                   <Input
82:                     value={impact.gene}
83:                     onChange={(e) => updateFamilyGeneticImpact(index, "gene", e.target.value)}
84:                     placeholder="e.g., MnSOD"
85:                     className="border-2 focus:border-blue-500 font-mono"
86:                   />
87:                 </div>
88:                 <div className="space-y-2">
89:                   <Label className="text-sm font-medium">Normal/Common Alleles</Label>
90:                   <Input
91:                     value={impact.normalAlleles}
92:                     onChange={(e) => updateFamilyGeneticImpact(index, "normalAlleles", e.target.value)}
93:                     placeholder="e.g., AA"
94:                     className="border-2 focus:border-blue-500 font-mono"
95:                   />
96:                 </div>
97:               </div>
98:             </div>
99:           ))}
100:         </div>
101:       </CardContent>
102:     </Card>
103:   )
104: }

```

```

98:         <Label className="text-sm font-medium">Your Result</Label>
99:         <Input
100:             value={impact.yourResult}
101:             onChange={(e) => updateFamilyGeneticImpact(index, "yourResult", e.target.value)}
102:             placeholder="e.g., AG"
103:             className="border-2 focus:border-blue-500 font-mono font-bold"
104:         />
105:     </div>
106:     <div className="space-y-2">
107:         <Label className="text-sm font-medium">Impact on Health</Label>
108:         <Textarea
109:             value={impact.healthImpact}
110:             onChange={(e) => updateFamilyGeneticImpact(index, "healthImpact", e.target.value)}
111:             rows={3}
112:             placeholder="e.g., INCREASED TENDENCY OF OXIDATIVE STRESS NEED OF ANTI-OXIDANT SUPPLEMEN
113:             className="border-2 focus:border-blue-500 text-sm"
114:         />
115:     </div>
116: </CardContent>
117: </Card>
118:   )}
119: </div>
120:
121:   { /* Section Guidelines */ }
122:   <div className="bg-blue-50 border-2 border-blue-200 rounded-lg p-6">
123:     <h4 className="font-semibold text-blue-800 mb-2">? Family Genetic Impact Guidelines:</h4>
124:     <div className="text-sm text-blue-700 space-y-2">
125:       <ul className="list-disc list-inside space-y-1">
126:         <li>
127:           This section highlights genetic variations observed in the patient that may also be present
128:           family members (parents, siblings, children).
129:         </li>
130:         <li>
131:           It serves as a recommendation for family members to consider genetic testing for similar
132:           predispositions.
133:         </li>
134:         <li>
135:           <strong>Gene:</strong> The specific gene with a noted variation.
136:         </li>
137:         <li>
138:           <strong>Normal/Common Alleles:</strong> The typical genetic variants.
139:         </li>
140:         <li>
141:           <strong>Your Result:</strong> The patient's specific genetic variant.
142:         </li>
143:         <li>
144:           <strong>Impact on Health:</strong> The health implications associated with the patient's ge
145:           which could also apply to family members.
146:         </li>
147:       </ul>
148:     </div>
149:   </div>
150:   </CardContent>
151: </Card>
152: )
153: }

```

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

```

=====
1: "use client"
2:
3: import { Button } from "@/components/ui/button"
4: import { Card, CardContent, CardDescription, CardHeader, CardTitle } from "@/components/ui/card"
5: import { Input } from "@/components/ui/input"
6: import { Label } from "@/components/ui/label"
7: import { ScrollArea } from "@/components/ui/scroll-area"
8: import { PlusCircle, Trash2, Save, RotateCcw } from "lucide-react"
9: import type { GeneTestResult } from "@/types/report-types"
10:
11: interface GeneResultsAdminProps {
12:   geneTestResults: GeneTestResult[]

```

```

13:   addGeneTestResult: () => void
14:   updateGeneTestResult: (index: number, field: keyof GeneTestResult, value: string) => void
15:   removeGeneTestResult: (index: number) => void
16:   onSave: () => void
17:   onReset: () => void
18: }
19:
20: export default function GeneResultsAdmin({
21:   geneTestResults,
22:   addGeneTestResult,
23:   updateGeneTestResult,
24:   removeGeneTestResult,
25:   onSave,
26:   onReset,
27: }: GeneResultsAdminProps) {
28:   return (
29:     <Card className="shadow-lg border-0 bg-white">
30:       <CardHeader className="bg-gradient-to-r from-indigo-500 to-blue-600 text-white rounded-t-lg">
31:         <CardTitle className="text-2xl">? Gene Test Results</CardTitle>
32:         <CardDescription className="text-indigo-100">Manage the complete list of gene test results</CardDescription>
33:       </CardHeader>
34:       <CardContent className="space-y-6 p-8">
35:         <div className="flex justify-between items-center">
36:           <h3 className="text-lg font-semibold">Gene Test Results ({geneTestResults.length})</h3>
37:           <div className="flex gap-2">
38:             <Button onClick={onReset} variant="outline" size="sm">
39:               <RotateCcw className="h-4 w-4 mr-2" />
40:               Reset
41:             </Button>
42:             <Button onClick={onSave} size="sm">
43:               <Save className="h-4 w-4 mr-2" />
44:               Save Changes
45:             </Button>
46:             <Button onClick={addGeneTestResult} className="bg-indigo-600 hover:bg-indigo-700">
47:               <PlusCircle className="h-4 w-4 mr-2" />
48:               Add Gene Result
49:             </Button>
50:           </div>
51:         </div>
52:
53:         <ScrollArea className="h-96">
54:           <div className="space-y-4">
55:             {geneTestResults.map((result, index) => (
56:               <div key={index} className="border-2 border-gray-200 rounded-lg p-4 bg-gray-50">
57:                 <div className="flex justify-between items-start mb-4">
58:                   <h4 className="font-semibold text-gray-800">Gene #{index + 1}</h4>
59:                   <Button onClick={() => removeGeneTestResult(index)} variant="destructive" size="sm">
60:                     <Trash2 className="h-4 w-4" />
61:                   </Button>
62:                 </div>
63:                 <div className="grid md:grid-cols-3 gap-4">
64:                   <div className="space-y-2">
65:                     <Label className="text-sm font-medium">Gene Name</Label>
66:                     <Input
67:                       value={result.geneName}
68:                       onChange={(e) => updateGeneTestResult(index, "geneName", e.target.value)}
69:                       placeholder="e.g., ALDH2 Aldehyde dehydrogenase 2"
70:                       className="border-2 focus:border-indigo-500"
71:                     />
72:                   </div>
73:                   <div className="space-y-2">
74:                     <Label className="text-sm font-medium">Variation</Label>
75:                     <Input
76:                       value={result.variation}
77:                       onChange={(e) => updateGeneTestResult(index, "variation", e.target.value)}
78:                       placeholder="e.g., G>A"
79:                       className="border-2 focus:border-indigo-500 font-mono"
80:                     />
81:                   </div>
82:                   <div className="space-y-2">
83:                     <Label className="text-sm font-medium">Result</Label>
84:                     <Input
85:                       value={result.result}

```

```

86:             onChange={(e) => updateGeneTestResult(index, "result", e.target.value)}
87:             placeholder="e.g., GG"
88:             className="border-2 focus:border-indigo-500 font-mono font-bold"
89:         />
90:     </div>
91: </div>
92: </div>
93: )}}
94: </div>
95: </ScrollArea>
96:
97: {geneTestResults.length === 0 && (
98:   <div className="text-center py-12 text-gray-500">
99:     <div className="text-6xl mb-4">?</div>
100:    <p className="text-lg mb-4">No gene test results added yet</p>
101:    <Button onClick={addGeneTestResult} className="bg-indigo-600 hover:bg-indigo-700">
102:      <PlusCircle className="h-4 w-4 mr-2" />
103:      Add First Gene Result
104:    </Button>
105:  </div>
106: )}
107: </CardContent>
108: </Card>
109: )
110: }

```

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

```

=====
1: "use client"
2:
3: import { Card, CardContent, CardDescription, 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 { Save, RotateCcw } from "lucide-react"
9: import type { GenesAndAddiction, AdditionData } from "@types/report-types"
10:
11: interface GenesAddictionAdminProps {
12:   genesAndAddiction: GenesAndAddiction
13:   updateGenesAndAddiction: (field: keyof GenesAndAddiction, data: Partial<AdditionData>) => void
14:   onSave: () => void
15:   onReset: () => void
16: }
17:
18: export default function GenesAddictionAdmin({
19:   genesAndAddiction,
20:   updateGenesAndAddiction,
21:   onSave,
22:   onReset,
23: }: GenesAddictionAdminProps) {
24:   const renderAdditionField = (field: keyof GenesAndAddiction, data: AdditionData, title: string) => (
25:     <Card key={field} className="border-2 border-purple-100 bg-purple-50/30">
26:       <CardHeader className="pb-3">
27:         <CardTitle className="text-lg text-purple-800">{title}</CardTitle>
28:       </CardHeader>
29:       <CardContent className="space-y-4">
30:         <div className="space-y-2">
31:           <Label className="text-sm font-medium">Tendency</Label>
32:           <Input
33:             value={data.tendency}
34:             onChange={(e) => updateGenesAndAddiction(field, { tendency: e.target.value })}
35:             placeholder="e.g., LOW, NORMAL, HIGH"
36:             className="border-2 focus:border-purple-500"
37:           />
38:         </div>
39:         <div className="space-y-2">
40:           <Label className="text-sm font-medium">Description</Label>
41:           <Textarea
42:             value={data.description}
43:             onChange={(e) => updateGenesAndAddiction(field, { description: e.target.value })}

```

```

44:         rows={3}
45:         className="border-2 focus:border-purple-500 text-sm"
46:       />
47:     </div>
48:   </CardContent>
49: </Card>
50: )
51:
52: return (
53:   <Card className="shadow-lg border-0 bg-white">
54:     <CardHeader className="bg-gradient-to-r from-purple-500 to-pink-600 text-white rounded-t-lg">
55:       <CardTitle className="text-2xl">? Genes & Addiction</CardTitle>
56:       <CardDescription className="text-purple-100">
57:         Manage genetic predispositions related to various addiction tendencies
58:       </CardDescription>
59:     </CardHeader>
60:     <CardContent className="space-y-8 p-8">
61:       {/* Action Buttons */}
62:       <div className="flex justify-end gap-2 mb-6">
63:         <Button onClick={onReset} variant="outline" size="sm">
64:           <RotateCcw className="h-4 w-4 mr-2" />
65:           Reset
66:         </Button>
67:         <Button onClick={onSave} size="sm">
68:           <Save className="h-4 w-4 mr-2" />
69:           Save Changes
70:         </Button>
71:       </div>
72:
73:       {/* Genes & Addiction Fields */}
74:       <div className="grid md:grid-cols-2 lg:grid-cols-2 gap-6">
75:         {Object.entries(genesAndAddiction).map(([key, item]) =>
76:           renderAddictionField(
77:             key as keyof GenesAndAddiction,
78:             item as AdditionData,
79:             key.replace(/[A-Z])/g, " $1").replace(/^.\/, (str) => str.toUpperCase()),
80:         ),
81:       )}
82:     </div>
83:
84:     {/* Section Guidelines */}
85:     <div className="bg-purple-50 border-2 border-purple-200 rounded-lg p-6">
86:       <h4 className="font-semibold text-purple-800 mb-2">? Genes & Addiction Guidelines:</h4>
87:       <div className="text-sm text-purple-700 space-y-2">
88:         <ul className="list-disc list-inside space-y-1">
89:           <li>
90:             <strong>Alcohol Addiction:</strong> Genetic vulnerability to alcohol dependency.
91:           </li>
92:           <li>
93:             <strong>Smoking Addiction:</strong> Genetic susceptibility to nicotine addiction.
94:           </li>
95:           <li>
96:             <strong>Snacking Habits:</strong> Genetic tendency towards frequent snacking or specific food preferences.
97:           </li>
98:           <li>
99:             <strong>Altruism:</strong> Genetic predisposition towards prosocial behaviors and helping others.
100:          </li>
101:          <li>Genetics influence vulnerability, but environmental factors and lifestyle choices are crucial.
102:          <li>This information can help tailor prevention strategies and personalized treatment plans.</li>
103:        </ul>
104:      </div>
105:    </div>
106:  </CardContent>
107: </Card>
108: )
109: }

```

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

```

=====
1: "use client"
2:

```

```

3: import type React from "react"
4: import { Button } from "@/components/ui/button"
5: import { Card, CardContent, CardDescription, CardHeader, CardTitle } from "@/components/ui/card"
6: import { Label } from "@/components/ui/label"
7: import { Download, Upload } from "lucide-react"
8: import type { ComprehensiveReportData } from "@/types/report-types"
9: import { useToast } from "@/components/ui/use-toast"
10:
11: interface ImportExportAdminProps {
12:   reportData: ComprehensiveReportData
13:   setReportData: (data: ComprehensiveReportData) => void
14: }
15:
16: export default function ImportExportAdmin({ reportData, setReportData }: ImportExportAdminProps) {
17:   const { toast } = useToast()
18:
19:   const exportData = () => {
20:     try {
21:       const dataStr = JSON.stringify(reportData, null, 2)
22:       const dataBlob = new Blob([dataStr], { type: "application/json" })
23:       const url = URL.createObjectURL(dataBlob)
24:       const link = document.createElement("a")
25:       link.href = url
26:       link.download = `genomics-report-${new Date().toISOString().split("T")[0]}.json`
27:       link.click()
28:       URL.revokeObjectURL(url)
29:       toast({
30:         title: "Export Successful!",
31:         description: "Report data has been downloaded as a JSON file.",
32:       })
33:     } catch (error) {
34:       console.error("Error exporting data:", error)
35:       toast({
36:         title: "Export Failed!",
37:         description: "There was an error exporting the data. Please try again.",
38:         variant: "destructive",
39:       })
40:     }
41:   }
42:
43:   const importData = (event: React.ChangeEvent<HTMLInputElement>) => {
44:     const file = event.target.files?.[0]
45:     if (file) {
46:       const reader = new FileReader()
47:       reader.onload = (e) => {
48:         try {
49:           const importedData = JSON.parse(e.target?.result as string) as ComprehensiveReportData
50:           // Basic validation to ensure it's a report data structure
51:           if (importedData && importedData.patientInfo && importedData.content) {
52:             setReportData(importedData)
53:             toast({
54:               title: "Import Successful!",
55:               description: "Report data has been imported successfully.",
56:             })
57:           } else {
58:             throw new Error("Invalid file format or missing essential data.")
59:           }
60:         } catch (error: any) {
61:           console.error("Error importing data:", error)
62:           toast({
63:             title: "Import Failed!",
64:             description: `Error importing data: ${error.message || "Please check the file format."}`,
65:             variant: "destructive",
66:           })
67:         }
68:       }
69:       reader.readAsText(file)
70:     }
71:   }
72:
73:   return (
74:     <Card className="shadow-lg border-0 bg-white">
75:       <CardHeader className="bg-gradient-to-r from-orange-500 to-red-600 text-white rounded-t-lg">

```



```

76:         <CardTitle className="text-2xl">? Import/Export Data</CardTitle>
77:         <CardDescription className="text-orange-100">Backup and restore your report data</CardDescription>
78:     </CardHeader>
79:     <CardContent className="space-y-8 p-8">
80:         <div className="grid md:grid-cols-2 gap-8">
81:             <div className="space-y-4">
82:                 <h3 className="text-xl font-semibold text-gray-800">? Export Data</h3>
83:                 <p className="text-gray-600">
84:                     Download your current report configuration as a JSON file for backup or sharing.
85:                 </p>
86:                 <Button onClick={exportData} className="w-full bg-orange-600 hover:bg-orange-700">
87:                     <Download className="h-4 w-4 mr-2" />
88:                     Export Current Data
89:                 </Button>
90:             </div>
91:
92:             <div className="space-y-4">
93:                 <h3 className="text-xl font-semibold text-gray-800">? Import Data</h3>
94:                 <p className="text-gray-600">Upload a previously exported JSON file to restore report configura
95:                 <div className="border-2 border-dashed border-gray-300 rounded-lg p-6 text-center">
96:                     <input type="file" accept=".json" onChange={importData} className="hidden" id="import-file" />
97:                     <Label htmlFor="import-file" className="cursor-pointer">
98:                         <Upload className="h-8 w-8 mx-auto mb-2 text-gray-400" />
99:                         <span className="text-sm text-gray-600">Click to select JSON file</span>
100:                     </Label>
101:                 </div>
102:             </div>
103:         </div>
104:
105:         <div className="bg-yellow-50 border-2 border-yellow-200 rounded-lg p-6">
106:             <h4 className="font-semibold text-yellow-800 mb-2">?? Important Notes:</h4>
107:             <ul className="text-sm text-yellow-700 space-y-1">
108:                 <li>? Importing data will completely replace your current configuration</li>
109:                 <li>? Always export your current data before importing new data</li>
110:                 <li>? Only import JSON files that were exported from this system</li>
111:                 <li>? Make sure to save your changes after importing</li>
112:             </ul>
113:         </div>
114:     </CardContent>
115: </Card>
116: )
117: }

```

File: components\admin\lifestyle-conditions-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 { Select, SelectContent, SelectItem, SelectTrigger, SelectValue } from "@components/ui/select"
6: import { Button } from "@components/ui/button"
7: import { Save, RotateCcw } from "lucide-react"
8: import { useState } from "react"
9: import type { LifestyleConditions, HealthConditionStatus } from "@types/report-types"
10:
11: interface LifestyleConditionsAdminProps {
12:   lifestyleConditions: LifestyleConditions
13:   updateLifestyleCondition: (
14:     section: keyof LifestyleConditions,
15:     field: string,
16:     status: HealthConditionStatus["status"],
17:   ) => void
18:   onSave: () => void
19:   onReset: () => void
20: }
21:
22: export default function LifestyleConditionsAdmin({
23:   lifestyleConditions,
24:   updateLifestyleCondition,
25:   onSave,
26:   onReset,

```

```

27: }: LifestyleConditionsAdminProps) {
28:   const [activeSection, setActiveSection] = useState<keyof LifestyleConditions>("heartVascularHealth")
29:
30:   const renderConditionField = (
31:     section: keyof LifestyleConditions,
32:     field: string,
33:     data: HealthConditionStatus,
34:     title: string,
35:   ) => (
36:     <div key={field} className="grid grid-cols-3 gap-4 items-center border-b pb-2">
37:       <Label className="font-medium text-gray-700">{title}</Label>
38:       <div className="col-span-2">
39:         <Select
40:           value={data.status}
41:           onChange={(value: HealthConditionStatus["status"]) => updateLifestyleCondition(section, field, value)}
42:         >
43:           <SelectTrigger className="border-2 focus:border-yellow-500">
44:             <SelectValue />
45:           </SelectTrigger>
46:           <SelectContent>
47:             <SelectItem value="strength">Strength (?)</SelectItem>
48:             <SelectItem value="improvement">Improvement Needed (?)</SelectItem>
49:           </SelectContent>
50:         </Select>
51:       </div>
52:     </div>
53:   )
54:
55:   const getSectionTitle = (section: keyof LifestyleConditions) => {
56:     switch (section) {
57:       case "heartVascularHealth":
58:         return "?? Heart & Vascular Health"
59:       case "diabetesity":
60:         return "? Diabetesity (Diabetes & Obesity)"
61:       case "liverHealth":
62:         return "? Liver Health"
63:       case "boneHealth":
64:         return "? Bone Health"
65:       case "gutHealth":
66:         return "? Gut Health"
67:       case "hormoneSystem":
68:         return "? Hormone System"
69:       default:
70:         return section
71:     }
72:   }
73:
74:   const getSectionIcon = (section: keyof LifestyleConditions) => {
75:     switch (section) {
76:       case "heartVascularHealth":
77:         return "???"
78:       case "diabetesity":
79:         return "?"
80:       case "liverHealth":
81:         return "?"
82:       case "boneHealth":
83:         return "?"
84:       case "gutHealth":
85:         return "?"
86:       case "hormoneSystem":
87:         return "?"
88:       default:
89:         return "?"
90:     }
91:   }
92:
93:   return (
94:     <Card className="shadow-lg border-0 bg-white">
95:       <CardHeader className="bg-gradient-to-r from-yellow-500 to-orange-600 text-white rounded-t-lg">
96:         <CardTitle className="text-2xl">? Lifestyle Conditions</CardTitle>
97:         <CardDescription className="text-yellow-100">
98:           Manage genetic predispositions for common lifestyle-related health conditions
99:         </CardDescription>

```

```

100:     </CardHeader>
101:     <CardContent className="space-y-8 p-8">
102:         {/* Action Buttons */}
103:         <div className="flex justify-between items-center">
104:             <div className="flex gap-2 flex-wrap">
105:                 {(Object.keys(lifestyleConditions) as Array<keyof LifestyleConditions>).map((section) => (
106:                     <Button
107:                         key={section}
108:                         variant={activeSection === section ? "default" : "outline"}
109:                         size="sm"
110:                         onClick={() => setActiveSection(section)}
111:                         className="flex items-center gap-2"
112:                     >
113:                         <span>{getSectionIcon(section)}</span>
114:                         <span className="capitalize">{section.replace(/[A-Z]/g, " $1")}</span>
115:                     </Button>
116:                 ))}
117:             </div>
118:             <div className="flex gap-2">
119:                 <Button onClick={onReset} variant="outline" size="sm">
120:                     <RotateCcw className="h-4 w-4 mr-2" />
121:                     Reset
122:                 </Button>
123:                 <Button onClick={onSave} size="sm">
124:                     <Save className="h-4 w-4 mr-2" />
125:                     Save Changes
126:                 </Button>
127:             </div>
128:         </div>
129:
130:         {/* Section Content */}
131:         <div className="space-y-6">
132:             <h3 className="text-xl font-semibold text-gray-800 border-b-2 border-yellow-200 pb-2">
133:                 {getSectionTitle(activeSection)}
134:             </h3>
135:
136:             <div className="grid md:grid-cols-2 gap-6">
137:                 {Object.entries(lifestyleConditions[activeSection]).map(([key, item]) =>
138:                     renderConditionField(
139:                         activeSection,
140:                         key,
141:                         item as HealthConditionStatus,
142:                         key.replace(/[A-Z]/g, " $1").replace(/^../, (str) => str.toUpperCase()),
143:                     ),
144:                 )}
145:             </div>
146:         </div>
147:
148:         {/* Section Guidelines */}
149:         <div className="bg-yellow-50 border-2 border-yellow-200 rounded-lg p-6">
150:             <h4 className="font-semibold text-yellow-800 mb-2">? Lifestyle Conditions Guidelines:</h4>
151:             <div className="text-sm text-yellow-700 space-y-2">
152:                 <ul className="list-disc list-inside space-y-1">
153:                     <li>
154:                         <strong>Strength (?):</strong> Indicates a genetic predisposition that is favorable or protective
155:                         against the condition.
156:                     </li>
157:                     <li>
158:                         <strong>Improvement Needed (?):</strong> Indicates a genetic predisposition that suggests a
159:                         or vulnerability to the condition, requiring proactive lifestyle interventions.
160:                     </li>
161:                     <li>
162:                         This section highlights areas where personalized interventions (diet, exercise, screenings)
163:                         impactful.
164:                     </li>
165:                     <li>Genetic predispositions are not deterministic; lifestyle and environment play crucial roles.
166:                     </li>
167:                 </ul>
168:             </div>
169:         </div>
170:
171:         {/* Summary Statistics */}
172:         <div className="grid md:grid-cols-2 gap-4">
173:             <div className="bg-green-50 border border-green-200 rounded-lg p-4 text-center">

```

```

173:         <div className="text-2xl font-bold text-green-600">
174:             {
175:                 Object.values(lifestyleConditions[activeSection]).filter((item: any) => item.status === "st
176:                 .length
177:             }
178:         </div>
179:         <div className="text-sm text-green-700">Your Strengths</div>
180:     </div>
181:     <div className="bg-red-50 border border-red-200 rounded-lg p-4 text-center">
182:         <div className="text-2xl font-bold text-red-600">
183:             {
184:                 Object.values(lifestyleConditions[activeSection]).filter((item: any) => item.status === "imp
185:                 .length
186:             }
187:         </div>
188:         <div className="text-sm text-red-700">Improvement Needed</div>
189:     </div>
190: </div>
191: </CardContent>
192: </Card>
193: )
194: }

```

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

```

=====
1: "use client"
2:
3: import { Card, CardContent, CardDescription, 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 { Badge } from "@components/ui/badge"
9: import { Save, RotateCcw, Plus, Trash2, Copy } from "lucide-react"
10: import { useState } from "react"
11: import type { MetabolicCore, MetabolicGeneData } from "@types/report-types"
12:
13: interface MetabolicCoreAdminProps {
14:     metabolicCore: MetabolicCore
15:     updateMetabolicCore: (field: keyof MetabolicCore, data: Partial<MetabolicGeneData>) => void
16:     onSave: () => void
17:     onReset: () => void
18: }
19:
20: export default function MetabolicCoreAdmin({
21:     metabolicCore,
22:     updateMetabolicCore,
23:     onSave,
24:     onReset,
25: }: MetabolicCoreAdminProps) {
26:     const [searchTerm, setSearchTerm] = useState("")
27:
28:     const metabolicAreas = Object.keys(metabolicCore) as Array<keyof MetabolicCore>
29:
30:     const filteredAreas = metabolicAreas.filter((area) => {
31:         const data = metabolicCore[area]
32:         const matchesSearch =
33:             area.toLowerCase().includes(searchTerm.toLowerCase()) ||
34:             data.impact.toLowerCase().includes(searchTerm.toLowerCase()) ||
35:             data.advice.toLowerCase().includes(searchTerm.toLowerCase()) ||
36:             data.genes.some((gene) => gene.toLowerCase().includes(searchTerm.toLowerCase()))
37:         return matchesSearch
38:     })
39:
40:     const getImpactColor = (impact: string) => {
41:         if (impact.toLowerCase().includes("high") || impact.toLowerCase().includes("increased")) {
42:             return "bg-red-100 border-red-300 text-red-800"
43:         }
44:         if (impact.toLowerCase().includes("low") || impact.toLowerCase().includes("reduced")) {
45:             return "bg-green-100 border-green-300 text-green-800"
46:         }

```

```

47:     if (
48:         impact.toLowerCase().includes("normal") ||
49:         impact.toLowerCase().includes("average") ||
50:         impact.toLowerCase().includes("balanced")
51:     ) {
52:         return "bg-blue-100 border-blue-300 text-blue-800"
53:     }
54:     return "bg-gray-100 border-gray-300 text-gray-800"
55: }
56:
57: const getImpactIcon = (impact: string) => {
58:     if (impact.toLowerCase().includes("high") || impact.toLowerCase().includes("increased")) {
59:         return "???"
60:     }
61:     if (impact.toLowerCase().includes("low") || impact.toLowerCase().includes("reduced")) {
62:         return "???"
63:     }
64:     if (
65:         impact.toLowerCase().includes("normal") ||
66:         impact.toLowerCase().includes("average") ||
67:         impact.toLowerCase().includes("balanced")
68:     ) {
69:         return "???"
70:     }
71:     return "???"
72: }
73:
74: const addGene = (area: keyof MetabolicCore) => {
75:     const currentGenes = metabolicCore[area].genes
76:     updateMetabolicCore(area, { genes: [...currentGenes, ""] })
77: }
78:
79: const removeGene = (area: keyof MetabolicCore, index: number) => {
80:     const currentGenes = metabolicCore[area].genes
81:     const newGenes = currentGenes.filter((_, i) => i !== index)
82:     updateMetabolicCore(area, { genes: newGenes })
83: }
84:
85: const updateGene = (area: keyof MetabolicCore, index: number, value: string) => {
86:     const currentGenes = [...metabolicCore[area].genes]
87:     currentGenes[index] = value
88:     updateMetabolicCore(area, { genes: currentGenes })
89: }
90:
91: const copyToClipboard = (text: string) => {
92:     navigator.clipboard.writeText(text)
93:     // You could add a toast notification here
94: }
95:
96: return (
97:     <Card className="shadow-lg border-0 bg-white">
98:         <CardHeader className="bg-gradient-to-r from-purple-500 to-indigo-600 text-white rounded-t-lg">
99:             <CardTitle className="text-2xl">? Metabolic Core Analysis</CardTitle>
100:             <CardDescription className="text-purple-100">
101:                 Comprehensive gene analysis and metabolic impacts - The STAR feature of genomics reporting
102:             </CardDescription>
103:         </CardHeader>
104:         <CardContent className="space-y-8 p-8">
105:             { /* Action Bar */ }
106:             <div className="flex flex-col md:flex-row justify-between items-start md:items-center gap-4">
107:                 <div className="flex-1">
108:                     <Input
109:                         placeholder="Search metabolic areas or impacts..."
110:                         value={searchTerm}
111:                         onChange={(e) => setSearchTerm(e.target.value)}
112:                         className="border-2 focus:border-purple-500"
113:                     />
114:                 </div>
115:                 <div className="flex gap-2">
116:                     <Button onClick={onReset} variant="outline" size="sm">
117:                         <RotateCcw className="h-4 w-4 mr-2" />
118:                         Reset
119:                     </Button>

```

```

120:         <Button onClick={onSave} size="sm">
121:             <Save className="h-4 w-4 mr-2" />
122:             Save Changes
123:         </Button>
124:     </div>
125: </div>
126:
127: { /* Summary Statistics */}
128: <div className="grid md:grid-cols-4 gap-4">
129:     <div className="bg-red-50 border border-red-200 rounded-lg p-4 text-center">
130:         <div className="text-2xl font-bold text-red-600">
131:             {
132:                 metabolicAreas.filter(
133:                     (area) =>
134:                         metabolicCore[area].impact.toLowerCase().includes("high") ||
135:                         metabolicCore[area].impact.toLowerCase().includes("increased"),
136:                 ).length
137:             }
138:         </div>
139:         <div className="text-sm text-red-700">High Risk Areas</div>
140:     </div>
141:     <div className="bg-green-50 border border-green-200 rounded-lg p-4 text-center">
142:         <div className="text-2xl font-bold text-green-600">
143:             {
144:                 metabolicAreas.filter(
145:                     (area) =>
146:                         metabolicCore[area].impact.toLowerCase().includes("normal") ||
147:                         metabolicCore[area].impact.toLowerCase().includes("low") ||
148:                         metabolicCore[area].impact.toLowerCase().includes("reduced") ||
149:                         metabolicCore[area].impact.toLowerCase().includes("balanced"),
150:                 ).length
151:             }
152:         </div>
153:         <div className="text-sm text-green-700">Normal/Low Risk</div>
154:     </div>
155:     <div className="bg-blue-50 border border-blue-200 rounded-lg p-4 text-center">
156:         <div className="text-2xl font-bold text-blue-600">
157:             {metabolicAreas.reduce((total, area) => total + metabolicCore[area].genes.length, 0)}
158:         </div>
159:         <div className="text-sm text-blue-700">Total Genes</div>
160:     </div>
161:     <div className="bg-purple-50 border border-purple-200 rounded-lg p-4 text-center">
162:         <div className="text-2xl font-bold text-purple-600">{metabolicAreas.length}</div>
163:         <div className="text-sm text-purple-700">Metabolic Areas</div>
164:     </div>
165: </div>
166:
167: { /* Metabolic Areas */}
168: <div className="space-y-6">
169:     {filteredAreas.map((area) => {
170:         const data = metabolicCore[area]
171:         return (
172:             <Card key={area} className="border-2 border-purple-100 bg-purple-50/30">
173:                 <CardHeader className="pb-4">
174:                     <div className="flex items-center justify-between">
175:                         <CardTitle className="text-lg text-purple-800 flex items-center gap-2">
176:                             <span>{getImpactIcon(data.impact)}</span>
177:                             <span>{area.replace(/[A-Z]/g, " $1").replace(/^./, (str) => str.toUpperCase())}</span>
178:                         </CardTitle>
179:                         <Badge className={` ${getImpactColor(data.impact)} border`} >{data.impact}</Badge>
180:                     </div>
181:                 </CardHeader>
182:                 <CardContent className="space-y-6">
183:                     { /* Genes Section */}
184:                     <div className="space-y-3">
185:                         <div className="flex items-center justify-between">
186:                             <Label className="text-sm font-semibold">Genes (Alleles)</Label>
187:                             <Button onClick={() => addGene(area)} size="sm" variant="outline" className="text-xs">
188:                                 <Plus className="h-3 w-3 mr-1" />
189:                                 Add Gene
190:                             </Button>
191:                         </div>
192:                     </div>
193:                 </CardContent>
194:             </Card>
195:         )
196:     })}
197: </div>

```

```

193:         {data.genes.map((gene, index) => (
194:             <div key={index} className="flex gap-2">
195:                 <Input
196:                     value={gene}
197:                     onChange={(e) => updateGene(area, index, e.target.value)}
198:                     placeholder="e.g., MTHFR (A>C)"
199:                     className="border-2 focus:border-purple-500 font-mono text-sm"
200:                 />
201:                 <Button
202:                     onClick={() => removeGene(area, index)}
203:                     size="sm"
204:                     variant="destructive"
205:                     className="px-2"
206:                 >
207:                     <Trash2 className="h-3 w-3" />
208:                 </Button>
209:             </div>
210:         )]}
211:     </div>
212: </div>
213:
214:     { /* Genotype */ }
215:     <div className="space-y-2">
216:         <Label className="text-sm font-semibold">Your Genotype</Label>
217:         <div className="flex gap-2">
218:             <Input
219:                 value={data.genotype}
220:                 onChange={(e) => updateMetabolicCore(area, { genotype: e.target.value })}
221:                 placeholder="e.g., TT, GG, GG, GG, CC"
222:                 className="border-2 focus:border-purple-500 font-mono"
223:             />
224:             <Button
225:                 onClick={() => copyToClipboard(data.genotype)}
226:                 size="sm"
227:                 variant="outline"
228:                 className="px-3"
229:             >
230:                 <Copy className="h-4 w-4" />
231:             </Button>
232:         </div>
233:     </div>
234:
235:     { /* Impact */ }
236:     <div className="space-y-2">
237:         <Label className="text-sm font-semibold">Genetic Impact</Label>
238:         <Input
239:             value={data.impact}
240:             onChange={(e) => updateMetabolicCore(area, { impact: e.target.value })}
241:             placeholder="e.g., NORMAL METHYLATION"
242:             className="border-2 focus:border-purple-500 font-semibold"
243:         />
244:     </div>
245:
246:     { /* Advice */ }
247:     <div className="space-y-2">
248:         <Label className="text-sm font-semibold">Advice & Recommendations</Label>
249:         <Textarea
250:             value={data.advice}
251:             onChange={(e) => updateMetabolicCore(area, { advice: e.target.value })}
252:             rows={3}
253:             placeholder="Detailed advice based on genetic analysis..."
254:             className="border-2 focus:border-purple-500 text-sm"
255:         />
256:     </div>
257: </CardContent>
258: </Card>
259: )
260: })}
261: </div>
262:
263: {filteredAreas.length === 0 && (
264:     <div className="text-center py-12">
265:         <div className="text-6xl mb-4">?</div>

```

```

266:         <h3 className="text-xl font-semibold text-gray-700 mb-2">No results found</h3>
267:         <p className="text-gray-500">Try adjusting your search terms</p>
268:     </div>
269: )}
270:
271: { /* Information Panel */}
272: <div className="bg-purple-50 border-2 border-purple-200 rounded-lg p-6">
273:     <h4 className="font-semibold text-purple-800 mb-3">? Metabolic Core - The STAR Feature</h4>
274:     <div className="text-sm text-purple-700 space-y-2">
275:         <p>
276:             <strong>Metabolic Core</strong> is the most comprehensive section that allows trained nutrigenomics
277:             experts to provide the best solutions for current health conditions and help prevent future r
278:         </p>
279:         <div className="grid md:grid-cols-2 gap-4 mt-4">
280:             <div>
281:                 <h5 className="font-semibold mb-2">Key Features:</h5>
282:                 <ul className="list-disc list-inside space-y-1 text-xs">
283:                     <li>Gene-by-gene analysis with specific alleles</li>
284:                     <li>Personalized genotype results</li>
285:                     <li>Direct health impact assessment</li>
286:                     <li>Actionable advice for each metabolic area</li>
287:                 </ul>
288:             </div>
289:             <div>
290:                 <h5 className="font-semibold mb-2">Expert Usage:</h5>
291:                 <ul className="list-disc list-inside space-y-1 text-xs">
292:                     <li>Functional medicine practitioners love this section</li>
293:                     <li>Enables precise intervention strategies</li>
294:                     <li>Supports both preventive and active management</li>
295:                     <li>Provides scientific basis for recommendations</li>
296:                 </ul>
297:             </div>
298:         </div>
299:     </div>
300: </div>
301: </CardContent>
302: </Card>
303: )
304: }

```

File: components\admin\nutrition-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 {
13:   Select,
14:   SelectContent,
15:   SelectItem,
16:   SelectTrigger,
17:   SelectValue,
18: } from "@/components/ui/select";
19: import { Button } from "@/components/ui/button";
20: import { Badge } from "@/components/ui/badge";
21: import {
22:   Dialog,
23:   DialogContent,
24:   DialogDescription,
25:   DialogFooter,
26:   DialogHeader,
27:   DialogTitle,
28:   DialogTrigger,
29: } from "@/components/ui/dialog";

```



```

30: import { Save, RotateCcw, Plus, Trash2, Edit3 } from "lucide-react";
31: import { useState } from "react";
32:
33: interface NutrientData {
34:   score: number;
35:   healthImpact: string;
36:   intakeLevel: string;
37:   source: string;
38: }
39:
40: interface NutritionData {
41:   vitamins: Record<string, NutrientData>;
42:   fattyAcids: Record<string, NutrientData>;
43:   elements: Record<string, NutrientData>;
44:   complexNutrients: Record<string, NutrientData>;
45: }
46:
47: interface NutritionAdminProps {
48:   nutritionData?: NutritionData;
49:   updateNutritionData?: (
50:     section: keyof NutritionData,
51:     field: string,
52:     data: Partial<NutrientData>
53:   ) => void;
54:   onSave?: () => void;
55:   onReset?: () => void;
56: }
57:
58: export default function NutritionAdmin({
59:   nutritionData: propNutritionData,
60:   updateNutritionData: propUpdateNutritionData,
61:   onSave: propOnSave,
62:   onReset: propOnReset,
63: }: NutritionAdminProps) {
64:   // Default data structure
65:   const defaultNutrientData: NutrientData = {
66:     score: 5,
67:     healthImpact: "",
68:     intakeLevel: "NORMAL INTAKE",
69:     source: "DIET",
70:   };
71:
72:   const initialNutritionData: NutritionData = {
73:     vitamins: {
74:       vitaminA: {
75:         score: 6,
76:         healthImpact: "Vision & Immune",
77:         intakeLevel: "ENHANCED INTAKE",
78:         source: "DIET",
79:       },
80:       vitaminB12: {
81:         score: 8,
82:         healthImpact: "Energy & Nerves",
83:         intakeLevel: "ENHANCED INTAKE (METHYLCOBALAMIN)",
84:         source: "SUPPLEMENTS",
85:       },
86:       vitaminC: {
87:         score: 4,
88:         healthImpact: "Immune & Collagen",
89:         intakeLevel: "NORMAL INTAKE",
90:         source: "DIET",
91:       },
92:       vitaminD: {
93:         score: 9,
94:         healthImpact: "Bones & Immune",
95:         intakeLevel: "ENHANCED INTAKE & SUN EXPOSURE",
96:         source: "DIET & SUN EXPOSURE",
97:       },
98:       folate: {
99:         score: 7,
100:         healthImpact: "DNA & Blood",
101:         intakeLevel: "ENHANCED INTAKE (L METHYLFOLATE)",
102:         source: "DIET & SUPPLEMENTS",

```

```

103:     },
104:   },
105:   fattyAcids: {
106:     omega3: {
107:       score: 8,
108:       healthImpact: "Heart & Brain",
109:       intakeLevel: "ENHANCED INTAKE",
110:       source: "ENHANCED INTAKE (VEG/FISH OIL) & DIET",
111:     },
112:     omega6: {
113:       score: 3,
114:       healthImpact: "Inflammation",
115:       intakeLevel: "RESTRICTED INTAKE",
116:       source: "DIET",
117:     },
118:   },
119:   elements: {
120:     iron: {
121:       score: 6,
122:       healthImpact: "Blood & Energy",
123:       intakeLevel: "ENHANCED INTAKE (VEG SOURCES)",
124:       source: "DIET",
125:     },
126:     zinc: {
127:       score: 5,
128:       healthImpact: "Immune & Healing",
129:       intakeLevel: "NORMAL INTAKE",
130:       source: "DIET",
131:     },
132:     magnesium: {
133:       score: 7,
134:       healthImpact: "Muscle & Nerve",
135:       intakeLevel: "ENHANCED INTAKE",
136:       source: "DIET & SUPPLEMENTS",
137:     },
138:   },
139:   complexNutrients: {
140:     antioxidants: {
141:       score: 5,
142:       healthImpact: "Cell Protection",
143:       intakeLevel: "ENHANCED INTAKE",
144:       source: "DIET",
145:     },
146:     polyphenols: {
147:       score: 4,
148:       healthImpact: "Anti-inflammatory",
149:       intakeLevel: "ENHANCED INTAKE",
150:       source: "DIET",
151:     },
152:   },
153: };
154:
155: // State management
156: const [nutritionData, setNutritionData] = useState<NutritionData>(
157:   propNutritionData || initialNutritionData
158: );
159: const [activeSection, setActiveSection] =
160:   useState<keyof NutritionData>("vitamins");
161: const [isAddDialogOpen, setIsAddDialogOpen] = useState(false);
162: const [newNutrientName, setNewNutrientName] = useState("");
163: const [isEditMode, setIsEditMode] = useState<string | null>(null);
164: const [editingName, setEditingName] = useState("");
165:
166: // CRUD Operations
167: const updateNutritionData = (
168:   section: keyof NutritionData,
169:   field: string,
170:   data: Partial<NutrientData>
171: ) => {
172:   setNutritionData((prev) => ({
173:     ...prev,
174:     [section]: {
175:       ...prev[section],

```

```

176:         [field]: {
177:             ...prev[section][field],
178:             ...data,
179:         },
180:     },
181: });
182:
183: // Call prop function if provided
184: if (propUpdateNutritionData) {
185:     propUpdateNutritionData(section, field, data);
186: }
187: };
188:
189: const addNutrient = (section: keyof NutritionData, name: string) => {
190:     if (
191:         !name.trim() ||
192:         nutritionData[section][name.toLowerCase().replace(/\s+/g, "")]
193:     ) {
194:         return false;
195:     }
196:
197:     const fieldName = name.toLowerCase().replace(/\s+/g, "");
198:     setNutritionData((prev) => ({
199:         ...prev,
200:         [section]: {
201:             ...prev[section],
202:             [fieldName]: { ...defaultNutrientData },
203:         },
204:     }));
205:     return true;
206: };
207:
208: const deleteNutrient = (section: keyof NutritionData, field: string) => {
209:     setNutritionData((prev) => {
210:         const newSection = { ...prev[section] };
211:         delete newSection[field];
212:         return {
213:             ...prev,
214:             [section]: newSection,
215:         };
216:     });
217: };
218:
219: const renameNutrient = (
220:     section: keyof NutritionData,
221:     oldField: string,
222:     newName: string
223: ) => {
224:     if (
225:         !newName.trim() ||
226:         newName.toLowerCase().replace(/\s+/g, "") === oldField
227:     ) {
228:         return false;
229:     }
230:
231:     const newFieldName = newName.toLowerCase().replace(/\s+/g, "");
232:     if (nutritionData[section][newFieldName] && newFieldName !== oldField) {
233:         return false; // Name already exists
234:     }
235:
236:     setNutritionData((prev) => {
237:         const newSection = { ...prev[section] };
238:         const data = newSection[oldField];
239:         delete newSection[oldField];
240:         newSection[newFieldName] = data;
241:
242:         return {
243:             ...prev,
244:             [section]: newSection,
245:         };
246:     });
247:     return true;
248: };

```

```

249:
250:   const onSave = () => {
251:     console.log("Saving nutrition data:", nutritionData);
252:     if (propOnSave) {
253:       propOnSave();
254:     } else {
255:       alert("Nutrition data saved successfully!");
256:     }
257:   };
258:
259:   const onReset = () => {
260:     setNutritionData(initialNutritionData);
261:     if (propOnReset) {
262:       propOnReset();
263:     }
264:   };
265:
266:   const handleAddNutrient = () => {
267:     if (addNutrient(activeSection, newNutrientName)) {
268:       setNewNutrientName("");
269:       setIsAddDialogOpen(false);
270:     } else {
271:       alert("Invalid name or nutrient already exists!");
272:     }
273:   };
274:
275:   const handleRename = (field: string) => {
276:     if (renameNutrient(activeSection, field, editingName)) {
277:       setIsEditMode(null);
278:       setEditingName("");
279:     } else {
280:       alert("Invalid name or nutrient already exists!");
281:     }
282:   };
283:
284:   const renderNutrientField = (
285:     section: keyof NutritionData,
286:     field: string,
287:     data: NutrientData,
288:     title: string
289:   ) => (
290:     <Card key={field} className="border-2 border-green-100 bg-green-50/30">
291:       <CardHeader className="pb-3">
292:         <CardTitle className="text-lg text-green-800 flex items-center justify-between">
293:           {isEditMode === field ? (
294:             <div className="flex items-center gap-2 flex-1">
295:               <Input
296:                 value={editingName}
297:                 onChange={(e) => setEditingName(e.target.value)}
298:                 className="text-lg font-semibold border-2 focus:border-green-500"
299:                 onPress={(e) => e.key === "Enter" && handleRename(field)}
300:               />
301:               <Button size="sm" onClick={() => handleRename(field)}>
302:                 ?
303:               </Button>
304:               <Button
305:                 size="sm"
306:                 variant="outline"
307:                 onClick={() => {
308:                   setIsEditMode(null);
309:                   setEditingName("");
310:                 }}
311:               >
312:                 ?
313:               </Button>
314:             </div>
315:           ) : (
316:             <div className="flex items-center gap-2 flex-1">
317:               <span>{title}</span>
318:               <Button
319:                 size="sm"
320:                 variant="ghost"
321:                 onClick={() => {

```

```

322:         setIsEditMode(field);
323:         setEditingName(title);
324:     }}
325:     >
326:     <Edit3 className="h-3 w-3" />
327: </Button>
328: </div>
329: )}
330:
331: <div className="flex items-center gap-2">
332:     <Badge
333:         variant="outline"
334:         className={`{
335:             data.score >= 7
336:               ? "border-red-500 text-red-700"
337:             : data.score >= 4
338:               ? "border-yellow-500 text-yellow-700"
339:             : "border-green-500 text-green-700"
340:           }`}
341:     >
342:       Score: {data.score}/10
343:     </Badge>
344:     <Button
345:       size="sm"
346:       variant="ghost"
347:       onClick={() => deleteNutrient(section, field)}
348:       className="text-red-600 hover:bg-red-100"
349:     >
350:       <Trash2 className="h-4 w-4" />
351:     </Button>
352: </div>
353: </CardTitle>
354: </CardHeader>
355: <CardContent className="space-y-4">
356:   <div className="grid grid-cols-2 gap-4">
357:     <div className="space-y-2">
358:       <Label className="text-sm font-medium">Score (1-10)</Label>
359:       <Input
360:         type="number"
361:         min="1"
362:         max="10"
363:         value={data.score}
364:         onChange={(e) => {
365:           const score = Math.max(
366:             1,
367:             Math.min(10, Number.parseInt(e.target.value) || 1)
368:           );
369:           updateNutritionData(section, field, { score });
370:         }}
371:         className="border-2 focus:border-green-500"
372:       />
373:     </div>
374:     <div className="space-y-2">
375:       <Label className="text-sm font-medium">Health Impact</Label>
376:       <Input
377:         value={data.healthImpact}
378:         onChange={(e) =>
379:           updateNutritionData(section, field, {
380:             healthImpact: e.target.value,
381:           })
382:         }
383:         placeholder="e.g., Skin & Vision"
384:         className="border-2 focus:border-green-500"
385:       />
386:     </div>
387:   </div>
388:
389:   <div className="space-y-2">
390:     <Label className="text-sm font-medium">Intake Level</Label>
391:     <Select
392:       value={data.intakeLevel}
393:       onChange={(value) =>
394:         updateNutritionData(section, field, { intakeLevel: value })

```

```

395:     }
396:   >
397:     <SelectTrigger className="border-2 focus:border-green-500">
398:       <SelectValue />
399:     </SelectTrigger>
400:     <SelectContent>
401:       <SelectItem value="ENHANCED INTAKE">ENHANCED INTAKE</SelectItem>
402:       <SelectItem value="NORMAL INTAKE">NORMAL INTAKE</SelectItem>
403:       <SelectItem value="RESTRICTED INTAKE">
404:         RESTRICTED INTAKE
405:       </SelectItem>
406:       <SelectItem value="ENHANCED INTAKE (METHYLCOBALAMIN)">
407:         ENHANCED INTAKE (METHYLCOBALAMIN)
408:       </SelectItem>
409:       <SelectItem value="ENHANCED INTAKE (L METHYLFOLATE)">
410:         ENHANCED INTAKE (L METHYLFOLATE)
411:       </SelectItem>
412:       <SelectItem value="ENHANCED INTAKE & SUN EXPOSURE">
413:         ENHANCED INTAKE & SUN EXPOSURE
414:       </SelectItem>
415:       <SelectItem value="ENHANCED INTAKE (VEG SOURCES)">
416:         ENHANCED INTAKE (VEG SOURCES)
417:       </SelectItem>
418:     </SelectContent>
419:   </Select>
420: </div>
421:
422: <div className="space-y-2">
423:   <Label className="text-sm font-medium">Source</Label>
424:   <Select
425:     value={data.source}
426:     onChange={(value) =>
427:       updateNutritionData(section, field, { source: value })
428:     }
429:   >
430:     <SelectTrigger className="border-2 focus:border-green-500">
431:       <SelectValue />
432:     </SelectTrigger>
433:     <SelectContent>
434:       <SelectItem value="DIET">DIET</SelectItem>
435:       <SelectItem value="SUPPLEMENTS">SUPPLEMENTS</SelectItem>
436:       <SelectItem value="DIET & SUPPLEMENTS">
437:         DIET & SUPPLEMENTS
438:       </SelectItem>
439:       <SelectItem value="DIET & SUN EXPOSURE">
440:         DIET & SUN EXPOSURE
441:       </SelectItem>
442:       <SelectItem value="ENHANCED INTAKE (VEG/FISH OIL) & DIET">
443:         ENHANCED INTAKE (VEG/FISH OIL) & DIET
444:       </SelectItem>
445:     </SelectContent>
446:   </Select>
447: </div>
448: </CardContent>
449: </Card>
450: );
451:
452: const getSectionTitle = (section: keyof NutritionData) => {
453:   switch (section) {
454:     case "vitamins":
455:       return "? Vitamins";
456:     case "fattyAcids":
457:       return "? Fatty Acids";
458:     case "elements":
459:       return "? Essential Elements";
460:     case "complexNutrients":
461:       return "? Complex Nutrients";
462:     default:
463:       return section;
464:   }
465: };
466:
467: const getSectionIcon = (section: keyof NutritionData) => {

```

```

468:     switch (section) {
469:         case "vitamins":
470:             return "?";
471:         case "fattyAcids":
472:             return "?";
473:         case "elements":
474:             return "?";
475:         case "complexNutrients":
476:             return "?";
477:         default:
478:             return "?";
479:     }
480: };
481:
482: return (
483:     <Card className="shadow-lg border-0 bg-white">
484:         <CardHeader className="bg-gradient-to-r from-green-500 to-emerald-600 text-white rounded-t-lg">
485:             <CardTitle className="text-2xl">? Nutrition Analysis</CardTitle>
486:             <CardDescription className="text-green-100">
487:                 Configure vitamins, minerals, fatty acids, and complex nutrients with
488:                 full CRUD operations
489:             </CardDescription>
490:         </CardHeader>
491:         <CardContent className="space-y-8 p-8">
492:             { /* Action Buttons */ }
493:             <div className="flex justify-between items-center flex-wrap gap-4">
494:                 <div className="flex gap-2 flex-wrap">
495:                     {(Object.keys(nutritionData) as Array<keyof NutritionData>).map(
496:                         (section) => (
497:                             <Button
498:                                 key={section}
499:                                 variant={activeSection === section ? "default" : "outline"}
500:                                 size="sm"
501:                                 onClick={() => setActiveSection(section)}
502:                                 className="flex items-center gap-2"
503:                             >
504:                                 <span>{getSectionIcon(section)}</span>
505:                                 <span className="capitalize">
506:                                     {section.replace(/([A-Z])/g, " $1")}
507:                                 </span>
508:                             </Button>
509:                         )
510:                     )}
511:                 </div>
512:
513:                 <div className="flex gap-2 flex-wrap">
514:                     <Dialog open={isAddDialogOpen} onOpenChange={setIsAddDialogOpen}>
515:                         <DialogTrigger asChild>
516:                             <Button
517:                                 variant="outline"
518:                                 size="sm"
519:                                 className="bg-green-50 border-green-200 text-green-700 hover:bg-green-100"
520:                             >
521:                                 <Plus className="h-4 w-4 mr-2" />
522:                                 Add New
523:                             </Button>
524:                         </DialogTrigger>
525:                         <DialogContent>
526:                             <DialogHeader>
527:                                 <DialogTitle>
528:                                     Add New{" "}
529:                                     {getSectionTitle(activeSection).replace(/^[^w\s]/gi, "")}{ " "}
530:                                     Item
531:                                 </DialogTitle>
532:                                 <DialogDescription>
533:                                     Enter the name for the new nutrient in the {activeSection}{ " "}
534:                                     section.
535:                                 </DialogDescription>
536:                             </DialogHeader>
537:                             <div className="space-y-4 py-4">
538:                                 <div className="space-y-2">
539:                                     <Label htmlFor="nutrientName">Nutrient Name</Label>
540:                                     <Input

```

```

541:         id="nutrientName"
542:         value={newNutrientName}
543:         onChange={(e) => setNewNutrientName(e.target.value)}
544:         placeholder="e.g., Vitamin E, Selenium, etc."
545:         onKeyPress={(e) =>
546:             e.key === "Enter" && handleAddNutrient()
547:         }
548:     />
549: </div>
550: </div>
551: <DialogFooter>
552:     <Button
553:         variant="outline"
554:         onClick={() => {
555:             setIsAddDialogOpen(false);
556:             setNewNutrientName("");
557:         }}
558:     >
559:         Cancel
560:     </Button>
561:     <Button
562:         onClick={handleAddNutrient}
563:         disabled={!newNutrientName.trim()}
564:     >
565:         Add Nutrient
566:     </Button>
567: </DialogFooter>
568: </DialogContent>
569: </Dialog>
570:
571: <Button onClick={onReset} variant="outline" size="sm">
572:     <RotateCcw className="h-4 w-4 mr-2" />
573:     Reset
574: </Button>
575: <Button onClick={onSave} size="sm">
576:     <Save className="h-4 w-4 mr-2" />
577:     Save Changes
578: </Button>
579: </div>
580: </div>
581:
582: {/* Section Content */}
583: <div className="space-y-6">
584:     <h3 className="text-xl font-semibold text-gray-800 border-b-2 border-green-200 pb-2">
585:         {getSectionTitle(activeSection)}
586:     </h3>
587:
588:     <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-6">
589:         {Object.entries(nutritionData[activeSection]).map(([key, item]) =>
590:             renderNutrientField(
591:                 activeSection,
592:                 key,
593:                 item as NutrientData,
594:                 key
595:                     .replace(/[A-Z]/g, " $1")
596:                     .replace(/^.\/, (str) => str.toUpperCase())
597:             )
598:         )}
599:     </div>
600:
601:     {Object.keys(nutritionData[activeSection]).length === 0 && (
602:         <div className="text-center py-12 bg-gray-50 rounded-lg border-2 border-dashed border-gray-300">
603:             <div className="text-gray-500 text-lg mb-2">
604:                 No items in this section
605:             </div>
606:             <div className="text-gray-400 text-sm">
607:                 Click "Add New" to add your first nutrient
608:             </div>
609:         </div>
610:     )}
611: </div>
612:
613: {/* Section Guidelines */}

```



```

614: <div className="bg-green-50 border-2 border-green-200 rounded-lg p-6">
615:   <h4 className="font-semibold text-green-800 mb-2">
616:     ? {getSectionTitle(activeSection)} Guidelines:
617:   </h4>
618:   <div className="text-sm text-green-700 space-y-2">
619:     {activeSection === "vitamins" && (
620:       <ul className="list-disc list-inside space-y-1">
621:         <li>
622:           Score 1-3: Low requirement, normal dietary intake sufficient
623:         </li>
624:         <li>
625:           Score 4-6: Moderate requirement, enhanced dietary intake
626:           recommended
627:         </li>
628:         <li>
629:           Score 7-10: High requirement, supplementation may be needed
630:         </li>
631:         <li>Consider bioavailable forms for better absorption</li>
632:       </ul>
633:     )}
634:     {activeSection === "elements" && (
635:       <ul className="list-disc list-inside space-y-1">
636:         <li>
637:           Essential minerals required for various metabolic processes
638:         </li>
639:         <li>
640:           Balance is key - both deficiency and excess can be harmful
641:         </li>
642:         <li>Consider interactions between different minerals</li>
643:         <li>Monitor through regular blood tests</li>
644:       </ul>
645:     )}
646:     {activeSection === "fattyAcids" && (
647:       <ul className="list-disc list-inside space-y-1">
648:         <li>
649:           Omega-3 fatty acids are essential for heart and brain health
650:         </li>
651:         <li>Consider EPA:DHA ratio for optimal benefits</li>
652:         <li>
653:           Vegetarian sources include flaxseed, chia seeds, walnuts
654:         </li>
655:         <li>Fish oil supplements for non-vegetarians</li>
656:       </ul>
657:     )}
658:     {activeSection === "complexNutrients" && (
659:       <ul className="list-disc list-inside space-y-1">
660:         <li>Antioxidants help protect against cellular damage</li>
661:         <li>Anti-inflammatory compounds reduce chronic inflammation</li>
662:         <li>Best obtained from colorful fruits and vegetables</li>
663:         <li>Synergistic effects when consumed together</li>
664:       </ul>
665:     )}
666:   </div>
667: </div>
668:
669: { /* Summary Statistics */ }
670: <div className="grid md:grid-cols-4 gap-4">
671:   <div className="bg-red-50 border border-red-200 rounded-lg p-4 text-center">
672:     <div className="text-2xl font-bold text-red-600">
673:       {
674:         Object.values(nutritionData[activeSection]).filter(
675:           (item: any) => item.score >= 7
676:         ).length
677:       }
678:     </div>
679:     <div className="text-sm text-red-700">High Priority</div>
680:   </div>
681:   <div className="bg-yellow-50 border border-yellow-200 rounded-lg p-4 text-center">
682:     <div className="text-2xl font-bold text-yellow-600">
683:       {
684:         Object.values(nutritionData[activeSection]).filter(
685:           (item: any) => item.score >= 4 && item.score < 7
686:         ).length

```

```

687:         }
688:     </div>
689:     <div className="text-sm text-yellow-700">Moderate Priority</div>
690: </div>
691: <div className="bg-green-50 border border-green-200 rounded-lg p-4 text-center">
692:     <div className="text-2xl font-bold text-green-600">
693:         {
694:             Object.values(nutritionData[activeSection]).filter(
695:                 (item: any) => item.score < 4
696:             ).length
697:         }
698:     </div>
699:     <div className="text-sm text-green-700">Low Priority</div>
700: </div>
701: <div className="bg-blue-50 border border-blue-200 rounded-lg p-4 text-center">
702:     <div className="text-2xl font-bold text-blue-600">
703:         {Object.keys(nutritionData[activeSection]).length}
704:     </div>
705:     <div className="text-sm text-blue-700">Total Items</div>
706: </div>
707: </div>
708: </CardContent>
709: </Card>
710: );
711: }

```

■ 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, useEffect } 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 = (
42:     event: React.ChangeEvent<HTMLInputElement>,
43:     signatureField: "signature1" | "signature2"

```

```

44:   ) => {
45:     const file = event.target.files?.[0];
46:     if (file) {
47:       // Validate file type
48:       if (!file.type.startsWith("image/")) {
49:         alert("Please select an image file");
50:         return;
51:       }
52:
53:       // Validate file size (5MB limit)
54:       if (file.size > 5 * 1024 * 1024) {
55:         alert("File size should be less than 5MB");
56:         return;
57:       }
58:
59:       const reader = new FileReader();
60:       reader.onload = (e) => {
61:         const base64String = e.target?.result as string;
62:         updatePatientInfo(signatureField, base64String);
63:       };
64:       reader.readAsDataURL(file);
65:     }
66:   };
67:
68:   const removeSignature = (
69:     signatureField: "signature1" | "signature2",
70:     fileInputRef: React.RefObject<HTMLInputElement>
71:   ) => {
72:     updatePatientInfo(signatureField, "");
73:     if (fileInputRef.current) {
74:       fileInputRef.current.value = "";
75:     }
76:   };
77:
78:   return (
79:     <Card className="shadow-lg border-0 bg-white">
80:       <CardHeader className="bg-gradient-to-r from-blue-500 to-purple-600 text-white rounded-t-lg">
81:         <CardTitle className="text-2xl">? Patient Information</CardTitle>
82:         <CardDescription className="text-blue-100">
83:           Complete patient details as they appear on the report cover page
84:         </CardDescription>
85:       </CardHeader>
86:       <CardContent className="space-y-6 p-8">
87:         { /* Action Buttons */ }
88:         <div className="flex justify-end gap-2 mb-6">
89:           <Button onClick={onReset} variant="outline" size="sm">
90:             <RotateCcw className="h-4 w-4 mr-2" />
91:             Reset
92:           </Button>
93:           <Button onClick={onSave} size="sm">
94:             <Save className="h-4 w-4 mr-2" />
95:             Save Changes
96:           </Button>
97:         </div>
98:
99:         { /* Basic Information */ }
100:         <div className="space-y-4">
101:           <h3 className="text-lg font-semibold text-gray-800 border-b pb-2">
102:             Basic Information
103:           </h3>
104:           <div className="grid md:grid-cols-2 gap-6">
105:             <div className="space-y-2">
106:               <Label
107:                 htmlFor="name"
108:                 className="text-sm font-semibold text-gray-700 required"
109:               >
110:                 Patient Name *
111:               </Label>
112:               <Input
113:                 id="name"
114:                 value={patientInfo.name}
115:                 onChange={(e) => updatePatientInfo("name", e.target.value)}
116:                 placeholder="Enter full patient name"

```

```

117:         className="border-2 focus:border-blue-500"
118:         required
119:     />
120: </div>
121: <div className="space-y-2">
122:     <Label
123:         htmlFor="gender"
124:         className="text-sm font-semibold text-gray-700 required"
125:     >
126:         Gender *
127:     </Label>
128:     <Select
129:         value={patientInfo.gender}
130:         onChange={(value) => updatePatientInfo("gender", value)}
131:     >
132:         <SelectTrigger className="border-2 focus:border-blue-500">
133:             <SelectValue placeholder="Select gender" />
134:         </SelectTrigger>
135:         <SelectContent>
136:             <SelectItem value="MALE">Male</SelectItem>
137:             <SelectItem value="FEMALE">Female</SelectItem>
138:             <SelectItem value="OTHER">Other</SelectItem>
139:         </SelectContent>
140:     </Select>
141: </div>
142: <div className="space-y-2">
143:     <Label
144:         htmlFor="birthDate"
145:         className="text-sm font-semibold text-gray-700 required"
146:     >
147:         Birth Date *
148:     </Label>
149:     <Input
150:         id="birthDate"
151:         type="date"
152:         value={patientInfo.birthDate}
153:         onChange={(e) => updatePatientInfo("birthDate", e.target.value)}
154:         className="border-2 focus:border-blue-500"
155:         required
156:     />
157: </div>
158: </div>
159: </div>
160:
161: { /* Sample Information */ }
162: <div className="space-y-4">
163:     <h3 className="text-lg font-semibold text-gray-800 border-b pb-2">
164:         Sample Information
165:     </h3>
166:     <div className="grid md:grid-cols-2 gap-6">
167:         <div className="space-y-2">
168:             <Label
169:                 htmlFor="sampleCode"
170:                 className="text-sm font-semibold text-gray-700 required"
171:             >
172:                 Sample Code *
173:             </Label>
174:             <Input
175:                 id="sampleCode"
176:                 value={patientInfo.sampleCode}
177:                 onChange={(e) =>
178:                     updatePatientInfo("sampleCode", e.target.value)
179:                 }
180:                 placeholder="e.g., DNL1000000110"
181:                 className="border-2 focus:border-blue-500 font-mono"
182:                 required
183:             />
184:             <p className="text-xs text-gray-500">
185:                 Unique identifier for the DNA sample
186:             </p>
187:         </div>
188:         <div className="space-y-2">
189:             <Label

```

```

190:         htmlFor="sampleDate"
191:         className="text-sm font-semibold text-gray-700 required"
192:     >
193:         Sample Collection Date *
194:     </Label>
195:     <Input
196:         id="sampleDate"
197:         type="date"
198:         value={patientInfo.sampleDate}
199:         onChange={(e) =>
200:             updatePatientInfo("sampleDate", e.target.value)
201:         }
202:         className="border-2 focus:border-blue-500"
203:         required
204:     />
205: </div>
206: <div className="space-y-2">
207:     <Label
208:         htmlFor="reportDate"
209:         className="text-sm font-semibold text-gray-700 required"
210:     >
211:         Report Date *
212:     </Label>
213:     <Input
214:         id="reportDate"
215:         type="date"
216:         value={patientInfo.reportDate}
217:         onChange={(e) =>
218:             updatePatientInfo("reportDate", e.target.value)
219:         }
220:         className="border-2 focus:border-blue-500"
221:         required
222:     />
223: </div>
224: </div>
225: </div>
226:
227: { /* Authorization & Verification */ }
228: <div className="space-y-4">
229:     <h3 className="text-lg font-semibold text-gray-800 border-b pb-2">
230:         Authorization & Verification
231:     </h3>
232:     <div className="grid md:grid-cols-2 gap-6">
233:         <div className="space-y-2">
234:             <Label
235:                 htmlFor="reportAuth"
236:                 className="text-sm font-semibold text-gray-700"
237:             >
238:                 Report Authentication & Analytics
239:             </Label>
240:             <Input
241:                 id="reportAuth"
242:                 value={patientInfo.reportAuth}
243:                 onChange={(e) =>
244:                     updatePatientInfo("reportAuth", e.target.value)
245:                 }
246:                 placeholder="Enter report authentication details"
247:                 className="border-2 focus:border-blue-500"
248:             />
249:             <p className="text-xs text-gray-500">
250:                 Person/organization responsible for report authentication
251:             </p>
252:         </div>
253:         <div className="space-y-2">
254:             <Label
255:                 htmlFor="genomicAnalytics"
256:                 className="text-sm font-semibold text-gray-700"
257:             >
258:                 Genomic Data Analytics
259:             </Label>
260:             <Input
261:                 id="genomicAnalytics"
262:                 value={patientInfo.genomicAnalytics}

```

```

263:         onChange={e =>
264:             updatePatientInfo("genomicAnalytics", e.target.value)
265:         }
266:         placeholder="Enter genomic data analytics info"
267:         className="border-2 focus:border-blue-500"
268:     />
269:     <p className="text-xs text-gray-500">
270:         Laboratory or system used for genomic analysis
271:     </p>
272: </div>
273: </div>
274:
275: { /* Signature Upload Section */}
276: <div className="space-y-4 mt-6 pt-6 border-t border-gray-200">
277:     <h4 className="text-md font-semibold text-gray-800">
278:         Digital Signatures
279:     </h4>
280:     <div className="grid md:grid-cols-2 gap-6">
281:         { /* Signature 1 */}
282:         <div className="space-y-3">
283:             <Label className="text-sm font-semibold text-gray-700">
284:                 Primary Signature
285:             </Label>
286:             <div className="border-2 border-dashed border-gray-300 rounded-lg p-4 bg-gray-50">
287:                 {patientInfo.signature1 ? (
288:                     <div className="relative">
289:                         <img
290:                             src={patientInfo.signature1}
291:                             alt="Primary Signature"
292:                             className="max-w-full max-h-32 object-contain mx-auto"
293:                         />
294:                         <Button
295:                             onClick={() =>
296:                                 removeSignature("signature1", fileInput1Ref)
297:                             }
298:                             variant="outline"
299:                             size="sm"
300:                             className="absolute top-0 right-0 h-6 w-6 p-0"
301:                         >
302:                             <X className="h-3 w-3" />
303:                         </Button>
304:                     </div>
305:                 ) : (
306:                     <div className="text-center">
307:                         <Upload className="h-8 w-8 text-gray-400 mx-auto mb-2" />
308:                         <p className="text-sm text-gray-500 mb-2">
309:                             Upload primary signature
310:                         </p>
311:                         <Button
312:                             onClick={() => fileInput1Ref.current?.click()}
313:                             variant="outline"
314:                             size="sm"
315:                         >
316:                             <Upload className="h-4 w-4 mr-2" />
317:                             Choose File
318:                         </Button>
319:                     </div>
320:                 )}
321:             <input
322:                 ref={fileInput1Ref}
323:                 type="file"
324:                 accept="image/*"
325:                 onChange={e => handleImageUpload(e, "signature1")}
326:                 className="hidden"
327:             />
328:         </div>
329:         <p className="text-xs text-gray-500">
330:             Upload signature image (PNG, JPG, JPEG - Max 5MB)
331:         </p>
332:     </div>
333:
334: { /* Signature 2 */}
335: <div className="space-y-3">

```

```

336:         <Label className="text-sm font-semibold text-gray-700">
337:             Secondary Signature
338:         </Label>
339:     <div className="border-2 border-dashed border-gray-300 rounded-lg p-4 bg-gray-50">
340:         {patientInfo.signature2 ? (
341:             <div className="relative">
342:                 <img
343:                     src={patientInfo.signature2}
344:                     alt="Secondary Signature"
345:                     className="max-w-full max-h-32 object-contain mx-auto"
346:                 />
347:                 <Button
348:                     onClick={() =>
349:                         removeSignature("signature2", fileInput2Ref)
350:                     }
351:                     variant="outline"
352:                     size="sm"
353:                     className="absolute top-0 right-0 h-6 w-6 p-0"
354:                 >
355:                     <X className="h-3 w-3" />
356:                 </Button>
357:             </div>
358:         ) : (
359:             <div className="text-center">
360:                 <Upload className="h-8 w-8 text-gray-400 mx-auto mb-2" />
361:                 <p className="text-sm text-gray-500 mb-2">
362:                     Upload secondary signature
363:                 </p>
364:                 <Button
365:                     onClick={() => fileInput2Ref.current?.click()}
366:                     variant="outline"
367:                     size="sm"
368:                 >
369:                     <Upload className="h-4 w-4 mr-2" />
370:                     Choose File
371:                 </Button>
372:             </div>
373:         )}
374:         <input
375:             ref={fileInput2Ref}
376:             type="file"
377:             accept="image/*"
378:             onChange={(e) => handleImageUpload(e, "signature2")}
379:             className="hidden"
380:         />
381:     </div>
382:     <p className="text-xs text-gray-500">
383:         Upload signature image (PNG, JPG, JPEG - Max 5MB)
384:     </p>
385: </div>
386: </div>
387: </div>
388:
389: {/* Rest of the verification fields */}
390: <div className="grid md:grid-cols-2 gap-6">
391:     <div className="space-y-2">
392:         <Label
393:             htmlFor="checkedBy"
394:             className="text-sm font-semibold text-gray-700"
395:         >
396:             Checked & Verified By
397:         </Label>
398:         <Input
399:             id="checkedBy"
400:             value={patientInfo.checkedBy}
401:             onChange={(e) => updatePatientInfo("checkedBy", e.target.value)}
402:             placeholder="Enter verifier name"
403:             className="border-2 focus:border-blue-500"
404:         />
405:         <p className="text-xs text-gray-500">
406:             Medical professional who verified the report
407:         </p>
408:     </div>

```

```

409:         <div className="space-y-2">
410:             <Label
411:                 htmlFor="scientificContent"
412:                 className="text-sm font-semibold text-gray-700"
413:             >
414:                 Scientific Content Verified By
415:             </Label>
416:             <Input
417:                 id="scientificContent"
418:                 value={patientInfo.scientificContent}
419:                 onChange={(e) =>
420:                     updatePatientInfo("scientificContent", e.target.value)
421:                 }
422:                 placeholder="Enter scientific content verifier"
423:                 className="border-2 focus:border-blue-500"
424:             />
425:             <p className="text-xs text-gray-500">
426:                 Scientific expert who verified the content accuracy
427:             </p>
428:         </div>
429:     </div>
430: </div>
431:
432:     <div className="space-y-3">
433:         <Label
434:             htmlFor="disclaimer"
435:             className="text-lg font-semibold text-gray-800"
436:         >
437:             Disclaimer
438:         </Label>
439:         <Textarea
440:             id="disclaimer"
441:             value={patientInfo.disclaimer}
442:             onChange={(e) => updatePatientInfo("disclaimer", e.target.value)}
443:             placeholder="Enter disclaimer text"
444:             rows={10}
445:             className="border-2 focus:border-green-500 text-sm leading-relaxed"
446:         />
447:     </div>
448:
449:     <style jsx>{`
450:         .required::after {
451:             content: " *";
452:             color: #ef4444;
453:         }
454:     `}</style>
455: </CardContent>
456: </Card>
457: );
458: }

```

File: components\admin\pdf-generator.tsx

```

=====
1: "use client"
2:
3: import { Button } from "@/components/ui/button"
4: import { Card, CardContent, CardDescription, CardHeader, CardTitle } from "@/components/ui/card"
5: import { Download, Printer, Eye } from "lucide-react"
6: import { useState } from "react"
7: import type { ComprehensiveReportData } from "@/types/report-types"
8: import { useToast } from "@/components/ui/use-toast" // Import useToast
9:
10: interface PDFGeneratorProps {
11:     reportData: ComprehensiveReportData
12: }
13:
14: export default function PDFGenerator({ reportData }: PDFGeneratorProps) {
15:     const [isGenerating, setIsGenerating] = useState(false)
16:     const { toast } = useToast() // Initialize useToast
17:
18:     // State for PDF settings (can be expanded if more options are needed for actual PDF generation)

```



```

19: const [pdfSettings, setPdfSettings] = useState({
20:   format: "A4",
21:   orientation: "portrait",
22:   margins: "normal",
23:   quality: "high",
24:   includeGraphics: true,
25:   includeColors: true,
26: })
27:
28: const handlePrintOrDownload = async (action: "print" | "download") => {
29:   setIsGenerating(true)
30:   try {
31:     // Open a new window to the PDF generation page
32:     const printWindow = window.open("/report/pdf", "_blank")
33:
34:     if (printWindow) {
35:       // Wait for the content to load in the new window
36:       printWindow.onload = () => {
37:         // Give a small delay to ensure all content and styles are rendered
38:         setTimeout(() => {
39:           if (action === "print") {
40:             printWindow.print()
41:             toast({
42:               title: "Print Dialog Opened",
43:               description: "Please select 'Save as PDF' or your printer in the dialog.",
44:             })
45:           } else {
46:             // For download, the browser's print dialog usually offers "Save as PDF"
47:             // We trigger print, and the user saves it.
48:             printWindow.print()
49:             toast({
50:               title: "Download Initiated",
51:               description: "Please select 'Save as PDF' in the print dialog to download.",
52:             })
53:           }
54:           setIsGenerating(false)
55:         }, 1500) // Increased delay for better rendering
56:       }
57:       printWindow.onerror = () => {
58:         toast({
59:           title: "Error",
60:           description: "Could not open print window. Please check pop-up blockers.",
61:           variant: "destructive",
62:         })
63:         setIsGenerating(false)
64:       }
65:     } else {
66:       toast({
67:         title: "Error",
68:         description: "Pop-up blocked or window could not be opened. Please allow pop-ups for this site.",
69:         variant: "destructive",
70:       })
71:       setIsGenerating(false)
72:     }
73:   } catch (error) {
74:     console.error("Error handling PDF:", error)
75:     toast({
76:       title: "Error",
77:       description: "An unexpected error occurred during PDF generation.",
78:       variant: "destructive",
79:     })
80:     setIsGenerating(false)
81:   }
82: }
83:
84: const previewPDF = () => {
85:   window.open("/report/preview", "_blank")
86:   toast({
87:     title: "Preview Opened",
88:     description: "Your report preview is open in a new tab.",
89:   })
90: }
91:

```

```

92:   return (
93:     <Card className="shadow-lg border-0 bg-white">
94:       <CardHeader className="bg-gradient-to-r from-red-500 to-pink-600 text-white rounded-t-lg">
95:         <CardTitle className="text-2xl">? PDF Generation</CardTitle>
96:         <CardDescription className="text-red-100">Generate and download professional PDF reports</CardDescr
97:       </CardHeader>
98:       <CardContent className="space-y-6 p-8">
99:         { /* Quick Actions */ }
100:        <div className="grid md:grid-cols-3 gap-4">
101:          <Button
102:            onClick={previewPDF}
103:            variant="outline"
104:            className="h-20 flex flex-col items-center justify-center gap-2 bg-transparent"
105:            disabled={isGenerating}
106:          >
107:            <Eye className="h-6 w-6" />
108:            <span>Preview Report</span>
109:          </Button>
110:
111:          <Button
112:            onClick={() => handlePrintOrDownload("print")}
113:            disabled={isGenerating}
114:            className="h-20 flex flex-col items-center justify-center gap-2 bg-blue-600 hover:bg-blue-700"
115:          >
116:            <Printer className="h-6 w-6" />
117:            <span>{isGenerating ? "Generating..." : "Print Report"}</span>
118:          </Button>
119:
120:          <Button
121:            onClick={() => handlePrintOrDownload("download")}
122:            disabled={isGenerating}
123:            className="h-20 flex flex-col items-center justify-center gap-2 bg-green-600 hover:bg-green-700"
124:          >
125:            <Download className="h-6 w-6" />
126:            <span>{isGenerating ? "Preparing..." : "Download PDF"}</span>
127:          </Button>
128:        </div>
129:
130:        { /* PDF Settings */ }
131:        <div className="space-y-4">
132:          <h3 className="text-lg font-semibold text-gray-800">PDF Settings (for Print Dialog)</h3>
133:          <div className="grid md:grid-cols-2 gap-6">
134:            <div className="space-y-4">
135:              <div>
136:                <label className="block text-sm font-medium text-gray-700 mb-2">Page Format</label>
137:                <select
138:                  value={pdfSettings.format}
139:                  onChange={(e) => setPdfSettings((prev) => ({ ...prev, format: e.target.value })))}
140:                  className="w-full border border-gray-300 rounded-md px-3 py-2 focus:outline-none focus:ri
141:                >
142:                  <option value="A4">A4 (210 × 297 mm)</option>
143:                  <option value="Letter">Letter (8.5 × 11 in)</option>
144:                  <option value="Legal">Legal (8.5 × 14 in)</option>
145:                </select>
146:              </div>
147:
148:              <div>
149:                <label className="block text-sm font-medium text-gray-700 mb-2">Orientation</label>
150:                <select
151:                  value={pdfSettings.orientation}
152:                  onChange={(e) => setPdfSettings((prev) => ({ ...prev, orientation: e.target.value })))}
153:                  className="w-full border border-gray-300 rounded-md px-3 py-2 focus:outline-none focus:ri
154:                >
155:                  <option value="portrait">Portrait</option>
156:                  <option value="landscape">Landscape</option>
157:                </select>
158:              </div>
159:
160:              <div>
161:                <label className="block text-sm font-medium text-gray-700 mb-2">Margins</label>
162:                <select
163:                  value={pdfSettings.margins}
164:                  onChange={(e) => setPdfSettings((prev) => ({ ...prev, margins: e.target.value })))}

```

```

165:         className="w-full border border-gray-300 rounded-md px-3 py-2 focus:outline-none focus:ring-1 focus:ring-blue-500"
166:     >
167:         <option value="none">None</option>
168:         <option value="minimum">Minimum</option>
169:         <option value="normal">Normal</option>
170:         <option value="maximum">Maximum</option>
171:     </select>
172: </div>
173: </div>
174:
175: <div className="space-y-4">
176:     <div>
177:         <label className="block text-sm font-medium text-gray-700 mb-2">Quality</label>
178:         <select
179:             value={pdfSettings.quality}
180:             onChange={(e) => setPdfSettings((prev) => ({ ...prev, quality: e.target.value })))}
181:             className="w-full border border-gray-300 rounded-md px-3 py-2 focus:outline-none focus:ring-1 focus:ring-blue-500"
182:         >
183:             <option value="draft">Draft</option>
184:             <option value="normal">Normal</option>
185:             <option value="high">High</option>
186:             <option value="maximum">Maximum</option>
187:         </select>
188:     </div>
189:
190:     <div className="space-y-3">
191:         <label className="flex items-center">
192:             <input
193:                 type="checkbox"
194:                 checked={pdfSettings.includeGraphics}
195:                 onChange={(e) => setPdfSettings((prev) => ({ ...prev, includeGraphics: e.target.checked })))}
196:                 className="mr-2"
197:             />
198:             <span className="text-sm text-gray-700">Include Graphics</span>
199:         </label>
200:
201:         <label className="flex items-center">
202:             <input
203:                 type="checkbox"
204:                 checked={pdfSettings.includeColors}
205:                 onChange={(e) => setPdfSettings((prev) => ({ ...prev, includeColors: e.target.checked })))}
206:                 className="mr-2"
207:             />
208:             <span className="text-sm text-gray-700">Include Colors</span>
209:         </label>
210:     </div>
211: </div>
212: </div>
213: </div>
214:
215: { /* Report Information */ }
216: <div className="bg-gray-50 border border-gray-200 rounded-lg p-4">
217:     <h4 className="font-semibold text-gray-800 mb-3">? Report Information</h4>
218:     <div className="grid md:grid-cols-2 gap-4 text-sm">
219:         <div>
220:             <span className="font-medium">Patient:</span> {reportData.patientInfo.name || "Not specified"}
221:         </div>
222:         <div>
223:             <span className="font-medium">Sample Code:</span> {reportData.patientInfo.sampleCode || "Not specified"}
224:         </div>
225:         <div>
226:             <span className="font-medium">Report Date:</span> {reportData.patientInfo.reportDate || "Not specified"}
227:         </div>
228:         <div>
229:             <span className="font-medium">Total Pages:</span> ~47 pages (estimate)
230:         </div>
231:     </div>
232: </div>
233:
234: { /* Instructions */ }
235: <div className="bg-blue-50 border border-blue-200 rounded-lg p-4">
236:     <h4 className="font-semibold text-blue-800 mb-2">? PDF Generation Instructions</h4>
237:     <ul className="text-sm text-blue-700 space-y-1">

```

```

238:         <li>? Click "Preview Report" to review the content before generating PDF.</li>
239:         <li>? Use "Print Report" or "Download PDF" to open your browser's print dialog.</li>
240:         <li>? In the print dialog, select "Save as PDF" as the destination to download the report.</li>
241:         <li>
242:             ? For optimal appearance, ensure "Background graphics" is enabled and margins are set to "Non
243:             print settings.
244:         </li>
245:         <li>? The report is optimized for A4 paper size in portrait orientation.</li>
246:     </ul>
247: </div>
248: </CardContent>
249: </Card>
250: )
251: }

```

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

```

=====
1: "use client"
2:
3: import { Card, CardContent, CardDescription, CardHeader, CardTitle } from "@components/ui/card"
4: import { Input } from "@components/ui/input"
5: import { Label } from "@components/ui/label"
6: import { Button } from "@components/ui/button"
7: import { Save, RotateCcw, Plus, Trash2 } from "lucide-react"
8: import { Checkbox } from "@components/ui/checkbox"
9: import type { PreventiveHealth } from "@types/report-types"
10:
11: interface PreventiveHealthAdminProps {
12:   preventiveHealth: PreventiveHealth
13:   updatePreventiveHealth: (section: keyof PreventiveHealth, data: any) => void
14:   onSave: () => void
15:   onReset: () => void
16: }
17:
18: export default function PreventiveHealthAdmin({
19:   preventiveHealth,
20:   updatePreventiveHealth,
21:   onSave,
22:   onReset,
23: }: PreventiveHealthAdminProps) {
24:   const addDiagnosticTest = (type: "halfYearly" | "yearly") => {
25:     updatePreventiveHealth("diagnosticTests", {
26:       ...preventiveHealth.diagnosticTests,
27:       [type]: [...preventiveHealth.diagnosticTests[type], ""],
28:     })
29:   }
30:
31:   const updateDiagnosticTest = (type: "halfYearly" | "yearly", index: number, value: string) => {
32:     const newTests = [...preventiveHealth.diagnosticTests[type]]
33:     newTests[index] = value
34:     updatePreventiveHealth("diagnosticTests", {
35:       ...preventiveHealth.diagnosticTests,
36:       [type]: newTests,
37:     })
38:   }
39:
40:   const removeDiagnosticTest = (type: "halfYearly" | "yearly", index: number) => {
41:     updatePreventiveHealth("diagnosticTests", {
42:       ...preventiveHealth.diagnosticTests,
43:       [type]: preventiveHealth.diagnosticTests[type].filter((_, i) => i !== index),
44:     })
45:   }
46:
47:   const addNutritionalSupplement = () => {
48:     updatePreventiveHealth("nutritionalSupplements", [
49:       ...preventiveHealth.nutritionalSupplements,
50:       { supplement: "", needed: true },
51:     ])
52:   }
53:
54:   const updateNutritionalSupplement = (index: number, field: "supplement" | "needed", value: string | boolean) => {

```

```

55:     const newSupplements = [...preventiveHealth.nutritionalSupplements]
56:     newSupplements[index] = { ...newSupplements[index], [field]: value }
57:     updatePreventiveHealth("nutritionalSupplements", newSupplements)
58:   }
59:
60:   const removeNutritionalSupplement = (index: number) => {
61:     updatePreventiveHealth(
62:       "nutritionalSupplements",
63:       preventiveHealth.nutritionalSupplements.filter((_, i) => i !== index),
64:     )
65:   }
66:
67:   return (
68:     <Card className="shadow-lg border-0 bg-white">
69:       <CardHeader className="bg-gradient-to-r from-teal-500 to-cyan-600 text-white rounded-t-lg">
70:         <CardTitle className="text-2xl">? Preventive Health</CardTitle>
71:         <CardDescription className="text-teal-100">
72:           Manage recommended diagnostic tests and nutritional supplements based on genetic insights
73:         </CardDescription>
74:       </CardHeader>
75:       <CardContent className="space-y-8 p-8">
76:         { /* Action Buttons */ }
77:         <div className="flex justify-end gap-2 mb-6">
78:           <Button onClick={onReset} variant="outline" size="sm">
79:             <RotateCcw className="h-4 w-4 mr-2" />
80:             Reset
81:           </Button>
82:           <Button onClick={onSave} size="sm">
83:             <Save className="h-4 w-4 mr-2" />
84:             Save Changes
85:           </Button>
86:         </div>
87:
88:         { /* Diagnostic Tests Section */ }
89:         <div className="space-y-4">
90:           <h3 className="text-xl font-semibold text-gray-800 border-b-2 border-teal-200 pb-2">? Diagnostic Tests</h3>
91:           <div className="grid md:grid-cols-2 gap-6">
92:             { /* Half-Yearly Tests */ }
93:             <Card className="border-2 border-blue-100 bg-blue-50/30">
94:               <CardHeader className="pb-3">
95:                 <CardTitle className="text-lg text-blue-800 flex justify-between items-center">
96:                   Half-Yearly Tests
97:                   <Button
98:                     onClick={() => addDiagnosticTest("halfYearly")}
99:                     size="sm"
100:                    variant="outline"
101:                    className="text-xs"
102:                  >
103:                    <Plus className="h-3 w-3 mr-1" /> Add Test
104:                  </Button>
105:                </CardTitle>
106:              </CardHeader>
107:              <CardContent className="space-y-3">
108:                {preventiveHealth.diagnosticTests.halfYearly.map((test, index) => (
109:                  <div key={index} className="flex items-center gap-2">
110:                    <Input
111:                      value={test}
112:                      onChange={(e) => updateDiagnosticTest("halfYearly", index, e.target.value)}
113:                      placeholder="e.g., PLASMA SUGAR (FASTING & PP)"
114:                      className="border-2 focus:border-blue-500 text-sm"
115:                    >
116:                    <Button
117:                      onClick={() => removeDiagnosticTest("halfYearly", index)}
118:                      size="sm"
119:                      variant="destructive"
120:                      className="px-2"
121:                    >
122:                      <Trash2 className="h-3 w-3" />
123:                    </Button>
124:                  </div>
125:                ))}
126:                {preventiveHealth.diagnosticTests.halfYearly.length === 0 && (
127:                  <p className="text-sm text-gray-500 italic">No half-yearly tests added.</p>

```

```

128:     })
129:     </CardContent>
130: </Card>
131:
132: { /* Yearly Tests */ }
133: <Card className="border-2 border-green-100 bg-green-50/30">
134:   <CardHeader className="pb-3">
135:     <CardTitle className="text-lg text-green-800 flex justify-between items-center">
136:       Yearly Tests
137:       <Button onClick={() => addDiagnosticTest("yearly")} size="sm" variant="outline" class=
138:         <Plus className="h-3 w-3 mr-1" /> Add Test
139:     </Button>
140:   </CardTitle>
141: </CardHeader>
142: <CardContent className="space-y-3">
143:   {preventiveHealth.diagnosticTests.yearly.map((test, index) => (
144:     <div key={index} className="flex items-center gap-2">
145:       <Input
146:         value={test}
147:         onChange={(e) => updateDiagnosticTest("yearly", index, e.target.value)}
148:         placeholder="e.g., HbA1c"
149:         className="border-2 focus:border-green-500 text-sm"
150:       />
151:       <Button
152:         onClick={() => removeDiagnosticTest("yearly", index)}
153:         size="sm"
154:         variant="destructive"
155:         className="px-2"
156:       >
157:         <Trash2 className="h-3 w-3" />
158:       </Button>
159:     </div>
160:   ))}
161:   {preventiveHealth.diagnosticTests.yearly.length === 0 && (
162:     <p className="text-sm text-gray-500 italic">No yearly tests added.</p>
163:   )}
164: </CardContent>
165: </Card>
166: </div>
167: </div>
168:
169: { /* Nutritional Supplements Section */ }
170: <div className="space-y-4">
171:   <h3 className="text-xl font-semibold text-gray-800 border-b-2 border-teal-200 pb-2">
172:     ? Nutritional Supplements
173:   </h3>
174:   <Card className="border-2 border-purple-100 bg-purple-50/30">
175:     <CardHeader className="pb-3">
176:       <CardTitle className="text-lg text-purple-800 flex justify-between items-center">
177:         Recommended Supplements
178:         <Button
179:           onClick={addNutritionalSupplement}
180:           size="sm"
181:           variant="outline"
182:           className="text-xs bg-transparent"
183:         >
184:           <Plus className="h-3 w-3 mr-1" /> Add Supplement
185:         </Button>
186:       </CardTitle>
187:     </CardHeader>
188:     <CardContent className="space-y-3">
189:       {preventiveHealth.nutritionalSupplements.map((supplement, index) => (
190:         <div key={index} className="grid grid-cols-[1fr_auto_auto] items-center gap-2">
191:           <Input
192:             value={supplement.supplement}
193:             onChange={(e) => updateNutritionalSupplement(index, "supplement", e.target.value)}
194:             placeholder="e.g., VITAMIN D"
195:             className="border-2 focus:border-purple-500 text-sm"
196:           />
197:           <div className="flex items-center space-x-2">
198:             <Checkbox
199:               id={`needed-${index}`}
200:               checked={supplement.needed}

```

```

201:             onCheckedChange={{checked: boolean} => updateNutritionalSupplement(index, "needed", checked)}
202:         />
203:         <Label htmlFor={`needed-${index}`} className="text-sm">
204:             Needed
205:         </Label>
206:     </div>
207:     <Button
208:         onClick={() => removeNutritionalSupplement(index)}
209:         size="sm"
210:         variant="destructive"
211:         className="px-2"
212:     >
213:         <Trash2 className="h-3 w-3" />
214:     </Button>
215: </div>
216: ))}
217: {preventiveHealth.nutritionalSupplements.length === 0 && (
218:     <p className="text-sm text-gray-500 italic">No nutritional supplements added.</p>
219: )}
220: </CardContent>
221: </Card>
222: </div>
223:
224: {/* Section Guidelines */}
225: <div className="bg-teal-50 border-2 border-teal-200 rounded-lg p-6">
226:     <h4 className="font-semibold text-teal-800 mb-2">? Preventive Health Guidelines:</h4>
227:     <div className="text-sm text-teal-700 space-y-2">
228:         <ul className="list-disc list-inside space-y-1">
229:             <li>
230:                 This section helps determine the requirement of nutrients in form of supplements and period
231:                 parameters.
232:             </li>
233:             <li>
234:                 <strong>Diagnostic Tests:</strong> Suggests blood tests or other diagnostics to be done hal
235:                 yearly based on genetic predispositions.
236:             </li>
237:             <li>
238:                 <strong>Nutritional Supplements:</strong> Recommends specific supplements and indicates whe
239:                 needed based on genetic analysis.
240:             </li>
241:             <li>Always consult with a healthcare professional before initiating supplement consumption.</li>
242:         </ul>
243:     </div>
244: </div>
245: </CardContent>
246: </Card>
247: )
248: }

```

■ File: components\admin\report-content-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 { ReportContent } from "@/types/report-types"
9:
10: interface ReportContentAdminProps {
11:     content: ReportContent
12:     updateContent: (field: keyof ReportContent, value: string) => void
13:     onSave: () => void
14:     onReset: () => void
15: }
16:
17: export default function ReportContentAdmin({ content, updateContent, onSave, onReset }: ReportContentAdminP
18:     return (
19:         <Card className="shadow-lg border-0 bg-white">
20:             <CardHeader className="bg-gradient-to-r from-green-500 to-teal-600 text-white rounded-t-lg">

```

```

21:         <CardTitle className="text-2xl">? Report Content</CardTitle>
22:         <CardDescription className="text-green-100">Edit the main content sections of the report</CardDescr
23:     </CardHeader>
24:     <CardContent className="space-y-8 p-8">
25:         { /* Action Buttons */ }
26:         <div className="flex justify-end gap-2 mb-6">
27:             <Button onClick={onReset} variant="outline" size="sm">
28:                 <RotateCcw className="h-4 w-4 mr-2" />
29:                 Reset
30:             </Button>
31:             <Button onClick={onSave} size="sm">
32:                 <Save className="h-4 w-4 mr-2" />
33:                 Save Changes
34:             </Button>
35:         </div>
36:
37:         <div className="space-y-3">
38:             <Label htmlFor="disclaimer" className="text-lg font-semibold text-gray-800">
39:                 ?? Disclaimer
40:             </Label>
41:             <Textarea
42:                 id="disclaimer"
43:                 value={content.disclaimer}
44:                 onChange={(e) => updateContent("disclaimer", e.target.value)}
45:                 placeholder="Enter disclaimer text"
46:                 rows={10}
47:                 className="border-2 focus:border-green-500 text-sm leading-relaxed"
48:             />
49:         </div>
50:         <div className="space-y-3">
51:             <Label htmlFor="introduction" className="text-lg font-semibold text-gray-800">
52:                 ? Welcome Letter Introduction
53:             </Label>
54:             <Textarea
55:                 id="introduction"
56:                 value={content.introduction}
57:                 onChange={(e) => updateContent("introduction", e.target.value)}
58:                 placeholder="Enter introduction text"
59:                 rows={8}
60:                 className="border-2 focus:border-green-500 text-sm leading-relaxed"
61:             />
62:         </div>
63:         <div className="space-y-3">
64:             <Label htmlFor="genomicsExplanation" className="text-lg font-semibold text-gray-800">
65:                 ? What is Genomics?
66:             </Label>
67:             <Textarea
68:                 id="genomicsExplanation"
69:                 value={content.genomicsExplanation}
70:                 onChange={(e) => updateContent("genomicsExplanation", e.target.value)}
71:                 placeholder="Enter genomics explanation"
72:                 rows={8}
73:                 className="border-2 focus:border-green-500 text-sm leading-relaxed"
74:             />
75:         </div>
76:         <div className="space-y-3">
77:             <Label htmlFor="genesHealthImpact" className="text-lg font-semibold text-gray-800">
78:                 ? Genes and Health Impact
79:             </Label>
80:             <Textarea
81:                 id="genesHealthImpact"
82:                 value={content.genesHealthImpact}
83:                 onChange={(e) => updateContent("genesHealthImpact", e.target.value)}
84:                 placeholder="Enter genes and health impact content"
85:                 rows={8}
86:                 className="border-2 focus:border-green-500 text-sm leading-relaxed"
87:             />
88:         </div>
89:         <div className="space-y-3">
90:             <Label htmlFor="fundamentalsPRS" className="text-lg font-semibold text-gray-800">
91:                 ? Fundamentals and PRS
92:             </Label>
93:             <Textarea

```



```

94:         id="fundamentalsPRS"
95:         value={content.fundamentalsPRS}
96:         onChange={(e) => updateContent("fundamentalsPRS", e.target.value)}
97:         placeholder="Enter fundamentals and PRS content"
98:         rows={8}
99:         className="border-2 focus:border-green-500 text-sm leading-relaxed"
100:     />
101: </div>
102: <div className="space-y-3">
103:     <Label htmlFor="utilityDoctors" className="text-lg font-semibold text-gray-800">
104:         ??? Utility for Doctors and Dietitians
105:     </Label>
106:     <Textarea
107:         id="utilityDoctors"
108:         value={content.utilityDoctors}
109:         onChange={(e) => updateContent("utilityDoctors", e.target.value)}
110:         placeholder="Enter utility for doctors content"
111:         rows={8}
112:         className="border-2 focus:border-green-500 text-sm leading-relaxed"
113:     />
114: </div>
115: <div className="space-y-3">
116:     <Label htmlFor="microarrayExplanation" className="text-lg font-semibold text-gray-800">
117:         ? Microarray Explanation
118:     </Label>
119:     <Textarea
120:         id="microarrayExplanation"
121:         value={content.microarrayExplanation}
122:         onChange={(e) => updateContent("microarrayExplanation", e.target.value)}
123:         placeholder="Enter microarray explanation"
124:         rows={8}
125:         className="border-2 focus:border-green-500 text-sm leading-relaxed"
126:     />
127: </div>
128: <div className="space-y-3">
129:     <Label htmlFor="microarrayData" className="text-lg font-semibold text-gray-800">
130:         ? Your Microarray Data
131:     </Label>
132:     <Textarea
133:         id="microarrayData"
134:         value={content.microarrayData}
135:         onChange={(e) => updateContent("microarrayData", e.target.value)}
136:         placeholder="Enter microarray data content"
137:         rows={8}
138:         className="border-2 focus:border-green-500 text-sm leading-relaxed"
139:     />
140: </div>
141: </CardContent>
142: </Card>
143: )
144: }

```

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

```

=====
1: "use client"
2:
3: import { Card, CardContent, CardDescription, CardHeader, CardTitle } from "@/components/ui/card"
4: import { Input } from "@/components/ui/input"
5: import { Label } from "@/components/ui/label"
6: import { Button } from "@/components/ui/button"
7: import { Save, RotateCcw } from "lucide-react"
8: import type { ReportSettings } from "@/types/report-types"
9:
10: interface ReportSettingsAdminProps {
11:     settings: ReportSettings
12:     updateSettings: (field: keyof ReportSettings, value: string) => void
13:     onSave: () => void
14:     onReset: () => void
15: }
16:
17: export default function ReportSettingsAdmin({ settings, updateSettings, onSave, onReset }: ReportSettingsAdminProps) {

```

```

18:   return (
19:     <Card className="shadow-lg border-0 bg-white">
20:       <CardHeader className="bg-gradient-to-r from-purple-500 to-pink-600 text-white rounded-t-lg">
21:         <CardTitle className="text-2xl">? Report Settings</CardTitle>
22:         <CardDescription className="text-purple-100">
23:           Customize the appearance and branding of your report
24:         </CardDescription>
25:       </CardHeader>
26:       <CardContent className="space-y-6 p-8">
27:         { /* Action Buttons */ }
28:         <div className="flex justify-end gap-2 mb-6">
29:           <Button onClick={onReset} variant="outline" size="sm">
30:             <RotateCcw className="h-4 w-4 mr-2" />
31:             Reset
32:           </Button>
33:           <Button onClick={onSave} size="sm">
34:             <Save className="h-4 w-4 mr-2" />
35:             Save Changes
36:           </Button>
37:         </div>
38:
39:         <div className="grid md:grid-cols-2 gap-6">
40:           <div className="space-y-2">
41:             <Label htmlFor="title" className="text-sm font-semibold text-gray-700">
42:               Report Title
43:             </Label>
44:             <Input
45:               id="title"
46:               value={settings.title}
47:               onChange={(e) => updateSettings("title", e.target.value)}
48:               className="border-2 focus:border-purple-500 font-bold"
49:               placeholder="e.g., Comprehensive Health Report"
50:             />
51:           </div>
52:           <div className="space-y-2">
53:             <Label htmlFor="subtitle" className="text-sm font-semibold text-gray-700">
54:               Report Subtitle
55:             </Label>
56:             <Input
57:               id="subtitle"
58:               value={settings.subtitle}
59:               onChange={(e) => updateSettings("subtitle", e.target.value)}
60:               className="border-2 focus:border-purple-500"
61:               placeholder="e.g., A Detailed Analysis of Your Genetics"
62:             />
63:           </div>
64:           <div className="space-y-2">
65:             <Label htmlFor="companyName" className="text-sm font-semibold text-gray-700">
66:               Company Name
67:             </Label>
68:             <Input
69:               id="companyName"
70:               value={settings.companyName}
71:               onChange={(e) => updateSettings("companyName", e.target.value)}
72:               className="border-2 focus:border-purple-500"
73:               placeholder="e.g., Health Solutions Inc."
74:             />
75:           </div>
76:           <div className="space-y-2">
77:             <Label htmlFor="headerColor" className="text-sm font-semibold text-gray-700">
78:               Header Color
79:             </Label>
80:             <div className="flex gap-2">
81:               <Input
82:                 id="headerColor"
83:                 type="color"
84:                 value={settings.headerColor}
85:                 onChange={(e) => updateSettings("headerColor", e.target.value)}
86:                 className="w-16 h-10 border-2 focus:border-purple-500"
87:               />
88:               <Input
89:                 value={settings.headerColor}
90:                 onChange={(e) => updateSettings("headerColor", e.target.value)}

```

```

91:         className="flex-1 border-2 focus:border-purple-500 font-mono"
92:         placeholder="#2C3E50"
93:     />
94: </div>
95: </div>
96: <div className="space-y-2">
97:   <Label htmlFor="accentColor" className="text-sm font-semibold text-gray-700">
98:     Accent Color
99:   </Label>
100:   <div className="flex gap-2">
101:     <Input
102:       id="accentColor"
103:       type="color"
104:       value={settings.accentColor}
105:       onChange={(e) => updateSettings("accentColor", e.target.value)}
106:       className="w-16 h-10 border-2 focus:border-purple-500"
107:     />
108:     <Input
109:       value={settings.accentColor}
110:       onChange={(e) => updateSettings("accentColor", e.target.value)}
111:       className="flex-1 border-2 focus:border-purple-500 font-mono"
112:       placeholder="#E74C3C"
113:     />
114:   </div>
115: </div>
116: </div>
117:
118: <div className="space-y-4">
119:   <h3 className="text-lg font-semibold text-gray-800 border-b pb-2">Font Settings</h3>
120:   <div className="grid md:grid-cols-3 gap-6">
121:     <div className="space-y-2">
122:       <Label htmlFor="primaryFont" className="text-sm font-semibold text-gray-700">
123:         Primary Font
124:       </Label>
125:       <Input
126:         id="primaryFont"
127:         value={settings.fonts.primary}
128:         onChange={(e) =>
129:           updateSettings("fonts", JSON.stringify({ ...settings.fonts, primary: e.target.value }))
130:         }
131:         className="border-2 focus:border-purple-500"
132:         placeholder="e.g., Helvetica, Arial, sans-serif"
133:       />
134:     </div>
135:     <div className="space-y-2">
136:       <Label htmlFor="secondaryFont" className="text-sm font-semibold text-gray-700">
137:         Secondary Font
138:       </Label>
139:       <Input
140:         id="secondaryFont"
141:         value={settings.fonts.secondary}
142:         onChange={(e) =>
143:           updateSettings("fonts", JSON.stringify({ ...settings.fonts, secondary: e.target.value }))
144:         }
145:         className="border-2 focus:border-purple-500"
146:         placeholder="e.g., Georgia, serif"
147:       />
148:     </div>
149:     <div className="space-y-2">
150:       <Label htmlFor="monoFont" className="text-sm font-semibold text-gray-700">
151:         Monospace Font
152:       </Label>
153:       <Input
154:         id="monoFont"
155:         value={settings.fonts.mono}
156:         onChange={(e) => updateSettings("fonts", JSON.stringify({ ...settings.fonts, mono: e.target
157:         className="border-2 focus:border-purple-500"
158:         placeholder="e.g., Courier New, monospace"
159:       />
160:     </div>
161:   </div>
162: </div>
163:

```

```

164:         <div className="bg-purple-50 border-2 border-purple-200 rounded-lg p-4">
165:             <h4 className="font-semibold text-purple-800 mb-2">? Customization Tips:</h4>
166:             <ul className="text-sm text-purple-700 space-y-1">
167:                 <li>? Use hex codes for precise color matching (e.g., #RRGGBB).</li>
168:                 <li>? Choose fonts that align with your brand's aesthetic.</li>
169:                 <li>? Ensure sufficient contrast for readability, especially for text.</li>
170:             </ul>
171:         </div>
172:     </CardContent>
173: </Card>
174: )
175: }

```

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

```

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

```

```

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

```

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

```

=====
1: "use client"
2:
3: import { Card, CardContent, CardDescription, CardHeader, CardTitle } from "@/components/ui/card"
4: import { Input } from "@/components/ui/input"
5: import { Label } from "@/components/ui/label"
6: import { Button } from "@/components/ui/button" // Import Button
7: import { Save, RotateCcw } from "lucide-react" // Import icons
8: import type { ReportSettings } from "@/types/report-types"
9:
10: interface SettingsAdminProps {
11:     settings: ReportSettings
12:     updateSettings: (field: keyof ReportSettings, value: string) => void
13:     onSave: () => void // Added onSave
14:     onReset: () => void // Added onReset
15: }
16:
17: export default function SettingsAdmin({ settings, updateSettings, onSave, onReset }: SettingsAdminProps) {
18:     return (
19:         <Card className="shadow-lg border-0 bg-white">
20:             <CardHeader className="bg-gradient-to-r from-purple-500 to-pink-600 text-white rounded-t-lg">
21:                 <CardTitle className="text-2xl">? Report Settings</CardTitle>
22:                 <CardDescription className="text-purple-100">
23:                     Customize the appearance and branding of your report
24:                 </CardDescription>
25:             </CardHeader>
26:             <CardContent className="space-y-6 p-8">
27:                 { /* Action Buttons */ }
28:                 <div className="flex justify-end gap-2 mb-6">
29:                     <Button onClick={onReset} variant="outline" size="sm">
30:                         <RotateCcw className="h-4 w-4 mr-2" />
31:                         Reset
32:                     </Button>
33:                     <Button onClick={onSave} size="sm">
34:                         <Save className="h-4 w-4 mr-2" />
35:                         Save Changes
36:                     </Button>
37:                 </div>
38:

```

```

39: <div className="grid md:grid-cols-2 gap-6">
40:   <div className="space-y-2">
41:     <Label htmlFor="title" className="text-sm font-semibold text-gray-700">
42:       Report Title
43:     </Label>
44:     <Input
45:       id="title"
46:       value={settings.title}
47:       onChange={(e) => updateSettings("title", e.target.value)}
48:       className="border-2 focus:border-purple-500 font-bold"
49:     />
50:   </div>
51:   <div className="space-y-2">
52:     <Label htmlFor="subtitle" className="text-sm font-semibold text-gray-700">
53:       Report Subtitle
54:     </Label>
55:     <Input
56:       id="subtitle"
57:       value={settings.subtitle}
58:       onChange={(e) => updateSettings("subtitle", e.target.value)}
59:       className="border-2 focus:border-purple-500"
60:     />
61:   </div>
62:   <div className="space-y-2">
63:     <Label htmlFor="companyName" className="text-sm font-semibold text-gray-700">
64:       Company Name
65:     </Label>
66:     <Input
67:       id="companyName"
68:       value={settings.companyName}
69:       onChange={(e) => updateSettings("companyName", e.target.value)}
70:       className="border-2 focus:border-purple-500"
71:     />
72:   </div>
73:   <div className="space-y-2">
74:     <Label htmlFor="headerColor" className="text-sm font-semibold text-gray-700">
75:       Header Color
76:     </Label>
77:     <div className="flex gap-2">
78:       <Input
79:         id="headerColor"
80:         type="color"
81:         value={settings.headerColor}
82:         onChange={(e) => updateSettings("headerColor", e.target.value)}
83:         className="w-16 h-10 border-2 focus:border-purple-500"
84:       />
85:       <Input
86:         value={settings.headerColor}
87:         onChange={(e) => updateSettings("headerColor", e.target.value)}
88:         className="flex-1 border-2 focus:border-purple-500 font-mono"
89:       />
90:     </div>
91:   </div>
92:   <div className="space-y-2">
93:     <Label htmlFor="accentColor" className="text-sm font-semibold text-gray-700">
94:       Accent Color
95:     </Label>
96:     <div className="flex gap-2">
97:       <Input
98:         id="accentColor"
99:         type="color"
100:        value={settings.accentColor}
101:        onChange={(e) => updateSettings("accentColor", e.target.value)}
102:        className="w-16 h-10 border-2 focus:border-purple-500"
103:      />
104:      <Input
105:        value={settings.accentColor}
106:        onChange={(e) => updateSettings("accentColor", e.target.value)}
107:        className="flex-1 border-2 focus:border-purple-500 font-mono"
108:      />
109:    </div>
110:  </div>
111: </div>

```

```

112:         <Label htmlFor="primaryFont" className="text-sm font-semibold text-gray-700">
113:             Primary Font
114:         </Label>
115:         <Input
116:             id="primaryFont"
117:             value={settings.fonts.primary}
118:             onChange={(e) => updateSettings("fonts", { ...settings.fonts, primary: e.target.value })}
119:             placeholder="e.g., Helvetica, Arial, sans-serif"
120:             className="border-2 focus:border-purple-500"
121:         />
122:     </div>
123:     <div className="space-y-2">
124:         <Label htmlFor="secondaryFont" className="text-sm font-semibold text-gray-700">
125:             Secondary Font
126:         </Label>
127:         <Input
128:             id="secondaryFont"
129:             value={settings.fonts.secondary}
130:             onChange={(e) => updateSettings("fonts", { ...settings.fonts, secondary: e.target.value })}
131:             placeholder="e.g., Georgia, serif"
132:             className="border-2 focus:border-purple-500"
133:         />
134:     </div>
135:     <div className="space-y-2">
136:         <Label htmlFor="monoFont" className="text-sm font-semibold text-gray-700">
137:             Monospace Font
138:         </Label>
139:         <Input
140:             id="monoFont"
141:             value={settings.fonts.mono}
142:             onChange={(e) => updateSettings("fonts", { ...settings.fonts, mono: e.target.value })}
143:             placeholder="e.g., Courier New, monospace"
144:             className="border-2 focus:border-purple-500"
145:         />
146:     </div>
147: </div>
148: </CardContent>
149: </Card>
150: )
151: }

```

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

```

=====
1: "use client"
2:
3: import { Card, CardContent, CardDescription, 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 { Save, RotateCcw } from "lucide-react"
9: import type { SleepAndRest, SleepData } from "@types/report-types"
10:
11: interface SleepRestAdminProps {
12:     sleepAndRest: SleepAndRest
13:     updateSleepAndRest: (field: keyof SleepAndRest, data: Partial<SleepData>) => void
14:     onSave: () => void
15:     onReset: () => void
16: }
17:
18: export default function SleepRestAdmin({ sleepAndRest, updateSleepAndRest, onSave, onReset }: SleepRestAdminProps) {
19:     const renderSleepField = (field: keyof SleepAndRest, data: SleepData, title: string) => (
20:         <Card key={field} className="border-2 border-indigo-100 bg-indigo-50/30">
21:             <CardHeader className="pb-3">
22:                 <CardTitle className="text-lg text-indigo-800">{title}</CardTitle>
23:             </CardHeader>
24:             <CardContent className="space-y-4">
25:                 <div className="space-y-2">
26:                     <Label className="text-sm font-medium">Genetic Impact</Label>
27:                     <Input
28:                         value={data.impact}

```

```

29:         onChange={(e) => updateSleepAndRest(field, { impact: e.target.value })}
30:         placeholder="e.g., DELAYED SLEEP CYCLE"
31:         className="border-2 focus:border-indigo-500"
32:     />
33: </div>
34: <div className="space-y-2">
35:     <Label className="text-sm font-medium">Intervention Needed</Label>
36:     <Textarea
37:         value={data.intervention}
38:         onChange={(e) => updateSleepAndRest(field, { intervention: e.target.value })}
39:         rows={3}
40:         className="border-2 focus:border-indigo-500 text-sm"
41:         placeholder="e.g., EXERCISE IN THE EVENING AVOID CARBS DURING BREAKFAST"
42:     />
43: </div>
44: </CardContent>
45: </Card>
46: )
47:
48: return (
49:     <Card className="shadow-lg border-0 bg-white">
50:         <CardHeader className="bg-gradient-to-r from-indigo-500 to-blue-600 text-white rounded-t-lg">
51:             <CardTitle className="text-2xl">? Sleep & Rest</CardTitle>
52:             <CardDescription className="text-indigo-100">
53:                 Manage genetic influences on sleep patterns, quality, and stress sensitivity
54:             </CardDescription>
55:         </CardHeader>
56:         <CardContent className="space-y-8 p-8">
57:             { /* Action Buttons */ }
58:             <div className="flex justify-end gap-2 mb-6">
59:                 <Button onClick={onReset} variant="outline" size="sm">
60:                     <RotateCcw className="h-4 w-4 mr-2" />
61:                     Reset
62:                 </Button>
63:                 <Button onClick={onSave} size="sm">
64:                     <Save className="h-4 w-4 mr-2" />
65:                     Save Changes
66:                 </Button>
67:             </div>
68:
69:             { /* Sleep & Rest Fields */ }
70:             <div className="grid md:grid-cols-2 lg:grid-cols-2 gap-6">
71:                 {Object.entries(sleepAndRest).map(([key, item]) =>
72:                     renderSleepField(
73:                         key as keyof SleepAndRest,
74:                         item as SleepData,
75:                         key.replace(/[A-Z]/g, " $1").replace(/^./, (str) => str.toUpperCase()),
76:                     ),
77:                 )}
78:             </div>
79:
80:             { /* Section Guidelines */ }
81:             <div className="bg-indigo-50 border-2 border-indigo-200 rounded-lg p-6">
82:                 <h4 className="font-semibold text-indigo-800 mb-2">? Sleep & Rest Guidelines:</h4>
83:                 <div className="text-sm text-indigo-700 space-y-2">
84:                     <ul className="list-disc list-inside space-y-1">
85:                         <li>
86:                             <strong>Sleep Cycle:</strong> Genetic influence on circadian rhythms and sleep timing.
87:                         </li>
88:                         <li>
89:                             <strong>Sleep Apnea:</strong> Genetic predisposition to sleep-disordered breathing.
90:                         </li>
91:                         <li>
92:                             <strong>Sleep Depth:</strong> Genetic factors affecting the quality and restorative nature
93:                         </li>
94:                         <li>
95:                             <strong>Stress Sensitivity:</strong> Genetic tendency to react to stress, impacting sleep.
96:                         </li>
97:                         <li>Understanding these factors can guide personalized approaches to sleep health management.</li>
98:                         <li>Sleep is crucial for overall physical and mental well-being.</li>
99:                     </ul>
100:                 </div>
101:             </div>

```



```

102:         </CardContent>
103:     </Card>
104: )
105: }

```

■ File: components\admin\sports-fitness-admin.tsx

```

=====
1: "use client"
2:
3: import { Card, CardContent, CardDescription, 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 { Save, RotateCcw } from "lucide-react"
9: import { useState } from "react"
10: import type { SportsAndFitness, ExerciseData } from "@types/report-types"
11:
12: interface SportsFitnessAdminProps {
13:   sportsAndFitness: SportsAndFitness
14:   updateSportsAndFitness: (section: keyof SportsAndFitness, field: string, data: Partial<ExerciseData>) => void
15:   onSave: () => void
16:   onReset: () => void
17: }
18:
19: export default function SportsFitnessAdmin({
20:   sportsAndFitness,
21:   updateSportsAndFitness,
22:   onSave,
23:   onReset,
24: }: SportsFitnessAdminProps) {
25:   const [activeSection, setActiveSection] = useState<keyof SportsAndFitness>("exerciseType")
26:
27:   const renderExerciseField = (section: keyof SportsAndFitness, field: string, data: ExerciseData, title: string) => {
28:     <Card key={field} className="border-2 border-green-100 bg-green-50/30">
29:       <CardHeader className="pb-3">
30:         <CardTitle className="text-lg text-green-800">{title}</CardTitle>
31:       </CardHeader>
32:       <CardContent className="space-y-4">
33:         <div className="space-y-2">
34:           <Label className="text-sm font-medium">Level</Label>
35:           <Input
36:             value={data.level}
37:             onChange={(e) => updateSportsAndFitness(section, field, { level: e.target.value })}
38:             placeholder="e.g., High Endurance Potential"
39:             className="border-2 focus:border-green-500"
40:           />
41:         </div>
42:         <div className="space-y-2">
43:           <Label className="text-sm font-medium">Description</Label>
44:           <Textarea
45:             value={data.description}
46:             onChange={(e) => updateSportsAndFitness(section, field, { description: e.target.value })}
47:             rows={3}
48:             className="border-2 focus:border-green-500 text-sm"
49:           />
50:         </div>
51:       </CardContent>
52:     </Card>
53:   )
54:
55:   const getSectionTitle = (section: keyof SportsAndFitness) => {
56:     switch (section) {
57:       case "exerciseType":
58:         return "???? Exercise Type Suitability"
59:       case "performance":
60:         return "? Performance Factors"
61:       default:
62:         return section
63:     }
64:   }

```

```

65:
66: const getSectionIcon = (section: keyof SportsAndFitness) => {
67:     switch (section) {
68:         case "exerciseType":
69:             return "???"
70:         case "performance":
71:             return "?"
72:         default:
73:             return "?"
74:     }
75: }
76:
77: return (
78:     <Card className="shadow-lg border-0 bg-white">
79:         <CardHeader className="bg-gradient-to-r from-green-500 to-emerald-600 text-white rounded-t-lg">
80:             <CardTitle className="text-2xl">??? Sports & Fitness</CardTitle>
81:             <CardDescription className="text-green-100">
82:                 Configure exercise type suitability and performance factors
83:             </CardDescription>
84:         </CardHeader>
85:         <CardContent className="space-y-8 p-8">
86:             { /* Action Buttons */ }
87:             <div className="flex justify-between items-center">
88:                 <div className="flex gap-2 flex-wrap">
89:                     {(Object.keys(sportsAndFitness) as Array<keyof SportsAndFitness>).map((section) => (
90:                         <Button
91:                             key={section}
92:                             variant={activeSection === section ? "default" : "outline"}
93:                             size="sm"
94:                             onClick={() => setActiveSection(section)}
95:                             className="flex items-center gap-2"
96:                         >
97:                             <span>{getSectionIcon(section)}</span>
98:                             <span className="capitalize">{section.replace(/[A-Z]/g, " $1")}</span>
99:                         </Button>
100:                     ))}
101:                 </div>
102:                 <div className="flex gap-2">
103:                     <Button onClick={onReset} variant="outline" size="sm">
104:                         <RotateCcw className="h-4 w-4 mr-2" />
105:                         Reset
106:                     </Button>
107:                     <Button onClick={onSave} size="sm">
108:                         <Save className="h-4 w-4 mr-2" />
109:                         Save Changes
110:                     </Button>
111:                 </div>
112:             </div>
113:
114:             { /* Section Content */ }
115:             <div className="space-y-6">
116:                 <h3 className="text-xl font-semibold text-gray-800 border-b-2 border-green-200 pb-2">
117:                     {getSectionTitle(activeSection)}
118:                 </h3>
119:
120:                 <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-6">
121:                     {Object.entries(sportsAndFitness[activeSection]).map(([key, item]) =>
122:                         renderExerciseField(
123:                             activeSection,
124:                             key,
125:                             item as ExerciseData,
126:                             key.replace(/[A-Z]/g, " $1").replace(/^.\/, (str) => str.toUpperCase(),
127:                         ),
128:                     )}
129:                 </div>
130:             </div>
131:
132:             { /* Section Guidelines */ }
133:             <div className="bg-green-50 border-2 border-green-200 rounded-lg p-6">
134:                 <h4 className="font-semibold text-green-800 mb-2">{getSectionTitle(activeSection)} Guidelines:<
135:                 <div className="text-sm text-green-700 space-y-2">
136:                     {activeSection === "exerciseType" && (
137:                         <ul className="list-disc list-inside space-y-1">

```

```

138:         <li>
139:             <strong>Endurance Potential:</strong> Genetic predisposition for long-duration activities
140:         </li>
141:         <li>
142:             <strong>Power Potential:</strong> Capacity for short, intense bursts of energy.
143:         </li>
144:         <li>
145:             <strong>Strength Profile:</strong> Genetic tendency for muscle strength and growth.
146:         </li>
147:         <li>
148:             <strong>Exercise Time:</strong> Optimal time of day for exercise based on circadian rhythm
149:         </li>
150:     </ul>
151: )}
152: {activeSection === "performance" && (
153:     <ul className="list-disc list-inside space-y-1">
154:         <li>
155:             <strong>Endurance Capacity:</strong> Aerobic efficiency and stamina.
156:         </li>
157:         <li>
158:             <strong>Fatigue Resistance:</strong> Ability to resist muscle fatigue during prolonged ac
159:         </li>
160:         <li>
161:             <strong>Blood Flow:</strong> Efficiency of blood circulation during exercise.
162:         </li>
163:         <li>
164:             <strong>Oxygen Efficiency (VO2 Max):</strong> Body's ability to use oxygen during intense
165:         </li>
166:         <li>
167:             <strong>Water Loss:</strong> Tendency for fluid and electrolyte loss.
168:         </li>
169:         <li>
170:             <strong>Lactate Clearing:</strong> Efficiency in removing lactic acid from muscles.
171:         </li>
172:         <li>
173:             <strong>Recovery Efficiency:</strong> Speed and effectiveness of post-exercise recovery.
174:         </li>
175:         <li>
176:             <strong>Body Composition:</strong> Genetic predisposition for muscle mass vs. fat accumul
177:         </li>
178:         <li>
179:             <strong>Injury Risk:</strong> Genetic susceptibility to various sports-related injuries.
180:         </li>
181:         <li>
182:             <strong>Achilles Tendon/Ligament/Rupture/Muscle Cramps:</strong> Specific injury predispo
183:         </li>
184:     </ul>
185: )}
186: </div>
187: </div>
188: </CardContent>
189: </Card>
190: )
191: }

```

■ File: components\admin\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="list-disc list-inside 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\comprehensive-report-viewer.tsx

```
=====
1: "use client";
2: import type {
3:   ComprehensiveReportData,
4:   AllergyData,
5: } from "@/types/report-types";
6: import React from "react";
7:
8: import { Card, CardContent, CardHeader, CardTitle } from "@/components/ui/card";
9: import { Badge } from "@/components/ui/badge";
10: import { Separator } from "@/components/ui/separator";
11:
12: interface ComprehensiveReportViewerProps {
13:   reportData: ComprehensiveReportData;
14: }
15:
16: const SectionTitle = ({ title, icon }: { title: string; icon: string }) => (
17:   <h2 className="text-3xl font-bold text-gray-800 mb-6 flex items-center gap-3 page-break-before">
18:     <span className="text-4xl">{icon}</span>
19:     {title}
20:   </h2>
21: );
22:
23: const SubSectionTitle = ({ title, icon }: { title: string; icon?: string }) => (
24:   <h3 className="text-2xl font-semibold text-gray-700 mb-4 mt-8 flex items-center gap-2">
25:     {icon && <span className="text-2xl">{icon}</span>}
26:     {title}
27:   </h3>
28: );
29:
30: const DataCard = ({
31:   title,
32:   children,
33: }: {
34:   title: string;
35:   children: React.ReactNode;
36: }) => (
37:   <Card className="border-2 border-gray-200 bg-gray-50/50 shadow-sm">
38:     <CardHeader className="pb-2">
39:       <CardTitle className="text-lg font-semibold text-gray-800">
40:         {title}
41:       </CardTitle>
42:     </CardHeader>
43:     <CardContent className="text-sm text-gray-700">{children}</CardContent>
44:   </Card>
45: );
46:
47: export default function ComprehensiveReportViewer({
48:   reportData,
49: }: ComprehensiveReportViewerProps) {
50:   const {
51:     patientInfo,
52:     content,
53:     settings,
54:     dynamicDietFieldDefinitions, // NEW
55:     patientDietAnalysisResults, // NEW
56:     nutritionData,
57:     sportsAndFitness,
58:     lifestyleConditions,
59:     metabolicCore,
60:     digestiveHealth,
61:     genesAndAddiction,
62:     sleepAndRest,
63:     allergiesAndSensitivity,
64:     geneTestResults,
65:     metabolicSummary,
66:     preventiveHealth,
67:     familyGeneticImpact,
68:     summaries,
69:   } = reportData;
70:
71:   const getImpactColor = (impact: string) => {
```

```

72:     if (
73:         impact.toLowerCase().includes("high") ||
74:         impact.toLowerCase().includes("increased")
75:     ) {
76:         return "bg-red-100 text-red-800 border-red-300";
77:     }
78:     if (
79:         impact.toLowerCase().includes("low") ||
80:         impact.toLowerCase().includes("reduced")
81:     ) {
82:         return "bg-green-100 text-green-800 border-green-300";
83:     }
84:     if (
85:         impact.toLowerCase().includes("normal") ||
86:         impact.toLowerCase().includes("average") ||
87:         impact.toLowerCase().includes("balanced")
88:     ) {
89:         return "bg-blue-100 text-blue-800 border-blue-300";
90:     }
91:     return "bg-gray-100 text-gray-800 border-gray-300";
92: };
93:
94: const getStatusColor = (status: "strength" | "improvement") => {
95:     return status === "strength"
96:         ? "bg-green-100 text-green-800 border-green-300"
97:         : "bg-red-100 text-red-800 border-red-300";
98: };
99:
100: // Helper to group dynamic diet fields by category
101: const groupedDietFields = patientDietAnalysisResults.reduce((acc, result) => {
102:     const fieldDef = dynamicDietFieldDefinitions.find(
103:         (def) => def.id === result.fieldId
104:     );
105:     if (fieldDef) {
106:         if (!acc[fieldDef.category]) {
107:             acc[fieldDef.category] = [];
108:         }
109:         acc[fieldDef.category].push({ ...result, label: fieldDef.label });
110:     }
111:     return acc;
112: }, {} as Record<string, (typeof patientDietAnalysisResults)[0] & { label: string }[]>);
113:
114: return (
115:     <div className="container mx-auto py-10 px-6 max-w-4xl bg-white shadow-lg min-h-screen">
116:         { /* Report Header */ }
117:         <header
118:             className="text-center mb-12 py-8 rounded-lg"
119:             style={{ backgroundColor: settings.headerColor, color: "#fff" }}
120:         >
121:             <h1
122:                 className="text-5xl font-extrabold mb-2"
123:                 style={{ fontFamily: settings.fonts.primary }}
124:             >
125:                 {settings.title}
126:             </h1>
127:             <p
128:                 className="text-2xl font-light"
129:                 style={{ fontFamily: settings.fonts.secondary }}
130:             >
131:                 {settings.subtitle}
132:             </p>
133:             <p
134:                 className="text-lg mt-4"
135:                 style={{ fontFamily: settings.fonts.secondary }}
136:             >
137:                 Patient: <span className="font-semibold">{patientInfo.name}</span> |
138:                 Sample Code:{" "}
139:                 <span className="font-mono">{patientInfo.sampleCode}</span>
140:             </p>
141:             <p className="text-md" style={{ fontFamily: settings.fonts.secondary }}>
142:                 Report Date: {patientInfo.reportDate} | Generated by:{" "}
143:                 {settings.companyName}
144:             </p>

```

```

145:     </header>
146:
147:     { /* Patient Information */ }
148:     <SectionTitle title="Patient Information" icon="?" />
149:     <div className="grid md:grid-cols-2 gap-6 mb-8">
150:         <DataCard title="Name">
151:             <p>{patientInfo.name}</p>
152:         </DataCard>
153:         <DataCard title="Gender">
154:             <p>{patientInfo.gender}</p>
155:         </DataCard>
156:         <DataCard title="Birth Date">
157:             <p>{patientInfo.birthDate}</p>
158:         </DataCard>
159:         <DataCard title="Sample Code">
160:             <p class="font-mono">{patientInfo.sampleCode}</p>
161:         </DataCard>
162:         <DataCard title="Sample Collection Date">
163:             <p>{patientInfo.sampleDate}</p>
164:         </DataCard>
165:         <DataCard title="Report Generation Date">
166:             <p>{patientInfo.reportDate}</p>
167:         </DataCard>
168:
169:         <Separator className="mt-12 col-span-full" />
170:
171:         <h3 className="text-center font-bold mb-3 text-gray-600 col-span-full">
172:             Report Authentication & Analytics
173:         </h3>
174:
175:         <DataCard title="Report Authentication & Analytics">
176:             <p>{patientInfo.reportAuth}</p>
177:         </DataCard>
178:         <DataCard title="Genomic Data Analytics">
179:             <p>{patientInfo.genomicAnalytics}</p>
180:         </DataCard>
181:         <DataCard title="Checked & Verified By">
182:             <p>{patientInfo.checkedBy}</p>
183:         </DataCard>
184:         <DataCard title="Scientific Content Verified By">
185:             <p>{patientInfo.scientificContent}</p>
186:         </DataCard>
187:
188:         { /* Full-width Disclaimer */ }
189:         <div className="col-span-full">
190:             <DataCard title="Disclaimer">
191:                 <p class="text-justify whitespace-pre-wrap break-words mb-8">
192:                     {patientInfo.disclaimer}
193:                 </p>
194:             </DataCard>
195:         </div>
196:     </div>
197:
198:     <Separator className="my-12" />
199:
200:     { /* Report Content */ }
201:     <SectionTitle title="Report Content" icon="?" />
202:     <div className="space-y-8 mb-8">
203:         <DataCard title="Welcome Letter Introduction">
204:             <p class="text-justify">{content.introduction}</p>
205:         </DataCard>
206:         <DataCard title="What is Genomics?">
207:             <p class="text-justify">{content.genomicsExplanation}</p>
208:         </DataCard>
209:         <DataCard title="Genes and Health Impact">
210:             <p class="text-justify">{content.genesHealthImpact}</p>
211:         </DataCard>
212:         <DataCard title="Fundamentals and PRS">
213:             <p class="text-justify">{content.fundamentalsPRS}</p>
214:         </DataCard>
215:         <DataCard title="Utility for Doctors and Dietitians">
216:             <p class="text-justify">{content.utilityDoctors}</p>
217:         </DataCard>

```

```

218:         <DataCard title="Microarray Explanation">
219:             <p className="text-justify">{content.microarrayExplanation}</p>
220:         </DataCard>
221:         <DataCard title="Your Microarray Data">
222:             <p className="text-justify">{content.microarrayData}</p>
223:         </DataCard>
224:     </div>
225:     <Separator className="my-12" />
226:
227:     {/* Dynamic Diet Analysis */}
228:     <SectionTitle title="Diet Analysis" icon="?" />
229:     {Object.keys(groupedDietFields).length === 0 && (
230:         <p className="text-center text-gray-500 italic">
231:             No diet analysis results available.
232:         </p>
233:     )}
234:     {Object.entries(groupedDietFields).map(([category, fields]) => (
235:         <React.Fragment key={category}>
236:             <SubSectionTitle title={category} icon="?" />
237:             <div className="grid md:grid-cols-2 gap-6 mb-8">
238:                 {fields.map((data, index) => (
239:                     <DataCard key={index} title={data.label}>
240:                         <p>
241:                             <strong>Score:</strong> {data.score}/10
242:                         </p>
243:                         <p>
244:                             <strong>Level:</strong> {data.level}
245:                         </p>
246:                         <p>
247:                             <strong>Recommendation:</strong> {data.recommendation}
248:                         </p>
249:                     </DataCard>
250:                 ))}
251:             </div>
252:         </React.Fragment>
253:     )}
254:     <Separator className="my-12" />
255:
256:     {/* Nutrition Data */}
257:     <SectionTitle title="Nutrition Analysis" icon="?" />
258:     <SubSectionTitle title="Vitamins" icon="?" />
259:     <div className="grid md:grid-cols-2 gap-6 mb-8">
260:         {Object.entries(nutritionData.vitamins).map(([key, data]) => (
261:             <DataCard
262:                 key={key}
263:                 title={key
264:                     .replace(/[A-Z])/g, " $1")
265:                     .replace(/^../, (str) => str.toUpperCase())}
266:             >
267:                 <p>
268:                     <strong>Score:</strong> {data.score}/10
269:                 </p>
270:                 <p>
271:                     <strong>Health Impact:</strong> {data.healthImpact}
272:                 </p>
273:                 <p>
274:                     <strong>Intake Level:</strong> {data.intakeLevel}
275:                 </p>
276:                 <p>
277:                     <strong>Source:</strong> {data.source}
278:                 </p>
279:             </DataCard>
280:         ))}
281:     </div>
282:
283:     <SubSectionTitle title="Fatty Acids" icon="?" />
284:     <div className="grid md:grid-cols-2 gap-6 mb-8">
285:         {Object.entries(nutritionData.fattyAcids).map(([key, data]) => (
286:             <DataCard
287:                 key={key}
288:                 title={key
289:                     .replace(/[A-Z])/g, " $1")
290:                     .replace(/^../, (str) => str.toUpperCase())}

```



```

291:         >
292:         <p>
293:             <strong>Score:</strong> {data.score}/10
294:         </p>
295:         <p>
296:             <strong>Health Impact:</strong> {data.healthImpact}
297:         </p>
298:         <p>
299:             <strong>Intake Level:</strong> {data.intakeLevel}
300:         </p>
301:         <p>
302:             <strong>Source:</strong> {data.source}
303:         </p>
304:     </DataCard>
305:   )})
306: </div>
307:
308: <SubSectionTitle title="Essential Elements" icon="?" />
309: <div className="grid md:grid-cols-2 gap-6 mb-8">
310:   {Object.entries(nutritionData.elements).map(([key, data]) => (
311:     <DataCard
312:       key={key}
313:       title={key
314:         .replace(/([A-Z])/g, " $1")
315:         .replace(/^./, (str) => str.toUpperCase())}
316:     >
317:       <p>
318:         <strong>Score:</strong> {data.score}/10
319:       </p>
320:       <p>
321:         <strong>Health Impact:</strong> {data.healthImpact}
322:       </p>
323:       <p>
324:         <strong>Intake Level:</strong> {data.intakeLevel}
325:       </p>
326:       <p>
327:         <strong>Source:</strong> {data.source}
328:       </p>
329:     </DataCard>
330:   )})
331: </div>
332:
333: <SubSectionTitle title="Complex Nutrients" icon="?" />
334: <div className="grid md:grid-cols-2 gap-6 mb-8">
335:   {Object.entries(nutritionData.complexNutrients).map(([key, data]) => (
336:     <DataCard
337:       key={key}
338:       title={key
339:         .replace(/([A-Z])/g, " $1")
340:         .replace(/^./, (str) => str.toUpperCase())}
341:     >
342:       <p>
343:         <strong>Score:</strong> {data.score}/10
344:       </p>
345:       <p>
346:         <strong>Health Impact:</strong> {data.healthImpact}
347:       </p>
348:       <p>
349:         <strong>Intake Level:</strong> {data.intakeLevel}
350:       </p>
351:       <p>
352:         <strong>Source:</strong> {data.source}
353:       </p>
354:     </DataCard>
355:   )})
356: </div>
357: <Separator className="my-12" />
358:
359: {/* Sports & Fitness */}
360: <SectionTitle title="Sports & Fitness" icon="?????" />
361: <SubSectionTitle title="Exercise Type Suitability" icon="?????" />
362: <div className="grid md:grid-cols-2 gap-6 mb-8">
363:   {Object.entries(sportsAndFitness.exerciseType).map(([key, data]) => (

```

```

364:         <DataCard
365:             key={key}
366:             title={key
367:                 .replace(/([A-Z])/g, " $1")
368:                 .replace(/^../, (str) => str.toUpperCase())}
369:         >
370:             <p>
371:                 <strong>Level:</strong> {data.level}
372:             </p>
373:             <p>
374:                 <strong>Description:</strong> {data.description}
375:             </p>
376:         </DataCard>
377:     )}
378: </div>
379:
380: <SubSectionTitle title="Performance Factors" icon="?" />
381: <div className="grid md:grid-cols-2 gap-6 mb-8">
382:     {Object.entries(sportsAndFitness.performance).map(([key, data]) => (
383:         <DataCard
384:             key={key}
385:             title={key
386:                 .replace(/([A-Z])/g, " $1")
387:                 .replace(/^../, (str) => str.toUpperCase())}
388:         >
389:             <p>
390:                 <strong>Level:</strong> {data.level}
391:             </p>
392:             <p>
393:                 <strong>Description:</strong> {data.description}
394:             </p>
395:         </DataCard>
396:     )}
397: </div>
398: <Separator className="my-12" />
399:
400: {/* Lifestyle Conditions */}
401: <SectionTitle title="Lifestyle Conditions" icon="?" />
402: <SubSectionTitle title="Heart & Vascular Health" icon="??" />
403: <div className="grid md:grid-cols-2 gap-6 mb-8">
404:     {Object.entries(lifestyleConditions.heartVascularHealth).map(
405:         ([key, data]) => (
406:             <DataCard
407:                 key={key}
408:                 title={key
409:                     .replace(/([A-Z])/g, " $1")
410:                     .replace(/^../, (str) => str.toUpperCase())}
411:             >
412:                 <p>
413:                     <strong>Status:</strong>{" "}
414:                     <Badge className={getStatusColor(data.status)}>
415:                         {data.status}
416:                     </Badge>
417:                 </p>
418:             </DataCard>
419:         )
420:     )}
421: </div>
422:
423: <SubSectionTitle title="Diabetes (Diabetes & Obesity)" icon="?" />
424: <div className="grid md:grid-cols-2 gap-6 mb-8">
425:     {Object.entries(lifestyleConditions.diabetes).map(([key, data]) => (
426:         <DataCard
427:             key={key}
428:             title={key
429:                 .replace(/([A-Z])/g, " $1")
430:                 .replace(/^../, (str) => str.toUpperCase())}
431:         >
432:             <p>
433:                 <strong>Status:</strong>{" "}
434:                 <Badge className={getStatusColor(data.status)}>
435:                     {data.status}
436:                 </Badge>

```

```

437:         </p>
438:     </DataCard>
439:   )}
440: </div>
441:
442: <SubSectionTitle title="Liver Health" icon="?" />
443: <div className="grid md:grid-cols-2 gap-6 mb-8">
444:   {Object.entries(lifestyleConditions.liverHealth).map(([key, data]) => (
445:     <DataCard
446:       key={key}
447:       title={key
448:         .replace(/[A-Z]/g, " $1")
449:         .replace(/^.\/, (str) => str.toUpperCase())}
450:     >
451:       <p>
452:         <strong>Status:</strong>{ " "}
453:         <Badge className={getStatusColor(data.status)}>
454:           {data.status}
455:         </Badge>
456:       </p>
457:     </DataCard>
458:   )})
459: </div>
460:
461: <SubSectionTitle title="Bone Health" icon="?" />
462: <div className="grid md:grid-cols-2 gap-6 mb-8">
463:   {Object.entries(lifestyleConditions.boneHealth).map(([key, data]) => (
464:     <DataCard
465:       key={key}
466:       title={key
467:         .replace(/[A-Z]/g, " $1")
468:         .replace(/^.\/, (str) => str.toUpperCase())}
469:     >
470:       <p>
471:         <strong>Status:</strong>{ " "}
472:         <Badge className={getStatusColor(data.status)}>
473:           {data.status}
474:         </Badge>
475:       </p>
476:     </DataCard>
477:   )})
478: </div>
479:
480: <SubSectionTitle title="Gut Health" icon="?" />
481: <div className="grid md:grid-cols-2 gap-6 mb-8">
482:   {Object.entries(lifestyleConditions.gutHealth).map(([key, data]) => (
483:     <DataCard
484:       key={key}
485:       title={key
486:         .replace(/[A-Z]/g, " $1")
487:         .replace(/^.\/, (str) => str.toUpperCase())}
488:     >
489:       <p>
490:         <strong>Status:</strong>{ " "}
491:         <Badge className={getStatusColor(data.status)}>
492:           {data.status}
493:         </Badge>
494:       </p>
495:     </DataCard>
496:   )})
497: </div>
498:
499: <SubSectionTitle title="Hormone System" icon="?" />
500: <div className="grid md:grid-cols-2 gap-6 mb-8">
501:   {Object.entries(lifestyleConditions.hormoneSystem).map(
502:     ([key, data]) => (
503:       <DataCard
504:         key={key}
505:         title={key
506:           .replace(/[A-Z]/g, " $1")
507:           .replace(/^.\/, (str) => str.toUpperCase())}
508:       >
509:         <p>

```

```

510:         <strong>Status:</strong>{" "}
511:         <Badge className={getStatusColor(data.status)}>
512:             {data.status}
513:         </Badge>
514:     </p>
515: </DataCard>
516: )
517: })
518: </div>
519: <Separator className="my-12" />
520:
521: { /* Metabolic Core */ }
522: <SectionTitle title="Metabolic Core" icon="?" />
523: <div className="space-y-8 mb-8">
524:     {Object.entries(metabolicCore).map(([key, data]) => (
525:         <Card
526:             key={key}
527:             className="border-2 border-purple-200 bg-purple-50/50 shadow-sm"
528:         >
529:             <CardHeader className="pb-2">
530:                 <CardTitle className="text-lg font-semibold text-purple-800 flex items-center justify-between">
531:                     {key}
532:                     .replace(/[A-Z]/g, " $1")
533:                     .replace(/^./, (str) => str.toUpperCase())
534:                 <Badge className={getImpactColor(data.impact)}>
535:                     {data.impact}
536:                 </Badge>
537:             </CardTitle>
538:             </CardHeader>
539:             <CardContent className="text-sm text-gray-700 space-y-2">
540:                 <p>
541:                     <strong>Genes:</strong> {data.genes.join(", ")}
542:                 </p>
543:                 <p>
544:                     <strong>Your Genotype:</strong>{" "}
545:                     <span className="font-mono">{data.genotype}</span>
546:                 </p>
547:                 <p>
548:                     <strong>Advice:</strong> {data.advice}
549:                 </p>
550:             </CardContent>
551:         </Card>
552:     ))}
553: </div>
554: <Separator className="my-12" />
555:
556: { /* Digestive Health */ }
557: <SectionTitle title="Digestive Health" icon="?" />
558: <div className="grid md:grid-cols-2 gap-6 mb-8">
559:     {Object.entries(digestiveHealth).map(([key, data]) => (
560:         <DataCard
561:             key={key}
562:             title={key}
563:                 .replace(/[A-Z]/g, " $1")
564:                 .replace(/^./, (str) => str.toUpperCase())
565:         >
566:             <p>
567:                 <strong>Level:</strong> {data.level}
568:             </p>
569:             <p>
570:                 <strong>Description:</strong> {data.description}
571:             </p>
572:         </DataCard>
573:     ))}
574: </div>
575: <Separator className="my-12" />
576:
577: { /* Genes & Addiction */ }
578: <SectionTitle title="Genes & Addiction" icon="?" />
579: <div className="grid md:grid-cols-2 gap-6 mb-8">
580:     {Object.entries(genesAndAddiction).map(([key, data]) => (
581:         <DataCard
582:             key={key}

```

```

583:         title={key
584:             .replace(/[A-Z])/g, " $1")
585:             .replace(/^.\/, (str) => str.toUpperCase())}
586:     >
587:         <p>
588:             <strong>Tendency:</strong> {data.tendency}
589:         </p>
590:         <p>
591:             <strong>Description:</strong> {data.description}
592:         </p>
593:     </DataCard>
594: }}
595: </div>
596: <Separator className="my-12" />
597:
598: {/* Sleep & Rest */}
599: <SectionTitle title="Sleep & Rest" icon="?" />
600: <div className="grid md:grid-cols-2 gap-6 mb-8">
601:     {Object.entries(sleepAndRest).map([key, data]) => (
602:         <DataCard
603:             key={key}
604:             title={key
605:                 .replace(/[A-Z])/g, " $1")
606:                 .replace(/^.\/, (str) => str.toUpperCase())}
607:         >
608:             <p>
609:                 <strong>Impact:</strong> {data.impact}
610:             </p>
611:             <p>
612:                 <strong>Intervention:</strong> {data.intervention}
613:             </p>
614:         </DataCard>
615:     )}}
616: </div>
617: <Separator className="my-12" />
618:
619: {/* Allergies & Sensitivity */}
620: <SectionTitle title="Allergies & Sensitivity" icon="?" />
621: <div className="grid md:grid-cols-2 gap-6 mb-8">
622:     {Object.entries(allergiesAndSensitivity)
623:         .filter([key]) => key !== "generalAdvice")
624:         .map([key, data]) => (
625:         <DataCard
626:             key={key}
627:             title={key
628:                 .replace(/[A-Z])/g, " $1")
629:                 .replace(/^.\/, (str) => str.toUpperCase())}
630:         >
631:             <p>
632:                 <strong>Tendency:</strong> {(data as AllergyData).tendency}
633:             </p>
634:         </DataCard>
635:     )}}
636: </div>
637: <DataCard title="General Advice" className="mt-8">
638:     <p className="text-justify">{allergiesAndSensitivity.generalAdvice}</p>
639: </DataCard>
640: <Separator className="my-12" />
641:
642: {/* Preventive Health */}
643: <SectionTitle title="Preventive Health" icon="?" />
644: <SubSectionTitle title="Diagnostic Tests" icon="?" />
645: <div className="grid md:grid-cols-2 gap-6 mb-8">
646:     <DataCard title="Half-Yearly Tests">
647:         <ul className="list-disc list-inside">
648:             {preventiveHealth.diagnosticTests.halfYearly.map((test, index) => (
649:                 <li key={index}>{test}</li>
650:             )}}
651:         </ul>
652:     </DataCard>
653:     <DataCard title="Yearly Tests">
654:         <ul className="list-disc list-inside">
655:             {preventiveHealth.diagnosticTests.yearly.map((test, index) => (

```

```

656:         <li key={index}>{test}</li>
657:     )})
658: </ul>
659: </DataCard>
660: </div>
661:
662: <SubSectionTitle title="Nutritional Supplements" icon="?" />
663: <DataCard title="Recommended Supplements">
664:     <ul className="list-disc list-inside">
665:         {preventiveHealth.nutritionalSupplements.map((supplement, index) => (
666:             <li key={index}>
667:                 {supplement.supplement}{" "}
668:                 {supplement.needed ? "(Needed)" : "(Optional)"}
669:             </li>
670:         ))}
671:     </ul>
672: </DataCard>
673: <Separator className="my-12" />
674:
675: {/* Family Genetic Impact */}
676: <SectionTitle title="Family Genetic Impact" icon="???????" />
677: <div className="space-y-8 mb-8">
678:     {familyGeneticImpact.length === 0 && (
679:         <p className="text-center text-gray-500 italic">
680:             No family genetic impacts recorded.
681:         </p>
682:     )}
683:     {familyGeneticImpact.map((impact, index) => (
684:         <Card
685:             key={index}
686:             className="border-2 border-blue-200 bg-blue-50/50 shadow-sm"
687:         >
688:             <CardHeader className="pb-2">
689:                 <CardTitle className="text-lg font-semibold text-blue-800">
690:                     Genetic Impact #{index + 1}
691:                 </CardTitle>
692:             </CardHeader>
693:             <CardContent className="text-sm text-gray-700 space-y-2">
694:                 <p>
695:                     <strong>Gene:</strong> {impact.gene}
696:                 </p>
697:                 <p>
698:                     <strong>Normal/Common Alleles:</strong> {impact.normalAlleles}
699:                 </p>
700:                 <p>
701:                     <strong>Your Result:</strong>{" "}
702:                     <span className="font-mono">{impact.yourResult}</span>
703:                 </p>
704:                 <p>
705:                     <strong>Health Impact:</strong> {impact.healthImpact}
706:                 </p>
707:             </CardContent>
708:         </Card>
709:     )})
710: </div>
711: <Separator className="my-12" />
712:
713: {/* Gene Test Results */}
714: <SectionTitle title="Gene Test Results" icon="?" />
715: <div className="space-y-8 mb-8">
716:     {geneTestResults.length === 0 && (
717:         <p className="text-center text-gray-500 italic">
718:             No gene test results recorded.
719:         </p>
720:     )}
721:     {geneTestResults.map((result, index) => (
722:         <Card
723:             key={index}
724:             className="border-2 border-indigo-200 bg-indigo-50/50 shadow-sm"
725:         >
726:             <CardHeader className="pb-2">
727:                 <CardTitle className="text-lg font-semibold text-indigo-800">
728:                     Gene: {result.geneName}

```

```

729:         </CardTitle>
730:     </CardHeader>
731:     <CardContent className="text-sm text-gray-700 space-y-2">
732:         <p>
733:             <strong>Variation:</strong>{ " " }
734:             <span className="font-mono">{result.variation}</span>
735:         </p>
736:         <p>
737:             <strong>Result:</strong>{ " " }
738:             <span className="font-mono">{result.result}</span>
739:         </p>
740:     </CardContent>
741: </Card>
742: ))}
743: </div>
744: <Separator className="my-12" />
745:
746: { /* Summaries */ }
747: <SectionTitle title="Summaries for Experts" icon="?" />
748: <div className="space-y-8 mb-8">
749:     <DataCard title="Nutrigenomics Summary">
750:         <p className="text-justify">{summaries.nutrigenomicsSummary}</p>
751:     </DataCard>
752:     <DataCard title="Exercise Genomics Summary">
753:         <p className="text-justify">{summaries.exerciseGenomicsSummary}</p>
754:     </DataCard>
755: </div>
756:
757: { /* Report Footer */ }
758: <footer className="text-center mt-12 py-6 text-gray-600 text-sm border-t border-gray-200">
759:     <p>
760:         &copy; {new Date().getFullYear()} {settings.companyName}. All rights
761:         reserved.
762:     </p>
763:     <p>Report Authentication: {patientInfo.reportAuth}</p>
764: </footer>
765: </div>
766: );
767: }

```

■ File: components/report-viewer.tsx

```

1: "use client";
2:
3: import { Card, CardContent, CardHeader, CardTitle } from "@/components/ui/card";
4: import { Badge } from "@/components/ui/badge";
5: import { Progress } from "@/components/ui/progress";
6: import type { ComprehensiveReportData } from "@/types/report-types";
7:
8: interface ReportViewerProps {
9:     data: ComprehensiveReportData;
10:    mode: "preview" | "pdf";
11: }
12:
13: export function ReportViewer({ data, mode }: ReportViewerProps) {
14:     const getScoreColor = (score: number) => {
15:         if (score >= data.settings.highThreshold) return data.settings.colors.high;
16:         if (score >= 4) return data.settings.colors.medium;
17:         return data.settings.colors.low;
18:     };
19:
20:     const getScoreLevel = (score: number) => {
21:         if (score >= data.settings.highThreshold) return "High";
22:         if (score >= 4) return "Medium";
23:         return "Low";
24:     };
25:
26:     return (
27:         <div className={`space-y-8 ${mode === "pdf" ? "print:space-y-4" : ""}`>
28:             { /* Page 1: Cover Page */ }
29:             <div className="bg-white p-8 rounded-lg shadow-sm page-break">

```

```

30:     <div
31:         className="text-center py-16 rounded-lg text-white"
32:         style={{ backgroundColor: data.settings.headerColor }}
33:     >
34:         <h1 className="text-4xl font-bold mb-4">{data.settings.title}</h1>
35:         <p className="text-xl italic mb-8">{data.settings.quote}</p>
36:         <div className="bg-white/20 p-6 rounded-lg inline-block">
37:             <h2 className="text-2xl font-semibold mb-2">
38:                 {data.patientInfo.name}
39:             </h2>
40:             <p className="text-lg">
41:                 Sample Code: {data.patientInfo.sampleCode}
42:             </p>
43:             <p className="text-lg">
44:                 Report Date: {data.patientInfo.reportDate}
45:             </p>
46:             <p className="text-gray-700 leading-relaxed whitespace-pre-line">
47:                 {data.patientInfo.disclaimer}
48:             </p>
49:         </div>
50:     </div>
51:
52:     { /* Page 3: Welcome Letter */ }
53:     <div className="bg-white p-8 rounded-lg shadow-sm page-break">
54:         <h2
55:             className="text-2xl font-bold mb-6"
56:             style={{ color: data.settings.headerColor }}
57:         >
58:             ? Welcome to Your Genomics Journey
59:         </h2>
60:         <div className="prose max-w-none">
61:             <p className="text-gray-700 leading-relaxed whitespace-pre-line mb-6">
62:                 {data.content.introduction}
63:             </p>
64:             <div className="bg-blue-50 p-6 rounded-lg">
65:                 <h3 className="text-lg font-semibold mb-3">What is Genomics?</h3>
66:                 <p className="text-gray-700 leading-relaxed whitespace-pre-line">
67:                     {data.content.genomicsExplanation}
68:                 </p>
69:             </div>
70:         </div>
71:     </div>
72:
73:     { /* Page 4-8: Diet Analysis */ }
74:     <div className="bg-white p-8 rounded-lg shadow-sm page-break">
75:         <h2
76:             className="text-2xl font-bold mb-6"
77:             style={{ color: data.settings.headerColor }}
78:         >
79:             ?? DIET ANALYSIS
80:         </h2>
81:
82:         { /* Macronutrients Section */ }
83:         <div className="mb-8">
84:             <h3 className="text-xl font-semibold mb-4">Macronutrients</h3>
85:             <div className="grid md:grid-cols-3 gap-4">
86:                 {Object.entries(data.dietAnalysis.macronutrients).map(
87:                     ([key, item]) => (
88:                         <Card key={key}>
89:                             <CardHeader className="pb-3">
90:                                 <CardTitle className="text-sm">
91:                                     {key
92:                                         .replace(/([A-Z])/g, " $1")
93:                                         .replace(/^./, (str) => str.toUpperCase())}
94:                                 </CardTitle>
95:                             </CardHeader>
96:                             <CardContent>
97:                                 <div className="flex items-center justify-between mb-2">
98:                                     <span className="text-2xl font-bold">
99:                                         {item.score}/10
100:                                     </span>
101:                                     <Badge
102:                                         style={{ backgroundColor: getScoreColor(item.score) }}

```



```

103:         className="text-white"
104:     >
105:         {getScoreLevel(item.score)}
106:     </Badge>
107: </div>
108:     <Progress value={item.score * 10} className="mb-3" />
109:     <p className="text-sm text-gray-600">
110:         {item.recommendation}
111:     </p>
112: </CardContent>
113: </Card>
114: )
115: }}
116: </div>
117: </div>
118:
119: { /* Food Sensitivities Section */}
120: <div className="mb-8">
121:     <h3 className="text-xl font-semibold mb-4">Food Sensitivities</h3>
122:     <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-4">
123:         {Object.entries(data.dietAnalysis.foodSensitivities).map(
124:             ([key, item]) => (
125:                 <Card key={key}>
126:                     <CardHeader className="pb-3">
127:                         <CardTitle className="text-sm">
128:                             {key
129:                                 .replace(/([A-Z])/g, " $1")
130:                                 .replace(/^./, (str) => str.toUpperCase())}
131:                         </CardTitle>
132:                     </CardHeader>
133:                     <CardContent>
134:                         <div className="flex items-center justify-between mb-2">
135:                             <span className="text-xl font-bold">{item.score}/10</span>
136:                             <Badge
137:                                 style={{ backgroundColor: getScoreColor(item.score) }}
138:                                 className="text-white"
139:                             >
140:                                 {item.level || getScoreLevel(item.score)}
141:                             </Badge>
142:                         </div>
143:                         <Progress value={item.score * 10} className="mb-3" />
144:                         <p className="text-xs text-gray-600">
145:                             {item.recommendation}
146:                         </p>
147:                     </CardContent>
148:                 </Card>
149:             )
150:         )}
151:     </div>
152: </div>
153: </div>
154:
155: { /* Page 9-15: Nutrition Data */}
156: <div className="bg-white p-8 rounded-lg shadow-sm page-break">
157:     <h2
158:         className="text-2xl font-bold mb-6"
159:         style={{ color: data.settings.headerColor }}
160:     >
161:         ? NUTRITION ANALYSIS
162:     </h2>
163:
164:     { /* Vitamins Section */}
165:     <div className="mb-8">
166:         <h3 className="text-xl font-semibold mb-4">Vitamins</h3>
167:         <div className="grid md:grid-cols-2 lg:grid-cols-4 gap-4">
168:             {Object.entries(data.nutritionData.vitamins).map(([key, item]) => (
169:                 <Card key={key}>
170:                     <CardHeader className="pb-3">
171:                         <CardTitle className="text-sm">
172:                             {key
173:                                 .replace(/([A-Z])/g, " $1")
174:                                 .replace(/^./, (str) => str.toUpperCase())}
175:                         </CardTitle>

```

```

176:         </CardHeader>
177:         <CardContent>
178:             <div className="flex items-center justify-between mb-2">
179:                 <span className="text-lg font-bold">{item.score}/10</span>
180:                 <Badge
181:                     style={{ backgroundColor: getScoreColor(item.score) }}
182:                     className="text-white text-xs"
183:                 >
184:                     {item.level || getScoreLevel(item.score)}
185:                 </Badge>
186:             </div>
187:             <Progress value={item.score * 10} className="mb-2" />
188:             <p className="text-xs text-gray-600 mb-2">
189:                 {item.healthImpact}
190:             </p>
191:             <p className="text-xs text-blue-600">{item.recommendation}</p>
192:         </CardContent>
193:     </Card>
194:   )}}
195: </div>
196: </div>
197:
198: { /* Elements Section */}
199: <div className="mb-8">
200:   <h3 className="text-xl font-semibold mb-4">Essential Elements</h3>
201:   <div className="grid md:grid-cols-2 lg:grid-cols-4 gap-4">
202:     {Object.entries(data.nutritionData.elements).map(([key, item]) => (
203:       <Card key={key}>
204:         <CardHeader className="pb-3">
205:           <CardTitle className="text-sm capitalize">{key}</CardTitle>
206:         </CardHeader>
207:         <CardContent>
208:           <div className="flex items-center justify-between mb-2">
209:             <span className="text-lg font-bold">{item.score}/10</span>
210:             <Badge
211:               style={{ backgroundColor: getScoreColor(item.score) }}
212:               className="text-white text-xs"
213:             >
214:               {item.level || getScoreLevel(item.score)}
215:             </Badge>
216:           </div>
217:           <Progress value={item.score * 10} className="mb-2" />
218:           <p className="text-xs text-gray-600 mb-2">
219:             {item.healthImpact}
220:           </p>
221:           <p className="text-xs text-blue-600">{item.recommendation}</p>
222:         </CardContent>
223:       </Card>
224:     )}}
225:   </div>
226: </div>
227: </div>
228:
229: { /* Page 16-22: Sports & Fitness */}
230: <div className="bg-white p-8 rounded-lg shadow-sm page-break">
231:   <h2
232:     className="text-2xl font-bold mb-6"
233:     style={{ color: data.settings.headerColor }}
234:   >
235:     ??? SPORTS & FITNESS
236:   </h2>
237:
238:   { /* Exercise Type Section */}
239:   <div className="mb-8">
240:     <h3 className="text-xl font-semibold mb-4">
241:       Exercise Type Suitability
242:     </h3>
243:     <div className="grid md:grid-cols-2 gap-4">
244:       {Object.entries(data.sportsAndFitness.exerciseType).map(
245:         ([key, item]) => (
246:           <Card key={key}>
247:             <CardHeader className="pb-3">
248:               <CardTitle className="text-sm">

```

```

249:         {key
250:             .replace(/[A-Z]/g, " $1")
251:             .replace(/^.\/, (str) => str.toUpperCase())}
252:         </CardTitle>
253:     </CardHeader>
254:     <CardContent>
255:         <div className="flex items-center justify-between mb-2">
256:             <span className="text-xl font-bold">{item.score}/10</span>
257:             <Badge
258:                 style={{ backgroundColor: getScoreColor(item.score) }}
259:                 className="text-white"
260:             >
261:                 {item.level || getScoreLevel(item.score)}
262:             </Badge>
263:         </div>
264:         <Progress value={item.score * 10} className="mb-3" />
265:         <p className="text-sm text-gray-600">{item.description}</p>
266:     </CardContent>
267: </Card>
268: )
269: }}
270: </div>
271: </div>
272:
273: { /* Performance Section */}
274: <div className="mb-8">
275:     <h3 className="text-xl font-semibold mb-4">Performance Factors</h3>
276:     <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-4">
277:         {Object.entries(data.sportsAndFitness.performance).map(
278:             ([key, item]) => (
279:                 <Card key={key}>
280:                     <CardHeader className="pb-3">
281:                         <CardTitle className="text-sm">
282:                             {key
283:                                 .replace(/[A-Z]/g, " $1")
284:                                 .replace(/^.\/, (str) => str.toUpperCase())}
285:                         </CardTitle>
286:                     </CardHeader>
287:                     <CardContent>
288:                         <div className="flex items-center justify-between mb-2">
289:                             <span className="text-lg font-bold">{item.score}/10</span>
290:                             <Badge
291:                                 style={{ backgroundColor: getScoreColor(item.score) }}
292:                                 className="text-white text-xs"
293:                             >
294:                                 {item.level || getScoreLevel(item.score)}
295:                             </Badge>
296:                         </div>
297:                         <Progress value={item.score * 10} className="mb-2" />
298:                         <p className="text-xs text-gray-600">{item.description}</p>
299:                     </CardContent>
300:                 </Card>
301:             )
302:         )}
303:     </div>
304: </div>
305:
306: { /* Injury Management Section */}
307: <div className="mb-8">
308:     <h3 className="text-xl font-semibold mb-4">Injury Risk Management</h3>
309:     <div className="grid md:grid-cols-2 lg:grid-cols-3 gap-4">
310:         {Object.entries(data.sportsAndFitness.injuryManagement).map(
311:             ([key, item]) => (
312:                 <Card key={key}>
313:                     <CardHeader className="pb-3">
314:                         <CardTitle className="text-sm">
315:                             {key
316:                                 .replace(/[A-Z]/g, " $1")
317:                                 .replace(/^.\/, (str) => str.toUpperCase())}
318:                         </CardTitle>
319:                     </CardHeader>
320:                     <CardContent>
321:                         <div className="flex items-center justify-between mb-2">

```

```

322:         <span className="text-lg font-bold">{item.score}/10</span>
323:         <Badge
324:           style={{ backgroundColor: getScoreColor(item.score) }}
325:           className="text-white text-xs"
326:         >
327:           {item.level || getScoreLevel(item.score)}
328:         </Badge>
329:       </div>
330:       <Progress value={item.score * 10} className="mb-2" />
331:       <p className="text-xs text-gray-600">{item.description}</p>
332:     </CardContent>
333:   </Card>
334: )
335: )}
336: </div>
337: </div>
338: </div>
339:
340: { /* Continue with remaining sections... */}
341: { /* This is a comprehensive start - I'll need to continue with all remaining sections to reach 56 pag
342:
343: <style jsx global>{`
344:   @media print {
345:     .page-break {
346:       page-break-after: always;
347:     }
348:     .no-print {
349:       display: none;
350:     }
351:   }
352: `}</style>
353: </div>
354: );
355: }

```

■ File: components\theme-provider.tsx

```

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

```

■ File: components\ui\accordion.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import * as AccordionPrimitive from "@radix-ui/react-accordion"
5: import { ChevronDown } from "lucide-react"
6:
7: import { cn } from "@lib/utils"
8:
9: const Accordion = AccordionPrimitive.Root
10:
11: const AccordionItem = React.forwardRef<
12:   React.ElementRef<typeof AccordionPrimitive.Item>,
13:   React.ComponentPropsWithoutRef<typeof AccordionPrimitive.Item>
14: >(({ className, ...props }, ref) => (
15:   <AccordionPrimitive.Item
16:     ref={ref}
17:     className={cn("border-b", className)}
18:     {...props}
19:   />
20: ))
21: AccordionItem.displayName = "AccordionItem"
22:

```

```

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

```

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

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import * as AlertDialogPrimitive from "@radix-ui/react-alert-dialog"
5:
6: import { cn } from "@lib/utils"
7: import { buttonVariants } from "@components/ui/button"
8:
9: const AlertDialog = AlertDialogPrimitive.Root
10:
11: const AlertDialogTrigger = AlertDialogPrimitive.Trigger
12:
13: const AlertDialogPortal = AlertDialogPrimitive.Portal
14:
15: const AlertDialogOverlay = React.forwardRef<
16:   React.ElementRef<typeof AlertDialogPrimitive.Overlay>,
17:   React.ComponentPropsWithoutRef<typeof AlertDialogPrimitive.Overlay>
18: >(({ className, ...props }, ref) => (
19:   <AlertDialogPrimitive.Overlay
20:     className={cn(
21:       "fixed inset-0 z-50 bg-black/80 data-[state=open]:animate-in data-[state=closed]:animate-out data-[s
22:     className
23:   )}
24:   {...props}
25:   ref={ref}
26: />
27: ))
28: AlertDialogOverlay.displayName = AlertDialogPrimitive.Overlay.displayName
29:
30: const AlertDialogContent = React.forwardRef<
31:   React.ElementRef<typeof AlertDialogPrimitive.Content>,
32:   React.ComponentPropsWithoutRef<typeof AlertDialogPrimitive.Content>

```

```

33: >(({ className, ...props }, ref) => (
34:   <AlertDialogPortal>
35:     <AlertDialogOverlay />
36:     <AlertDialogPrimitive.Content
37:       ref={ref}
38:       className={cn(
39:         "fixed left-[50%] top-[50%] z-50 grid w-full max-w-lg translate-x-[-50%] translate-y-[-50%] gap-4 b
40:         className
41:       )}
42:       {...props}
43:     />
44:   </AlertDialogPortal>
45: ))
46: AlertDialogContent.displayName = AlertDialogPrimitive.Content.displayName
47:
48: const AlertDialogHeader = ({
49:   className,
50:   ...props
51: }: React.HTMLAttributes<HTMLDivElement>) => (
52:   <div
53:     className={cn(
54:       "flex flex-col space-y-2 text-center sm:text-left",
55:       className
56:     )}
57:     {...props}
58:   />
59: )
60: AlertDialogHeader.displayName = "AlertDialogHeader"
61:
62: const AlertDialogFooter = ({
63:   className,
64:   ...props
65: }: React.HTMLAttributes<HTMLDivElement>) => (
66:   <div
67:     className={cn(
68:       "flex flex-col-reverse sm:flex-row sm:justify-end sm:space-x-2",
69:       className
70:     )}
71:     {...props}
72:   />
73: )
74: AlertDialogFooter.displayName = "AlertDialogFooter"
75:
76: const AlertDialogTitle = React.forwardRef<
77:   React.ElementRef<typeof AlertDialogPrimitive.Title>,
78:   React.ComponentPropsWithoutRef<typeof AlertDialogPrimitive.Title>
79: >(({ className, ...props }, ref) => (
80:   <AlertDialogPrimitive.Title
81:     ref={ref}
82:     className={cn("text-lg font-semibold", className)}
83:     {...props}
84:   />
85: ))
86: AlertDialogTitle.displayName = AlertDialogPrimitive.Title.displayName
87:
88: const AlertDialogDescription = React.forwardRef<
89:   React.ElementRef<typeof AlertDialogPrimitive.Description>,
90:   React.ComponentPropsWithoutRef<typeof AlertDialogPrimitive.Description>
91: >(({ className, ...props }, ref) => (
92:   <AlertDialogPrimitive.Description
93:     ref={ref}
94:     className={cn("text-sm text-muted-foreground", className)}
95:     {...props}
96:   />
97: ))
98: AlertDialogDescription.displayName =
99:   AlertDialogPrimitive.Description.displayName
100:
101: const AlertDialogAction = React.forwardRef<
102:   React.ElementRef<typeof AlertDialogPrimitive.Action>,
103:   React.ComponentPropsWithoutRef<typeof AlertDialogPrimitive.Action>
104: >(({ className, ...props }, ref) => (
105:   <AlertDialogPrimitive.Action

```

```

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

```

■ File: components/ui/alert.tsx

```

=====
1: import * as React from "react"
2: import { cva, type VariantProps } from "class-variance-authority"
3:
4: import { cn } from "@/lib/utils"
5:
6: const alertVariants = cva(
7:   "relative w-full rounded-lg border p-4 [&svg~*]:pl-7 [&svg+div]:translate-y-[-3px] [&svg]:absolute [&
8:   {
9:     variants: {
10:       variant: {
11:         default: "bg-background text-foreground",
12:         destructive:
13:           "border-destructive/50 text-destructive dark:border-destructive [&svg]:text-destructive",
14:       },
15:     },
16:     defaultVariants: {
17:       variant: "default",
18:     },
19:   }
20: )
21:
22: const Alert = React.forwardRef<
23:   HTMLDivElement,
24:   React.HTMLAttributes<HTMLDivElement> & VariantProps<typeof alertVariants>
25: >(({ className, variant, ...props }, ref) => (
26:   <div
27:     ref={ref}
28:     role="alert"
29:     className={cn(alertVariants({ variant }), className)}
30:     {...props}
31:   />
32: ))

```

```

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

```

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

```

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

```

■ File: components/ui/avatar.tsx

```

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

```



```

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

```

■ File: components\ui\badge.tsx

```

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

```

■ File: components\ui\breadcrumb.tsx

```

1: import * as React from "react"
2: import { Slot } from "@radix-ui/react-slot"
3: import { ChevronRight, MoreHorizontal } from "lucide-react"
4:
5: import { cn } from "@/lib/utils"
6:

```

```

7: const Breadcrumb = React.forwardRef<
8:   HTMLElement,
9:   React.ComponentPropsWithoutRef<"nav"> & {
10:     separator?: React.ReactNode
11:   }
12: >(({ ...props }, ref) => <nav ref={ref} aria-label="breadcrumb" {...props} />)
13: Breadcrumb.displayName = "Breadcrumb"
14:
15: const BreadcrumbList = React.forwardRef<
16:   HTMLListElement,
17:   React.ComponentPropsWithoutRef<"ol">
18: >(({ className, ...props }, ref) => (
19:   <ol
20:     ref={ref}
21:     className={cn(
22:       "flex flex-wrap items-center gap-1.5 break-words text-sm text-muted-foreground sm:gap-2.5",
23:       className
24:     )}
25:     {...props}
26:   />
27: ))
28: BreadcrumbList.displayName = "BreadcrumbList"
29:
30: const BreadcrumbItem = React.forwardRef<
31:   HTMLLIElement,
32:   React.ComponentPropsWithoutRef<"li">
33: >(({ className, ...props }, ref) => (
34:   <li
35:     ref={ref}
36:     className={cn("inline-flex items-center gap-1.5", className)}
37:     {...props}
38:   />
39: ))
40: BreadcrumbItem.displayName = "BreadcrumbItem"
41:
42: const BreadcrumbLink = React.forwardRef<
43:   HTMLAnchorElement,
44:   React.ComponentPropsWithoutRef<"a"> & {
45:     asChild?: boolean
46:   }
47: >(({ asChild, className, ...props }, ref) => {
48:   const Comp = asChild ? Slot : "a"
49:
50:   return (
51:     <Comp
52:       ref={ref}
53:       className={cn("transition-colors hover:text-foreground", className)}
54:       {...props}
55:     />
56:   )
57: })
58: BreadcrumbLink.displayName = "BreadcrumbLink"
59:
60: const BreadcrumbPage = React.forwardRef<
61:   HTMLSpanElement,
62:   React.ComponentPropsWithoutRef<"span">
63: >(({ className, ...props }, ref) => (
64:   <span
65:     ref={ref}
66:     role="link"
67:     aria-disabled="true"
68:     aria-current="page"
69:     className={cn("font-normal text-foreground", className)}
70:     {...props}
71:   />
72: ))
73: BreadcrumbPage.displayName = "BreadcrumbPage"
74:
75: const BreadcrumbSeparator = ({
76:   children,
77:   className,
78:   ...props
79: }: React.ComponentProps<"li">) => (

```

```

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

```

■ File: components/ui/button.tsx

```

=====
1: import * as React from "react"
2: import { Slot } from "@radix-ui/react-slot"
3: import { cva, type VariantProps } from "class-variance-authority"
4:
5: import { cn } from "@lib/utils"
6:
7: const buttonVariants = cva(
8:   "inline-flex items-center justify-center gap-2 whitespace-nowrap rounded-md text-sm font-medium ring-off
9:   {
10:     variants: {
11:       variant: {
12:         default: "bg-primary text-primary-foreground hover:bg-primary/90",
13:         destructive:
14:           "bg-destructive text-destructive-foreground hover:bg-destructive/90",
15:         outline:
16:           "border border-input bg-background hover:bg-accent hover:text-accent-foreground",
17:         secondary:
18:           "bg-secondary text-secondary-foreground hover:bg-secondary/80",
19:         ghost: "hover:bg-accent hover:text-accent-foreground",
20:         link: "text-primary underline-offset-4 hover:underline",
21:       },
22:       size: {
23:         default: "h-10 px-4 py-2",
24:         sm: "h-9 rounded-md px-3",
25:         lg: "h-11 rounded-md px-8",
26:         icon: "h-10 w-10",
27:       },
28:     },
29:     defaultVariants: {
30:       variant: "default",
31:       size: "default",
32:     },

```

```

33:   }
34: )
35:
36: export interface ButtonProps
37:   extends React.ButtonHTMLAttributes<HTMLButtonElement>,
38:     VariantProps<typeof buttonVariants> {
39:   asChild?: boolean
40: }
41:
42: const Button = React.forwardRef<HTMLButtonElement, ButtonProps>(
43:   ({ className, variant, size, asChild = false, ...props }, ref) => {
44:     const Comp = asChild ? Slot : "button"
45:     return (
46:       <Comp
47:         className={cn(buttonVariants({ variant, size, className }))}
48:         ref={ref}
49:         {...props}
50:       />
51:     )
52:   }
53: )
54: Button.displayName = "Button"
55:
56: 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-prima
47:         day_today: "bg-accent text-accent-foreground",
48:         day_outside:
49:             "day-outside text-muted-foreground aria-selected:bg-accent/50 aria-selected:text-muted-foreground
50:         day_disabled: "text-muted-foreground opacity-50",
51:         day_range_middle:
52:             "aria-selected:bg-accent aria-selected:text-accent-foreground",
53:         day_hidden: "invisible",
54:         ...classNames,
55:     }}
56:     components={{
57:         IconLeft: ({ ...props }) => <ChevronLeft className="h-4 w-4" />,
58:         IconRight: ({ ...props }) => <ChevronRight className="h-4 w-4" />,
59:     }}
60:     {...props}
61: />
62: )
63: }
64: Calendar.displayName = "Calendar"
65:
66: export { Calendar }

```

■ File: components\ui\card.tsx

```

1: import * as React from "react"
2:
3: import { cn } from "@/lib/utils"
4:
5: const Card = React.forwardRef<
6:   HTMLDivElement,
7:   React.HTMLAttributes<HTMLDivElement>
8: >(({ className, ...props }, ref) => (
9:   <div
10:     ref={ref}
11:     className={cn(
12:       "rounded-lg border bg-card text-card-foreground shadow-sm",
13:       className
14:     )}
15:     {...props}
16:   />
17: ))
18: Card.displayName = "Card"
19:
20: const CardHeader = React.forwardRef<
21:   HTMLDivElement,
22:   React.HTMLAttributes<HTMLDivElement>
23: >(({ className, ...props }, ref) => (
24:   <div
25:     ref={ref}
26:     className={cn("flex flex-col space-y-1.5 p-6", className)}
27:     {...props}
28:   />
29: ))
30: CardHeader.displayName = "CardHeader"
31:
32: const CardTitle = React.forwardRef<
33:   HTMLDivElement,
34:   React.HTMLAttributes<HTMLDivElement>
35: >(({ className, ...props }, ref) => (
36:   <div
37:     ref={ref}
38:     className={cn(
39:       "text-2xl font-semibold leading-none tracking-tight",
40:       className
41:     )}
42:     {...props}
43:   />
44: ))
45: CardTitle.displayName = "CardTitle"
46:

```

```

47: const CardDescription = React.forwardRef<
48:   HTMLDivElement,
49:   React.HTMLAttributes<HTMLDivElement>
50: >(({ className, ...props }, ref) => (
51:   <div
52:     ref={ref}
53:     className={cn("text-sm text-muted-foreground", className)}
54:     {...props}
55:   />
56: ))
57: CardDescription.displayName = "CardDescription"
58:
59: const CardContent = React.forwardRef<
60:   HTMLDivElement,
61:   React.HTMLAttributes<HTMLDivElement>
62: >(({ className, ...props }, ref) => (
63:   <div ref={ref} className={cn("p-6 pt-0", className)} {...props} />
64: ))
65: CardContent.displayName = "CardContent"
66:
67: const CardFooter = React.forwardRef<
68:   HTMLDivElement,
69:   React.HTMLAttributes<HTMLDivElement>
70: >(({ className, ...props }, ref) => (
71:   <div
72:     ref={ref}
73:     className={cn("flex items-center p-6 pt-0", className)}
74:     {...props}
75:   />
76: ))
77: CardFooter.displayName = "CardFooter"
78:
79: export { Card, CardHeader, CardFooter, CardTitle, CardDescription, CardContent }

```

■ File: components/ui/carousel.tsx

```

1: "use client"
2:
3: import * as React from "react"
4: import useEmblaCarousel, {
5:   type UseEmblaCarouselType,
6: } from "embla-carousel-react"
7: import { ArrowLeft, ArrowRight } from "lucide-react"
8:
9: import { cn } from "@/lib/utils"
10: import { Button } from "@/components/ui/button"
11:
12: type CarouselApi = UseEmblaCarouselType[1]
13: type UseCarouselParameters = Parameters<typeof useEmblaCarousel>
14: type CarouselOptions = UseCarouselParameters[0]
15: type CarouselPlugin = UseCarouselParameters[1]
16:
17: type CarouselProps = {
18:   opts?: CarouselOptions
19:   plugins?: CarouselPlugin
20:   orientation?: "horizontal" | "vertical"
21:   setApi?: (api: CarouselApi) => void
22: }
23:
24: type CarouselContextProps = {
25:   carouselRef: ReturnType<typeof useEmblaCarousel>[0]
26:   api: ReturnType<typeof useEmblaCarousel>[1]
27:   scrollPrev: () => void
28:   scrollNext: () => void
29:   canScrollPrev: boolean
30:   canScrollNext: boolean
31: } & CarouselProps
32:
33: const CarouselContext = React.createContext<CarouselContextProps | null>(null)
34:
35: function useCarousel() {

```

```

36:   const context = React.useContext(CarouselContext)
37:
38:   if (!context) {
39:     throw new Error("useCarousel must be used within a <Carousel />")
40:   }
41:
42:   return context
43: }
44:
45: const Carousel = React.forwardRef<
46:   HTMLDivElement,
47:   React.HTMLAttributes<HTMLDivElement> & CarouselProps
48: > (
49:   (
50:     {
51:       orientation = "horizontal",
52:       opts,
53:       setApi,
54:       plugins,
55:       className,
56:       children,
57:       ...props
58:     },
59:     ref
60:   ) => {
61:     const [carouselRef, api] = useEmblaCarousel(
62:       {
63:         ...opts,
64:         axis: orientation === "horizontal" ? "x" : "y",
65:       },
66:       plugins
67:     )
68:     const [canScrollPrev, setCanScrollPrev] = React.useState(false)
69:     const [canScrollNext, setCanScrollNext] = React.useState(false)
70:
71:     const onSelect = React.useCallback((api: CarouselApi) => {
72:       if (!api) {
73:         return
74:       }
75:
76:       setCanScrollPrev(api.canScrollPrev())
77:       setCanScrollNext(api.canScrollNext())
78:     }, [])
79:
80:     const scrollPrev = React.useCallback(() => {
81:       api?.scrollPrev()
82:     }, [api])
83:
84:     const scrollNext = React.useCallback(() => {
85:       api?.scrollNext()
86:     }, [api])
87:
88:     const handleKeyDown = React.useCallback(
89:       (event: React.KeyboardEvent<HTMLDivElement>) => {
90:         if (event.key === "ArrowLeft") {
91:           event.preventDefault()
92:           scrollPrev()
93:         } else if (event.key === "ArrowRight") {
94:           event.preventDefault()
95:           scrollNext()
96:         }
97:       },
98:       [scrollPrev, scrollNext]
99:     )
100:
101:     React.useEffect(() => {
102:       if (!api || !setApi) {
103:         return
104:       }
105:
106:       setApi(api)
107:     }, [api, setApi])
108:

```

```

109:     React.useEffect(() => {
110:         if (!api) {
111:             return
112:         }
113:
114:         onSelect(api)
115:         api.on("reInit", onSelect)
116:         api.on("select", onSelect)
117:
118:         return () => {
119:             api?.off("select", onSelect)
120:         }
121:     }, [api, onSelect])
122:
123:     return (
124:         <CarouselContext.Provider
125:             value={{
126:                 carouselRef,
127:                 api: api,
128:                 opts,
129:                 orientation:
130:                     orientation || (opts?.axis === "y" ? "vertical" : "horizontal"),
131:                 scrollPrev,
132:                 scrollNext,
133:                 canScrollPrev,
134:                 canScrollNext,
135:             }}
136:         >
137:             <div
138:                 ref={ref}
139:                 onKeyDownCapture={handleKeyDown}
140:                 className={cn("relative", className)}
141:                 role="region"
142:                 aria-roledescription="carousel"
143:                 {...props}
144:             >
145:                 {children}
146:             </div>
147:         </CarouselContext.Provider>
148:     )
149: }
150: )
151: Carousel.displayName = "Carousel"
152:
153: const CarouselContent = React.forwardRef<
154:     HTMLDivElement,
155:     React.HTMLAttributes<HTMLDivElement>
156: >(({ className, ...props }, ref) => {
157:     const { carouselRef, orientation } = useCarousel()
158:
159:     return (
160:         <div ref={carouselRef} className="overflow-hidden">
161:             <div
162:                 ref={ref}
163:                 className={cn(
164:                     "flex",
165:                     orientation === "horizontal" ? "-ml-4" : "-mt-4 flex-col",
166:                     className
167:                 )}
168:                 {...props}
169:             />
170:         </div>
171:     )
172: })
173: CarouselContent.displayName = "CarouselContent"
174:
175: const CarouselItem = React.forwardRef<
176:     HTMLDivElement,
177:     React.HTMLAttributes<HTMLDivElement>
178: >(({ className, ...props }, ref) => {
179:     const { orientation } = useCarousel()
180:
181:     return (

```



```

182:     <div
183:       ref={ref}
184:       role="group"
185:       aria-roledescription="slide"
186:       className={cn(
187:         "min-w-0 shrink-0 grow-0 basis-full",
188:         orientation === "horizontal" ? "pl-4" : "pt-4",
189:         className
190:       )}
191:       {...props}
192:     />
193:   )
194: })
195: CarouselItem.displayName = "CarouselItem"
196:
197: const CarouselPrevious = React.forwardRef<
198:   HTMLButtonElement,
199:   React.ComponentProps<typeof Button>
200: >(({ className, variant = "outline", size = "icon", ...props }, ref) => {
201:   const { orientation, scrollPrev, canScrollPrev } = useCarousel()
202:
203:   return (
204:     <Button
205:       ref={ref}
206:       variant={variant}
207:       size={size}
208:       className={cn(
209:         "absolute h-8 w-8 rounded-full",
210:         orientation === "horizontal"
211:           ? "-left-12 top-1/2 -translate-y-1/2"
212:           : "-top-12 left-1/2 -translate-x-1/2 rotate-90",
213:         className
214:       )}
215:       disabled={!canScrollPrev}
216:       onClick={scrollPrev}
217:       {...props}
218:     >
219:       <ArrowLeft className="h-4 w-4" />
220:       <span className="sr-only">Previous slide</span>
221:     </Button>
222:   )
223: })
224: CarouselPrevious.displayName = "CarouselPrevious"
225:
226: const CarouselNext = React.forwardRef<
227:   HTMLButtonElement,
228:   React.ComponentProps<typeof Button>
229: >(({ className, variant = "outline", size = "icon", ...props }, ref) => {
230:   const { orientation, scrollNext, canScrollNext } = useCarousel()
231:
232:   return (
233:     <Button
234:       ref={ref}
235:       variant={variant}
236:       size={size}
237:       className={cn(
238:         "absolute h-8 w-8 rounded-full",
239:         orientation === "horizontal"
240:           ? "-right-12 top-1/2 -translate-y-1/2"
241:           : "-bottom-12 left-1/2 -translate-x-1/2 rotate-90",
242:         className
243:       )}
244:       disabled={!canScrollNext}
245:       onClick={scrollNext}
246:       {...props}
247:     >
248:       <ArrowRight className="h-4 w-4" />
249:       <span className="sr-only">Next slide</span>
250:     </Button>
251:   )
252: })
253: CarouselNext.displayName = "CarouselNext"
254:

```

```

255: export {
256:   type CarouselApi,
257:   Carousel,
258:   CarouselContent,
259:   CarouselItem,
260:   CarouselPrevious,
261:   CarouselNext,
262: }

```

■ File: components\ui\chart.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import * as RechartsPrimitive from "recharts"
5:
6: import { cn } from "@/lib/utils"
7:
8: // Format: { THEME_NAME: CSS_SELECTOR }
9: const THEMES = { light: "", dark: ".dark" } as const
10:
11: export type ChartConfig = {
12:   [k in string]: {
13:     label?: React.ReactNode
14:     icon?: React.ComponentType
15:   } & (
16:     | { color?: string; theme?: never }
17:     | { color?: never; theme: Record<keyof typeof THEMES, string> }
18:   )
19: }
20:
21: type ChartContextProps = {
22:   config: ChartConfig
23: }
24:
25: const ChartContext = React.createContext<ChartContextProps | null>(null)
26:
27: function useChart() {
28:   const context = React.useContext(ChartContext)
29:
30:   if (!context) {
31:     throw new Error("useChart must be used within a <ChartContainer />")
32:   }
33:
34:   return context
35: }
36:
37: const ChartContainer = React.forwardRef<
38:   HTMLDivElement,
39:   React.ComponentProps<"div"> & {
40:     config: ChartConfig
41:     children: React.ComponentProps<
42:       typeof RechartsPrimitive.ResponsiveContainer
43:     >["children"]
44:   }
45: >(({ id, className, children, config, ...props }, ref) => {
46:   const uniqueId = React.useId()
47:   const chartId = `chart-${id || uniqueId.replace(/:/g, "")}`
48:
49:   return (
50:     <ChartContext.Provider value={{ config }}>
51:       <div
52:         data-chart={chartId}
53:         ref={ref}
54:         className={cn(
55:           "flex aspect-video justify-center text-xs [&_.recharts-cartesian-axis-tick_text]:fill-muted-foreground",
56:           className
57:         )}
58:         {...props}
59:       >
60:         <ChartStyle id={chartId} config={config} />

```

```

61:         <RechartsPrimitive.ResponsiveContainer>
62:             {children}
63:         </RechartsPrimitive.ResponsiveContainer>
64:     </div>
65: </ChartContext.Provider>
66: )
67: })
68: ChartContainer.displayName = "Chart"
69:
70: const ChartStyle = ({ id, config }: { id: string; config: ChartConfig }) => {
71:     const colorConfig = Object.entries(config).filter(
72:         ([_, config]) => config.theme || config.color
73:     )
74:
75:     if (!colorConfig.length) {
76:         return null
77:     }
78:
79:     return (
80:         <style
81:             dangerouslySetInnerHTML={{
82:                 __html: Object.entries(THEMES)
83:                     .map(
84:                         ([theme, prefix]) => `
85: ${prefix} [data-chart=${id}] {
86: ${colorConfig
87:     .map(([key, itemConfig]) => {
88:         const color =
89:             itemConfig.theme?.[theme as keyof typeof itemConfig.theme] ||
90:             itemConfig.color
91:         return color ? `  --color-${key}: ${color};` : null
92:     })
93:     .join("\n")}
94: }
95: `
96:         )
97:         .join("\n"),
98:         }}
99:     />
100: )
101: }
102:
103: const ChartTooltip = RechartsPrimitive.Tooltip
104:
105: const ChartTooltipContent = React.forwardRef<
106:     HTMLDivElement,
107:     React.ComponentProps<typeof RechartsPrimitive.Tooltip> &
108:     React.ComponentProps<"div"> & {
109:         hideLabel?: boolean
110:         hideIndicator?: boolean
111:         indicator?: "line" | "dot" | "dashed"
112:         nameKey?: string
113:         labelKey?: string
114:     }
115: >((
116:     (
117:         {
118:             active,
119:             payload,
120:             className,
121:             indicator = "dot",
122:             hideLabel = false,
123:             hideIndicator = false,
124:             label,
125:             labelFormatter,
126:             labelClassName,
127:             formatter,
128:             color,
129:             nameKey,
130:             labelKey,
131:         },
132:         ref
133:     ) => {

```

```

134:     const { config } = useChart()
135:
136:     const tooltipLabel = React.useMemo(() => {
137:       if (hideLabel || !payload?.length) {
138:         return null
139:       }
140:
141:       const [item] = payload
142:       const key = `${labelKey || item.dataKey || item.name || "value"}`
143:       const itemConfig = getPayloadConfigFromPayload(config, item, key)
144:       const value =
145:         !labelKey && typeof label === "string"
146:         ? config[label as keyof typeof config]?.label || label
147:         : itemConfig?.label
148:
149:       if (labelFormatter) {
150:         return (
151:           <div className={cn("font-medium", labelClassName)}>
152:             {labelFormatter(value, payload)}
153:           </div>
154:         )
155:       }
156:
157:       if (!value) {
158:         return null
159:       }
160:
161:       return <div className={cn("font-medium", labelClassName)}>{value}</div>
162:     }, [
163:       label,
164:       labelFormatter,
165:       payload,
166:       hideLabel,
167:       labelClassName,
168:       config,
169:       labelKey,
170:     ])
171:
172:     if (!active || !payload?.length) {
173:       return null
174:     }
175:
176:     const nestLabel = payload.length === 1 && indicator !== "dot"
177:
178:     return (
179:       <div
180:         ref={ref}
181:         className={cn(
182:           "grid min-w-[8rem] items-start gap-1.5 rounded-lg border border-border/50 bg-background px-2.5 py-2",
183:           className
184:         )}
185:       >
186:         {!nestLabel ? tooltipLabel : null}
187:         <div className="grid gap-1.5">
188:           {payload.map((item, index) => {
189:             const key = `${nameKey || item.name || item.dataKey || "value"}`
190:             const itemConfig = getPayloadConfigFromPayload(config, item, key)
191:             const indicatorColor = color || item.payload.fill || item.color
192:
193:             return (
194:               <div
195:                 key={item.dataKey}
196:                 className={cn(
197:                   "flex w-full flex-wrap items-stretch gap-2 [>svg]:h-2.5 [>svg]:w-2.5 [>svg]:text-muted-foreground",
198:                   indicator === "dot" && "items-center"
199:                 )}
200:               >
201:                 {formatter && item?.value !== undefined && item.name ? (
202:                   formatter(item.value, item.name, item, index, item.payload)
203:                 ) : (
204:                   <>
205:                     {itemConfig?.icon ? (
206:                       <itemConfig.icon />

```

```

207:         ) : (
208:             !hideIndicator && (
209:                 <div
210:                     className={cn(
211:                         "shrink-0 rounded-[2px] border-[--color-border] bg-[--color-bg]",
212:                         {
213:                             "h-2.5 w-2.5": indicator === "dot",
214:                             "w-1": indicator === "line",
215:                             "w-0 border-[1.5px] border-dashed bg-transparent":
216:                                 indicator === "dashed",
217:                             "my-0.5": nestLabel && indicator === "dashed",
218:                         }
219:                     )}
220:                     style={
221:                         {
222:                             "--color-bg": indicatorColor,
223:                             "--color-border": indicatorColor,
224:                         } as React.CSSProperties
225:                     }
226:                 />
227:             )
228:         )}
229:         <div
230:             className={cn(
231:                 "flex flex-1 justify-between leading-none",
232:                 nestLabel ? "items-end" : "items-center"
233:             )}
234:         >
235:             <div className="grid gap-1.5">
236:                 {nestLabel ? tooltipLabel : null}
237:                 <span className="text-muted-foreground">
238:                     {itemConfig?.label || item.name}
239:                 </span>
240:             </div>
241:             {item.value && (
242:                 <span className="font-mono font-medium tabular-nums text-foreground">
243:                     {item.value.toLocaleString()}
244:                 </span>
245:             )}
246:         </div>
247:     </>
248:     )}
249: </div>
250: )
251: )}
252: </div>
253: </div>
254: )
255: }
256: )
257: ChartTooltipContent.displayName = "ChartTooltip"
258:
259: const ChartLegend = RechartsPrimitive.Legend
260:
261: const ChartLegendContent = React.forwardRef<
262:     HTMLDivElement,
263:     React.ComponentProps<"div"> &
264:         Pick<RechartsPrimitive.LegendProps, "payload" | "verticalAlign"> & {
265:             hideIcon?: boolean
266:             nameKey?: string
267:         }
268: >((
269:     (
270:         { className, hideIcon = false, payload, verticalAlign = "bottom", nameKey },
271:         ref
272:     ) => {
273:         const { config } = useChart()
274:
275:         if (!payload?.length) {
276:             return null
277:         }
278:
279:         return (

```

```

280:     <div
281:       ref={ref}
282:       className={cn(
283:         "flex items-center justify-center gap-4",
284:         verticalAlign === "top" ? "pb-3" : "pt-3",
285:         className
286:       )}
287:     >
288:       {payload.map((item) => {
289:         const key = `${nameKey} || item.dataKey || "value"}`
290:         const itemConfig = getPayloadConfigFromPayload(config, item, key)
291:
292:         return (
293:           <div
294:             key={item.value}
295:             className={cn(
296:               "flex items-center gap-1.5 [&svg]:h-3 [&svg]:w-3 [&svg]:text-muted-foreground"
297:             )}
298:           >
299:             {itemConfig?.icon && !hideIcon ? (
300:               <itemConfig.icon />
301:             ) : (
302:               <div
303:                 className="h-2 w-2 shrink-0 rounded-[2px]"
304:                 style={{
305:                   backgroundColor: item.color,
306:                 }}
307:               />
308:             )}
309:             {itemConfig?.label}
310:           </div>
311:         )
312:       )}
313:     </div>
314:   )
315: }
316: )
317: ChartLegendContent.displayName = "ChartLegend"
318:
319: // Helper to extract item config from a payload.
320: function getPayloadConfigFromPayload(
321:   config: ChartConfig,
322:   payload: unknown,
323:   key: string
324: ) {
325:   if (typeof payload !== "object" || payload === null) {
326:     return undefined
327:   }
328:
329:   const payloadPayload =
330:     "payload" in payload &&
331:     typeof payload.payload === "object" &&
332:     payload.payload !== null
333:       ? payload.payload
334:       : undefined
335:
336:   let configLabelKey: string = key
337:
338:   if (
339:     key in payload &&
340:     typeof payload[key as keyof typeof payload] === "string"
341:   ) {
342:     configLabelKey = payload[key as keyof typeof payload] as string
343:   } else if (
344:     payloadPayload &&
345:     key in payloadPayload &&
346:     typeof payloadPayload[key as keyof typeof payloadPayload] === "string"
347:   ) {
348:     configLabelKey = payloadPayload[
349:       key as keyof typeof payloadPayload
350:     ] as string
351:   }
352:

```

```

353:   return configLabelKey in config
354:     ? config[configLabelKey]
355:     : config[key as keyof typeof config]
356: }
357:
358: export {
359:   ChartContainer,
360:   ChartTooltip,
361:   ChartTooltipContent,
362:   ChartLegend,
363:   ChartLegendContent,
364:   ChartStyle,
365: }

```

■ File: components/ui/checkbox.tsx

```

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

```

■ File: components/ui/collapsible.tsx

```

=====
1: "use client"
2:
3: import * as CollapsiblePrimitive from "@radix-ui/react-collapsible"
4:
5: const Collapsible = CollapsiblePrimitive.Root
6:
7: const CollapsibleTrigger = CollapsiblePrimitive.CollapsibleTrigger
8:
9: const CollapsibleContent = CollapsiblePrimitive.CollapsibleContent
10:
11: export { Collapsible, CollapsibleTrigger, CollapsibleContent }

```

■ File: components/ui/command.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import { type DialogProps } from "@radix-ui/react-dialog"

```

```

5: import { Command as CommandPrimitive } from "cmdk"
6: import { Search } from "lucide-react"
7:
8: import { cn } from "@lib/utils"
9: import { Dialog, DialogContent } from "@components/ui/dialog"
10:
11: const Command = React.forwardRef<
12:   React.ElementRef<typeof CommandPrimitive>,
13:   React.ComponentPropsWithoutRef<typeof CommandPrimitive>
14: >(({ className, ...props }, ref) => (
15:   <CommandPrimitive
16:     ref={ref}
17:     className={cn(
18:       "flex h-full w-full flex-col overflow-hidden rounded-md bg-popover text-popover-foreground",
19:       className
20:     )}
21:     {...props}
22:   />
23: ))
24: Command.displayName = CommandPrimitive.displayName
25:
26: const CommandDialog = ({ children, ...props }: DialogProps) => {
27:   return (
28:     <Dialog {...props}>
29:       <DialogContent className="overflow-hidden p-0 shadow-lg">
30:         <Command className="[&_[:cmdk-group-heading]]:px-2 [&_[:cmdk-group-heading]]:font-medium [&_[:cmdk-group-
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-foregr
48:         className
49:       )}
50:       {...props}
51:     />
52:   </div>
53: ))
54:
55: CommandInput.displayName = CommandPrimitive.Input.displayName
56:
57: const CommandList = React.forwardRef<
58:   React.ElementRef<typeof CommandPrimitive.List>,
59:   React.ComponentPropsWithoutRef<typeof CommandPrimitive.List>
60: >(({ className, ...props }, ref) => (
61:   <CommandPrimitive.List
62:     ref={ref}
63:     className={cn("max-h-[300px] overflow-y-auto overflow-x-hidden", className)}
64:     {...props}
65:   />
66: ))
67:
68: CommandList.displayName = CommandPrimitive.List.displayName
69:
70: const CommandEmpty = React.forwardRef<
71:   React.ElementRef<typeof CommandPrimitive.Empty>,
72:   React.ComponentPropsWithoutRef<typeof CommandPrimitive.Empty>
73: >((props, ref) => (
74:   <CommandPrimitive.Empty
75:     ref={ref}
76:     className="py-6 text-center text-sm"
77:     {...props}

```



```

78:   />
79: ))
80:
81: CommandEmpty.displayName = CommandPrimitive.Empty.displayName
82:
83: const CommandGroup = React.forwardRef<
84:   React.ElementRef<typeof CommandPrimitive.Group>,
85:   React.ComponentPropsWithoutRef<typeof CommandPrimitive.Group>
86: >(({ className, ...props }, ref) => (
87:   <CommandPrimitive.Group
88:     ref={ref}
89:     className={cn(
90:       "overflow-hidden p-1 text-foreground [&_cmdk-group-heading]:px-2 [&_cmdk-group-heading]:py-1.5 [&
91:       className
92:     )}
93:     {...props}
94:   />
95: ))
96:
97: CommandGroup.displayName = CommandPrimitive.Group.displayName
98:
99: const CommandSeparator = React.forwardRef<
100:   React.ElementRef<typeof CommandPrimitive.Separator>,
101:   React.ComponentPropsWithoutRef<typeof CommandPrimitive.Separator>
102: >(({ className, ...props }, ref) => (
103:   <CommandPrimitive.Separator
104:     ref={ref}
105:     className={cn("-mx-1 h-px bg-border", className)}
106:     {...props}
107:   />
108: ))
109: CommandSeparator.displayName = CommandPrimitive.Separator.displayName
110:
111: const CommandItem = React.forwardRef<
112:   React.ElementRef<typeof CommandPrimitive.Item>,
113:   React.ComponentPropsWithoutRef<typeof CommandPrimitive.Item>
114: >(({ className, ...props }, ref) => (
115:   <CommandPrimitive.Item
116:     ref={ref}
117:     className={cn(
118:       "relative flex cursor-default gap-2 select-none items-center rounded-sm px-2 py-1.5 text-sm outline-n
119:     className
120:   )}
121:   {...props}
122: />
123: ))
124:
125: CommandItem.displayName = CommandPrimitive.Item.displayName
126:
127: const CommandShortcut = ({
128:   className,
129:   ...props
130: }: React.HTMLAttributes<HTMLSpanElement>) => {
131:   return (
132:     <span
133:       className={cn(
134:         "ml-auto text-xs tracking-widest text-muted-foreground",
135:         className
136:       )}
137:       {...props}
138:     />
139:   )
140: }
141: CommandShortcut.displayName = "CommandShortcut"
142:
143: export {
144:   Command,
145:   CommandDialog,
146:   CommandInput,
147:   CommandList,
148:   CommandEmpty,
149:   CommandGroup,
150:   CommandItem,

```

```
151:   CommandShortcut,
152:   CommandSeparator,
153: }
```

■ File: components\ui\context-menu.tsx

```
=====
1: "use client"
2:
3: import * as React from "react"
4: import * as ContextMenuPrimitive from "@radix-ui/react-context-menu"
5: import { Check, ChevronRight, Circle } from "lucide-react"
6:
7: import { cn } from "@lib/Utils"
8:
9: const ContextMenu = ContextMenuPrimitive.Root
10:
11: const ContextMenuTrigger = ContextMenuPrimitive.Trigger
12:
13: const ContextMenuGroup = ContextMenuPrimitive.Group
14:
15: const ContextMenuPortal = ContextMenuPrimitive.Portal
16:
17: const ContextMenuSub = ContextMenuPrimitive.Sub
18:
19: const ContextMenuRadioGroup = ContextMenuPrimitive.RadioGroup
20:
21: const ContextMenuSubTrigger = React.forwardRef<
22:   React.ElementRef<typeof ContextMenuPrimitive.SubTrigger>,
23:   React.ComponentPropsWithoutRef<typeof ContextMenuPrimitive.SubTrigger> & {
24:     inset?: boolean
25:   }
26: >(({ className, inset, children, ...props }, ref) => (
27:   <ContextMenuPrimitive.SubTrigger
28:     ref={ref}
29:     className={cn(
30:       "flex cursor-default select-none items-center rounded-sm px-2 py-1.5 text-sm outline-none focus:bg-accent",
31:       inset && "pl-8",
32:       className
33:     )}
34:     {...props}
35:   >
36:     {children}
37:     <ChevronRight className="ml-auto h-4 w-4" />
38:   </ContextMenuPrimitive.SubTrigger>
39: ))
40: ContextMenuSubTrigger.displayName = ContextMenuPrimitive.SubTrigger.displayName
41:
42: const ContextMenuSubContent = React.forwardRef<
43:   React.ElementRef<typeof ContextMenuPrimitive.SubContent>,
44:   React.ComponentPropsWithoutRef<typeof ContextMenuPrimitive.SubContent>
45: >(({ className, ...props }, ref) => (
46:   <ContextMenuPrimitive.SubContent
47:     ref={ref}
48:     className={cn(
49:       "z-50 min-w-[8rem] overflow-hidden rounded-md border bg-popover p-1 text-popover-foreground shadow-md",
50:       className
51:     )}
52:     {...props}
53:   />
54: ))
55: ContextMenuSubContent.displayName = ContextMenuPrimitive.SubContent.displayName
56:
57: const ContextMenuContent = React.forwardRef<
58:   React.ElementRef<typeof ContextMenuPrimitive.Content>,
59:   React.ComponentPropsWithoutRef<typeof ContextMenuPrimitive.Content>
60: >(({ className, ...props }, ref) => (
61:   <ContextMenuPrimitive.Portal>
62:     <ContextMenuPrimitive.Content
63:       ref={ref}
64:       className={cn(
65:         "z-50 min-w-[8rem] overflow-hidden rounded-md border bg-popover p-1 text-popover-foreground shadow-m
```

```

66:         className
67:     )}
68:     {...props}
69: />
70: </ContextMenuPrimitive.Portal>
71: ))
72: ContextMenuContent.displayName = ContextMenuPrimitive.Content.displayName
73:
74: const ContextMenuItem = React.forwardRef<
75:   React.ElementRef<typeof ContextMenuPrimitive.Item>,
76:   React.ComponentPropsWithoutRef<typeof ContextMenuPrimitive.Item> & {
77:     inset?: boolean
78:   }
79: >(({ className, inset, ...props }, ref) => (
80:   <ContextMenuPrimitive.Item
81:     ref={ref}
82:     className={cn(
83:       "relative flex cursor-default select-none items-center rounded-sm px-2 py-1.5 text-sm outline-none fo
84:       inset && "pl-8",
85:       className
86:     )}
87:     {...props}
88:   />
89: ))
90: ContextMenuItem.displayName = ContextMenuPrimitive.Item.displayName
91:
92: const ContextMenuCheckboxItem = React.forwardRef<
93:   React.ElementRef<typeof ContextMenuPrimitive.CheckboxItem>,
94:   React.ComponentPropsWithoutRef<typeof ContextMenuPrimitive.CheckboxItem>
95: >(({ className, children, checked, ...props }, ref) => (
96:   <ContextMenuPrimitive.CheckboxItem
97:     ref={ref}
98:     className={cn(
99:       "relative flex cursor-default select-none items-center rounded-sm py-1.5 pl-8 pr-2 text-sm outline-no
100:     className
101:   )}
102:   checked={checked}
103:   {...props}
104:   >
105:     <span className="absolute left-2 flex h-3.5 w-3.5 items-center justify-center">
106:       <ContextMenuPrimitive.ItemIndicator>
107:         <Check className="h-4 w-4" />
108:       </ContextMenuPrimitive.ItemIndicator>
109:     </span>
110:     {children}
111:   </ContextMenuPrimitive.CheckboxItem>
112: ))
113: ContextMenuCheckboxItem.displayName =
114:   ContextMenuPrimitive.CheckboxItem.displayName
115:
116: const ContextMenuRadioItem = React.forwardRef<
117:   React.ElementRef<typeof ContextMenuPrimitive.RadioItem>,
118:   React.ComponentPropsWithoutRef<typeof ContextMenuPrimitive.RadioItem>
119: >(({ className, children, ...props }, ref) => (
120:   <ContextMenuPrimitive.RadioItem
121:     ref={ref}
122:     className={cn(
123:       "relative flex cursor-default select-none items-center rounded-sm py-1.5 pl-8 pr-2 text-sm outline-no
124:     className
125:   )}
126:   {...props}
127:   >
128:     <span className="absolute left-2 flex h-3.5 w-3.5 items-center justify-center">
129:       <ContextMenuPrimitive.ItemIndicator>
130:         <Circle className="h-2 w-2 fill-current" />
131:       </ContextMenuPrimitive.ItemIndicator>
132:     </span>
133:     {children}
134:   </ContextMenuPrimitive.RadioItem>
135: ))
136: ContextMenuRadioItem.displayName = ContextMenuPrimitive.RadioItem.displayName
137:
138: const ContextMenuLabel = React.forwardRef<

```

```

139:   React.ElementRef<typeof ContextMenuPrimitive.Label>,
140:   React.ComponentPropsWithoutRef<typeof ContextMenuPrimitive.Label> & {
141:     inset?: boolean
142:   }
143: >(({ className, inset, ...props }, ref) => (
144:   <ContextMenuPrimitive.Label
145:     ref={ref}
146:     className={cn(
147:       "px-2 py-1.5 text-sm font-semibold text-foreground",
148:       inset && "pl-8",
149:       className
150:     )}
151:     {...props}
152:   />
153: ))
154: ContextMenuLabel.displayName = ContextMenuPrimitive.Label.displayName
155:
156: const ContextMenuSeparator = React.forwardRef<
157:   React.ElementRef<typeof ContextMenuPrimitive.Separator>,
158:   React.ComponentPropsWithoutRef<typeof ContextMenuPrimitive.Separator>
159: >(({ className, ...props }, ref) => (
160:   <ContextMenuPrimitive.Separator
161:     ref={ref}
162:     className={cn("-mx-1 my-1 h-px bg-border", className)}
163:     {...props}
164:   />
165: ))
166: ContextMenuSeparator.displayName = ContextMenuPrimitive.Separator.displayName
167:
168: const ContextMenuShortcut = ({
169:   className,
170:   ...props
171: }: React.HTMLAttributes<HTMLSpanElement>) => {
172:   return (
173:     <span
174:       className={cn(
175:         "ml-auto text-xs tracking-widest text-muted-foreground",
176:         className
177:       )}
178:       {...props}
179:     />
180:   )
181: }
182: ContextMenuShortcut.displayName = "ContextMenuShortcut"
183:
184: export {
185:   ContextMenu,
186:   ContextMenuTrigger,
187:   ContextMenuContent,
188:   ContextMenuItem,
189:   ContextMenuCheckboxItem,
190:   ContextMenuRadioItem,
191:   ContextMenuLabel,
192:   ContextMenuSeparator,
193:   ContextMenuShortcut,
194:   ContextMenuGroup,
195:   ContextMenuPortal,
196:   ContextMenuSub,
197:   ContextMenuSubContent,
198:   ContextMenuSubTrigger,
199:   ContextMenuRadioGroup,
200: }

```

■ File: components/ui/dialog.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import * as DialogPrimitive from "@radix-ui/react-dialog"
5: import { X } from "lucide-react"
6:

```

```

7: import { cn } from "@/lib/utils"
8:
9: const Dialog = DialogPrimitive.Root
10:
11: const DialogTrigger = DialogPrimitive.Trigger
12:
13: const DialogPortal = DialogPrimitive.Portal
14:
15: const DialogClose = DialogPrimitive.Close
16:
17: const DialogOverlay = React.forwardRef<
18:   React.ElementRef<typeof DialogPrimitive.Overlay>,
19:   React.ComponentPropsWithoutRef<typeof DialogPrimitive.Overlay>
20: >(({ className, ...props }, ref) => (
21:   <DialogPrimitive.Overlay
22:     ref={ref}
23:     className={cn(
24:       "fixed inset-0 z-50 bg-black/80 data-[state=open]:animate-in data-[state=closed]:animate-out data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0",
25:       className
26:     )}
27:     {...props}
28:   />
29: ))
30: DialogOverlay.displayName = DialogPrimitive.Overlay.displayName
31:
32: const DialogContent = React.forwardRef<
33:   React.ElementRef<typeof DialogPrimitive.Content>,
34:   React.ComponentPropsWithoutRef<typeof DialogPrimitive.Content>
35: >(({ className, children, ...props }, ref) => (
36:   <DialogPortal>
37:     <DialogOverlay />
38:     <DialogPrimitive.Content
39:       ref={ref}
40:       className={cn(
41:         "fixed left-[50%] top-[50%] z-50 grid w-full max-w-lg translate-x-[-50%] translate-y-[-50%] gap-4 bg-white p-4 shadow-lg",
42:         className
43:       )}
44:       {...props}
45:     >
46:       {children}
47:       <DialogPrimitive.Close className="absolute right-4 top-4 rounded-sm opacity-70 ring-offset-background transition-opacity hover:opacity-100 focus:outline-none focus:ring-2 focus:ring-gray-400 focus:ring-offset-2 disabled:cursor-not-allowed disabled:opacity-50" />
48:       <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-lg",
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-none",
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-none",
103:       className
104:     )}
105:     checked={checked}
106:     {...props}
107:   >
108:     <span className="absolute left-2 flex h-3.5 w-3.5 items-center justify-center">
109:       <DropdownMenuPrimitive.ItemIndicator>
110:         <Check className="h-4 w-4" />
111:       </DropdownMenuPrimitive.ItemIndicator>
112:     </span>
113:     {children}
114:   </DropdownMenuPrimitive.CheckboxItem>
115: ))
116: DropdownMenuCheckboxItem.displayName =
117:   DropdownMenuPrimitive.CheckboxItem.displayName
118:
119: const DropdownMenuRadioItem = React.forwardRef<
120:   React.ElementRef<typeof DropdownMenuPrimitive.RadioItem>,
121:   React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.RadioItem>

```

```

122: >(({ className, children, ...props }, ref) => (
123:   <DropdownMenuPrimitive.RadioItem
124:     ref={ref}
125:     className={cn(
126:       "relative flex cursor-default select-none items-center rounded-sm py-1.5 pl-8 pr-2 text-sm outline-none"
127:       className
128:     )}
129:     {...props}
130:   >
131:     <span className="absolute left-2 flex h-3.5 w-3.5 items-center justify-center">
132:       <DropdownMenuPrimitive.ItemIndicator>
133:         <Circle className="h-2 w-2 fill-current" />
134:       </DropdownMenuPrimitive.ItemIndicator>
135:     </span>
136:     {children}
137:   </DropdownMenuPrimitive.RadioItem>
138: ))
139: DropdownMenuRadioItem.displayName = DropdownMenuPrimitive.RadioItem.displayName
140:
141: const DropdownMenuLabel = React.forwardRef<
142:   React.ElementRef<typeof DropdownMenuPrimitive.Label>,
143:   React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.Label> & {
144:     inset?: boolean
145:   }
146: >(({ className, inset, ...props }, ref) => (
147:   <DropdownMenuPrimitive.Label
148:     ref={ref}
149:     className={cn(
150:       "px-2 py-1.5 text-sm font-semibold",
151:       inset && "pl-8",
152:       className
153:     )}
154:     {...props}
155:   />
156: ))
157: DropdownMenuLabel.displayName = DropdownMenuPrimitive.Label.displayName
158:
159: const DropdownMenuSeparator = React.forwardRef<
160:   React.ElementRef<typeof DropdownMenuPrimitive.Separator>,
161:   React.ComponentPropsWithoutRef<typeof DropdownMenuPrimitive.Separator>
162: >(({ className, ...props }, ref) => (
163:   <DropdownMenuPrimitive.Separator
164:     ref={ref}
165:     className={cn("-mx-1 my-1 h-px bg-muted", className)}
166:     {...props}
167:   />
168: ))
169: DropdownMenuSeparator.displayName = DropdownMenuPrimitive.Separator.displayName
170:
171: const DropdownMenuShortcut = ({
172:   className,
173:   ...props
174: }: React.HTMLAttributes<HTMLSpanElement>) => {
175:   return (
176:     <span
177:       className={cn("ml-auto text-xs tracking-widest opacity-60", className)}
178:       {...props}
179:     />
180:   )
181: }
182: DropdownMenuShortcut.displayName = "DropdownMenuShortcut"
183:
184: export {
185:   DropdownMenu,
186:   DropdownMenuTrigger,
187:   DropdownMenuContent,
188:   DropdownMenuItem,
189:   DropdownMenuCheckboxItem,
190:   DropdownMenuRadioItem,
191:   DropdownMenuLabel,
192:   DropdownMenuSeparator,
193:   DropdownMenuShortcut,
194:   DropdownMenuGroup,

```

```

195:   DropdownMenuPortal,
196:   DropdownMenuSub,
197:   DropdownMenuSubContent,
198:   DropdownMenuSubTrigger,
199:   DropdownMenuRadioGroup,
200: }

```

■ File: components/ui/form.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import * as LabelPrimitive from "@radix-ui/react-label"
5: import { Slot } from "@radix-ui/react-slot"
6: import {
7:   Controller,
8:   ControllerProps,
9:   FieldPath,
10:  FieldValues,
11:  FormProvider,
12:  useFormContext,
13: } from "react-hook-form"
14:
15: import { cn } from "@/lib/utils"
16: import { Label } from "@/components/ui/label"
17:
18: const Form = FormProvider
19:
20: type FormFieldContextValue<
21:   TFieldValues extends FieldValues = FieldValues,
22:   TName extends FieldPath<TFieldValues> = FieldPath<TFieldValues>
23: > = {
24:   name: TName
25: }
26:
27: const FormFieldContext = React.createContext<FormFieldContextValue>(
28:   {} as FormFieldContextValue
29: )
30:
31: const FormField = <
32:   TFieldValues extends FieldValues = FieldValues,
33:   TName extends FieldPath<TFieldValues> = FieldPath<TFieldValues>
34: >({
35:   ...props
36: }): ControllerProps<TFieldValues, TName> => {
37:   return (
38:     <FormFieldContext.Provider value={{ name: props.name }}>
39:       <Controller {...props} />
40:     </FormFieldContext.Provider>
41:   )
42: }
43:
44: const useFormField = () => {
45:   const fieldContext = React.useContext(FormFieldContext)
46:   const itemContext = React.useContext(FormItemContext)
47:   const { getFieldState, formState } = useFormContext()
48:
49:   const fieldState = getFieldState(fieldContext.name, formState)
50:
51:   if (!fieldContext) {
52:     throw new Error("useFormField should be used within <FormField>")
53:   }
54:
55:   const { id } = itemContext
56:
57:   return {
58:     id,
59:     name: fieldContext.name,
60:     formItemId: `${id}-form-item`,
61:     formDescriptionId: `${id}-form-item-description`,
62:     formMessageId: `${id}-form-item-message`,

```

```

63:     ...fieldState,
64:   }
65: }
66:
67: type FormItemContextValue = {
68:   id: string
69: }
70:
71: const FormItemContext = React.createContext<FormItemContextValue>(
72:   {} as FormItemContextValue
73: )
74:
75: const FormItem = React.forwardRef<
76:   HTMLDivElement,
77:   React.HTMLAttributes<HTMLDivElement>
78: >(({ className, ...props }, ref) => {
79:   const id = React.useId()
80:
81:   return (
82:     <FormItemContext.Provider value={{ id }}>
83:       <div ref={ref} className={cn("space-y-2", className)} {...props} />
84:     </FormItemContext.Provider>
85:   )
86: })
87: FormItem.displayName = "FormItem"
88:
89: const FormLabel = React.forwardRef<
90:   React.ElementRef<typeof LabelPrimitive.Root>,
91:   React.ComponentPropsWithoutRef<typeof LabelPrimitive.Root>
92: >(({ className, ...props }, ref) => {
93:   const { error, formItemId } = useFormField()
94:
95:   return (
96:     <Label
97:       ref={ref}
98:       className={cn(error && "text-destructive", className)}
99:       htmlFor={formItemId}
100:       {...props}
101:     />
102:   )
103: })
104: FormLabel.displayName = "FormLabel"
105:
106: const FormControl = React.forwardRef<
107:   React.ElementRef<typeof Slot>,
108:   React.ComponentPropsWithoutRef<typeof Slot>
109: >(({ ...props }, ref) => {
110:   const { error, formItemId, formDescriptionId, formMessageId } = useFormField()
111:
112:   return (
113:     <Slot
114:       ref={ref}
115:       id={formItemId}
116:       aria-describedby={
117:         !error
118:           ? `${formDescriptionId}`
119:           : `${formDescriptionId} ${formMessageId}`
120:       }
121:       aria-invalid={!!error}
122:       {...props}
123:     />
124:   )
125: })
126: FormControl.displayName = "FormControl"
127:
128: const FormDescription = React.forwardRef<
129:   HTMLParagraphElement,
130:   React.HTMLAttributes<HTMLParagraphElement>
131: >(({ className, ...props }, ref) => {
132:   const { formDescriptionId } = useFormField()
133:
134:   return (
135:     <p

```

```

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

```

File: components\ui\hover-card.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import * as HoverCardPrimitive from "@radix-ui/react-hover-card"
5:
6: import { cn } from "@lib/Utils"
7:
8: const HoverCard = HoverCardPrimitive.Root
9:
10: const HoverCardTrigger = HoverCardPrimitive.Trigger
11:
12: const HoverCardContent = React.forwardRef<
13:   React.ElementRef<typeof HoverCardPrimitive.Content>,
14:   React.ComponentPropsWithoutRef<typeof HoverCardPrimitive.Content>
15: >(({ className, align = "center", sideOffset = 4, ...props }, ref) => (
16:   <HoverCardPrimitive.Content
17:     ref={ref}
18:     align={align}
19:     sideOffset={sideOffset}
20:     className={cn(
21:       "z-50 w-64 rounded-md border bg-popover p-4 text-popover-foreground shadow-md outline-none data-[state=open]:animate-in data-[state=closed]:animate-out data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0",
22:       className
23:     )}
24:     {...props}
25:   />

```

```

26: ))
27: HoverCardContent.displayName = HoverCardPrimitive.Content.displayName
28:
29: export { HoverCard, HoverCardTrigger, HoverCardContent }

```

■ File: components/ui/input-otp.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import { OTPInput, OTPInputContext } from "input-otp"
5: import { Dot } from "lucide-react"
6:
7: import { cn } from "@/lib/utils"
8:
9: const InputOTP = React.forwardRef<
10:   React.ElementRef<typeof OTPInput>,
11:   React.ComponentPropsWithoutRef<typeof OTPInput>
12: >(({ className, containerClassName, ...props }, ref) => (
13:   <OTPInput
14:     ref={ref}
15:     containerClassName={cn(
16:       "flex items-center gap-2 has-[:disabled]:opacity-50",
17:       containerClassName
18:     )}
19:     className={cn("disabled:cursor-not-allowed", className)}
20:     {...props}
21:   />
22: ))
23: InputOTP.displayName = "InputOTP"
24:
25: const InputOTPGroup = React.forwardRef<
26:   React.ElementRef<"div">,
27:   React.ComponentPropsWithoutRef<"div">
28: >(({ className, ...props }, ref) => (
29:   <div ref={ref} className={cn("flex items-center", className)} {...props} />
30: ))
31: InputOTPGroup.displayName = "InputOTPGroup"
32:
33: const InputOTPSlot = React.forwardRef<
34:   React.ElementRef<"div">,
35:   React.ComponentPropsWithoutRef<"div"> & { index: number }
36: >(({ index, className, ...props }, ref) => {
37:   const inputOTPContext = React.useContext(OTPInputContext)
38:   const { char, hasFakeCaret, isActive } = inputOTPContext.slots[index]
39:
40:   return (
41:     <div
42:       ref={ref}
43:       className={cn(
44:         "relative flex h-10 w-10 items-center justify-center border-y border-r border-input text-sm 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-[sta
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 sha
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 fo
120:       inset && "pl-8",
121:       className
122:     )}
123:     {...props}
124:   />
125: ))
126: MenubarItem.displayName = MenubarPrimitive.Item.displayName
127:
128: const MenubarCheckboxItem = React.forwardRef<
129:   React.ElementRef<typeof MenubarPrimitive.CheckboxItem>,
130:   React.ComponentPropsWithoutRef<typeof MenubarPrimitive.CheckboxItem>
131: >(({ className, children, checked, ...props }, ref) => (
132:   <MenubarPrimitive.CheckboxItem
133:     ref={ref}
134:     className={cn(
135:       "relative flex cursor-default select-none items-center rounded-sm py-1.5 pl-8 pr-2 text-sm outline-no
136:       className
137:     )}
138:     checked={checked}
139:     {...props}
140:   >
141:     <span className="absolute left-2 flex h-3.5 w-3.5 items-center justify-center">
142:       <MenubarPrimitive.ItemIndicator>
143:         <Check className="h-4 w-4" />
144:       </MenubarPrimitive.ItemIndicator>
145:     </span>
146:     {children}
147:   </MenubarPrimitive.CheckboxItem>
148: ))
149: MenubarCheckboxItem.displayName = MenubarPrimitive.CheckboxItem.displayName

```

```

150:
151: const MenubarRadioItem = React.forwardRef<
152:   React.ElementRef<typeof MenubarPrimitive.RadioItem>,
153:   React.ComponentPropsWithoutRef<typeof MenubarPrimitive.RadioItem>
154: >(({ className, children, ...props }, ref) => (
155:   <MenubarPrimitive.RadioItem
156:     ref={ref}
157:     className={cn(
158:       "relative flex cursor-default select-none items-center rounded-sm py-1.5 pl-8 pr-2 text-sm outline-no
159:       className
160:     )}
161:     {...props}
162:   >
163:     <span className="absolute left-2 flex h-3.5 w-3.5 items-center justify-center">
164:       <MenubarPrimitive.ItemIndicator>
165:         <Circle className="h-2 w-2 fill-current" />
166:       </MenubarPrimitive.ItemIndicator>
167:     </span>
168:     {children}
169:   </MenubarPrimitive.RadioItem>
170: ))
171: MenubarRadioItem.displayName = MenubarPrimitive.RadioItem.displayName
172:
173: const MenubarLabel = React.forwardRef<
174:   React.ElementRef<typeof MenubarPrimitive.Label>,
175:   React.ComponentPropsWithoutRef<typeof MenubarPrimitive.Label> & {
176:     inset?: boolean
177:   }
178: >(({ className, inset, ...props }, ref) => (
179:   <MenubarPrimitive.Label
180:     ref={ref}
181:     className={cn(
182:       "px-2 py-1.5 text-sm font-semibold",
183:       inset && "pl-8",
184:       className
185:     )}
186:     {...props}
187:   />
188: ))
189: MenubarLabel.displayName = MenubarPrimitive.Label.displayName
190:
191: const MenubarSeparator = React.forwardRef<
192:   React.ElementRef<typeof MenubarPrimitive.Separator>,
193:   React.ComponentPropsWithoutRef<typeof MenubarPrimitive.Separator>
194: >(({ className, ...props }, ref) => (
195:   <MenubarPrimitive.Separator
196:     ref={ref}
197:     className={cn("-mx-1 my-1 h-px bg-muted", className)}
198:     {...props}
199:   />
200: ))
201: MenubarSeparator.displayName = MenubarPrimitive.Separator.displayName
202:
203: const MenubarShortcut = ({
204:   className,
205:   ...props
206: }: React.HTMLAttributes<HTMLSpanElement>) => {
207:   return (
208:     <span
209:       className={cn(
210:         "ml-auto text-xs tracking-widest text-muted-foreground",
211:         className
212:       )}
213:       {...props}
214:     />
215:   )
216: }
217: MenubarShortcut.displayName = "MenubarShortcut"
218:
219: export {
220:   Menubar,
221:   MenubarMenu,
222:   MenubarTrigger,

```

```

223:   MenubarContent,
224:   MenubarItem,
225:   MenubarSeparator,
226:   MenubarLabel,
227:   MenubarCheckboxItem,
228:   MenubarRadioGroup,
229:   MenubarRadioItem,
230:   MenubarPortal,
231:   MenubarSubContent,
232:   MenubarSubTrigger,
233:   MenubarGroup,
234:   MenubarSub,
235:   MenubarShortcut,
236: }

```

■ File: components\ui\navigation-menu.tsx

```

1: import * as React from "react"
2: import * as NavigationMenuPrimitive from "@radix-ui/react-navigation-menu"
3: import { cva } from "class-variance-authority"
4: import { ChevronDown } from "lucide-react"
5:
6: import { cn } from "@/lib/utils"
7:
8: const NavigationMenu = React.forwardRef<
9:   React.ElementRef<typeof NavigationMenuPrimitive.Root>,
10:   React.ComponentPropsWithoutRef<typeof NavigationMenuPrimitive.Root>
11: >(({ className, children, ...props }, ref) => (
12:   <NavigationMenuPrimitive.Root
13:     ref={ref}
14:     className={cn(
15:       "relative z-10 flex max-w-max flex-1 items-center justify-center",
16:       className
17:     )}
18:     {...props}
19:   >
20:     {children}
21:     <NavigationMenuViewport />
22:   </NavigationMenuPrimitive.Root>
23: ))
24: NavigationMenu.displayName = NavigationMenuPrimitive.Root.displayName
25:
26: const NavigationMenuList = React.forwardRef<
27:   React.ElementRef<typeof NavigationMenuPrimitive.List>,
28:   React.ComponentPropsWithoutRef<typeof NavigationMenuPrimitive.List>
29: >(({ className, ...props }, ref) => (
30:   <NavigationMenuPrimitive.List
31:     ref={ref}
32:     className={cn(
33:       "group flex flex-1 list-none items-center justify-center space-x-1",
34:       className
35:     )}
36:     {...props}
37:   />
38: ))
39: NavigationMenuList.displayName = NavigationMenuPrimitive.List.displayName
40:
41: const NavigationMenuItem = NavigationMenuPrimitive.Item
42:
43: const navigationMenuTriggerStyle = cva(
44:   "group inline-flex h-10 w-max items-center justify-center rounded-md bg-background px-4 py-2 text-sm font-medium",
45: )
46:
47: const NavigationMenuTrigger = React.forwardRef<
48:   React.ElementRef<typeof NavigationMenuPrimitive.Trigger>,
49:   React.ComponentPropsWithoutRef<typeof NavigationMenuPrimitive.Trigger>
50: >(({ className, children, ...props }, ref) => (
51:   <NavigationMenuPrimitive.Trigger
52:     ref={ref}
53:     className={cn(navigationMenuTriggerStyle(), "group", className)}
54:     {...props}

```

```

55:   >
56:   {children}{" "}
57:   <ChevronDown
58:     className="relative top-[1px] ml-1 h-3 w-3 transition duration-200 group-data-[state=open]:rotate-180
59:     aria-hidden="true"
60:   />
61: </NavigationMenuPrimitive.Trigger>
62: ))
63: NavigationMenuTrigger.displayName = NavigationMenuPrimitive.Trigger.displayName
64:
65: const NavigationMenuContent = React.forwardRef<
66:   React.ElementRef<typeof NavigationMenuPrimitive.Content>,
67:   React.ComponentPropsWithoutRef<typeof NavigationMenuPrimitive.Content>
68: >(({ className, ...props }, ref) => (
69:   <NavigationMenuPrimitive.Content
70:     ref={ref}
71:     className={cn(
72:       "left-0 top-0 w-full data-[motion^=from]:animate-in data-[motion^=to]:animate-out data-[motion^=from]
73:       className
74:     )}
75:     {...props}
76:   />
77: ))
78: NavigationMenuContent.displayName = NavigationMenuPrimitive.Content.displayName
79:
80: const NavigationMenuLink = NavigationMenuPrimitive.Link
81:
82: const NavigationMenuViewport = React.forwardRef<
83:   React.ElementRef<typeof NavigationMenuPrimitive.Viewport>,
84:   React.ComponentPropsWithoutRef<typeof NavigationMenuPrimitive.Viewport>
85: >(({ className, ...props }, ref) => (
86:   <div className={cn("absolute left-0 top-full flex justify-center")}>
87:     <NavigationMenuPrimitive.Viewport
88:       className={cn(
89:         "origin-top-center relative mt-1.5 h-[var(--radix-navigation-menu-viewport-height)] w-full overflow
90:         className
91:       )}
92:       ref={ref}
93:       {...props}
94:     />
95:   </div>
96: ))
97: NavigationMenuViewport.displayName =
98:   NavigationMenuPrimitive.Viewport.displayName
99:
100: const NavigationMenuIndicator = React.forwardRef<
101:   React.ElementRef<typeof NavigationMenuPrimitive.Indicator>,
102:   React.ComponentPropsWithoutRef<typeof NavigationMenuPrimitive.Indicator>
103: >(({ className, ...props }, ref) => (
104:   <NavigationMenuPrimitive.Indicator
105:     ref={ref}
106:     className={cn(
107:       "top-full z-[1] flex h-1.5 items-end justify-center overflow-hidden data-[state=visible]:animate-in d
108:       className
109:     )}
110:     {...props}
111:   >
112:     <div className="relative top-[60%] h-2 w-2 rotate-45 rounded-tl-sm bg-border shadow-md" />
113:   </NavigationMenuPrimitive.Indicator>
114: ))
115: NavigationMenuIndicator.displayName =
116:   NavigationMenuPrimitive.Indicator.displayName
117:
118: export {
119:   navigationMenuTriggerStyle,
120:   NavigationMenu,
121:   NavigationMenuList,
122:   NavigationMenuItem,
123:   NavigationMenuContent,
124:   NavigationMenuTrigger,
125:   NavigationMenuLink,
126:   NavigationMenuIndicator,
127:   NavigationMenuViewport,

```

128: }

■ File: components\ui\pagination.tsx

```
=====
1: import * as React from "react"
2: import { ChevronLeft, ChevronRight, MoreHorizontal } from "lucide-react"
3:
4: import { cn } from "@lib/utils"
5: import { ButtonProps, buttonVariants } from "@components/ui/button"
6:
7: const Pagination = ({ className, ...props }: React.ComponentProps<"nav">) => (
8:   <nav
9:     role="navigation"
10:    aria-label="pagination"
11:    className={cn("mx-auto flex w-full justify-center", className)}
12:    {...props}
13:  />
14: )
15: Pagination.displayName = "Pagination"
16:
17: const PaginationContent = React.forwardRef<
18:   HTMLUListElement,
19:   React.ComponentProps<"ul">
20: >(({ className, ...props }, ref) => (
21:   <ul
22:     ref={ref}
23:     className={cn("flex flex-row items-center gap-1", className)}
24:     {...props}
25:   />
26: ))
27: PaginationContent.displayName = "PaginationContent"
28:
29: const PaginationItem = React.forwardRef<
30:   HTMLLIElement,
31:   React.ComponentProps<"li">
32: >(({ className, ...props }, ref) => (
33:   <li ref={ref} className={cn("", className)} {...props} />
34: ))
35: PaginationItem.displayName = "PaginationItem"
36:
37: type PaginationLinkProps = {
38:   isActive?: boolean
39: } & Pick<ButtonProps, "size"> &
40:   React.ComponentProps<"a">
41:
42: const PaginationLink = ({
43:   className,
44:   isActive,
45:   size = "icon",
46:   ...props
47: }: PaginationLinkProps) => (
48:   <a
49:     aria-current={isActive ? "page" : undefined}
50:     className={cn(
51:       buttonVariants({
52:         variant: isActive ? "outline" : "ghost",
53:         size,
54:       }),
55:       className
56:     )}
57:     {...props}
58:   />
59: )
60: PaginationLink.displayName = "PaginationLink"
61:
62: const PaginationPrevious = ({
63:   className,
64:   ...props
65: }: React.ComponentProps<typeof PaginationLink>) => (
66:   <PaginationLink
67:     aria-label="Go to previous page"
```

```

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

```

■ File: components/ui/popover.tsx

```

1: "use client"
2:
3: import * as React from "react"
4: import * as PopoverPrimitive from "@radix-ui/react-popover"
5:
6: import { cn } from "@/lib/utils"
7:
8: const Popover = PopoverPrimitive.Root
9:
10: const PopoverTrigger = PopoverPrimitive.Trigger
11:
12: const PopoverContent = React.forwardRef<
13:   React.ElementRef<typeof PopoverPrimitive.Content>,
14:   React.ComponentPropsWithoutRef<typeof PopoverPrimitive.Content>
15: >(({ className, align = "center", sideOffset = 4, ...props }, ref) => (
16:   <PopoverPrimitive.Portal>
17:     <PopoverPrimitive.Content
18:       ref={ref}

```

```

19:     align={align}
20:     sideOffset={sideOffset}
21:     className={cn(
22:       "z-50 w-72 rounded-md border bg-popover p-4 text-popover-foreground shadow-md outline-none data-[st
23:       className
24:     )}
25:     {...props}
26:   />
27: </PopoverPrimitive.Portal>
28: ))
29: PopoverContent.displayName = PopoverPrimitive.Content.displayName
30:
31: export { Popover, PopoverTrigger, PopoverContent }

```

■ File: components\ui\progress.tsx

```

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

```

■ File: components\ui\radio-group.tsx

```

1: "use client"
2:
3: import * as React from "react"
4: import * as RadioGroupPrimitive from "@radix-ui/react-radio-group"
5: import { Circle } from "lucide-react"
6:
7: import { cn } from "@lib/Utils"
8:
9: const RadioGroup = React.forwardRef<
10:   React.ElementRef<typeof RadioGroupPrimitive.Root>,
11:   React.ComponentPropsWithoutRef<typeof RadioGroupPrimitive.Root>
12: >(({ className, ...props }, ref) => {
13:   return (
14:     <RadioGroupPrimitive.Root
15:       className={cn("grid gap-2", className)}
16:       {...props}
17:       ref={ref}
18:     />
19:   )
20: })
21: RadioGroup.displayName = RadioGroupPrimitive.Root.displayName
22:

```

```

23: const RadioGroupItem = React.forwardRef<
24:   React.ElementRef<typeof RadioGroupPrimitive.Item>,
25:   React.ComponentPropsWithoutRef<typeof RadioGroupPrimitive.Item>
26: >(({ className, ...props }, ref) => {
27:   return (
28:     <RadioGroupPrimitive.Item
29:       ref={ref}
30:       className={cn(
31:         "aspect-square h-4 w-4 rounded-full border border-primary text-primary ring-offset-background focus
32:       className
33:     )}
34:     {...props}
35:   >
36:     <RadioGroupPrimitive.Indicator className="flex items-center justify-center">
37:       <Circle className="h-2.5 w-2.5 fill-current text-current" />
38:     </RadioGroupPrimitive.Indicator>
39:   </RadioGroupPrimitive.Item>
40: )
41: })
42: RadioGroupItem.displayName = RadioGroupPrimitive.Item.displayName
43:
44: export { RadioGroup, RadioGroupItem }

```

■ File: components\ui\resizable.tsx

```

1: "use client"
2:
3: import { GripVertical } from "lucide-react"
4: import * as ResizablePrimitive from "react-resizable-panels"
5:
6: import { cn } from "@lib/Utils"
7:
8: const ResizablePanelGroup = ({
9:   className,
10:  ...props
11: }: React.ComponentProps<typeof ResizablePrimitive.PanelGroup>) => (
12:   <ResizablePrimitive.PanelGroup
13:     className={cn(
14:       "flex h-full w-full data-[panel-group-direction=vertical]:flex-col",
15:     className
16:   )}
17:   {...props}
18: />
19: )
20:
21: const ResizablePanel = ResizablePrimitive.Panel
22:
23: const ResizableHandle = ({
24:   withHandle,
25:   className,
26:   ...props
27: }: React.ComponentProps<typeof ResizablePrimitive.PanelResizeHandle> & {
28:   withHandle?: boolean
29: }) => (
30:   <ResizablePrimitive.PanelResizeHandle
31:     className={cn(
32:       "relative flex w-px items-center justify-center bg-border after:absolute after:inset-y-0 after:left-1
33:     className
34:   )}
35:   {...props}
36:   >
37:     {withHandle && (
38:       <div className="z-10 flex h-4 w-3 items-center justify-center rounded-sm border bg-border">
39:         <GripVertical className="h-2.5 w-2.5" />
40:       </div>
41:     )}
42:   </ResizablePrimitive.PanelResizeHandle>
43: )
44:
45: export { ResizablePanelGroup, ResizablePanel, ResizableHandle }

```


■ File: components/ui/scroll-area.tsx

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

■ File: components/ui/select.tsx

```
=====
1: "use client"
2:
3: import * as React from "react"
4: import * as SelectPrimitive from "@radix-ui/react-select"
5: import { Check, ChevronDown, ChevronUp } from "lucide-react"
6:
7: import { cn } from "@/lib/utils"
8:
9: const Select = SelectPrimitive.Root
10:
11: const SelectGroup = SelectPrimitive.Group
12:
13: const SelectValue = SelectPrimitive.Value
14:
15: const SelectTrigger = React.forwardRef<
16:   React.ElementRef<typeof SelectPrimitive.Trigger>,
17:   React.ComponentPropsWithoutRef<typeof SelectPrimitive.Trigger>
18: >(({ className, children, ...props }, ref) => (
```

```

19:   <SelectPrimitive.Trigger
20:     ref={ref}
21:     className={cn(
22:       "flex h-10 w-full items-center justify-between rounded-md border border-input bg-background px-3 py-2
23:       className
24:     )}
25:     {...props}
26:   >
27:     {children}
28:     <SelectPrimitive.Icon asChild>
29:       <ChevronDown className="h-4 w-4 opacity-50" />
30:     </SelectPrimitive.Icon>
31:   </SelectPrimitive.Trigger>
32: ))
33: SelectTrigger.displayName = SelectPrimitive.Trigger.displayName
34:
35: const SelectScrollUpButton = React.forwardRef<
36:   React.ElementRef<typeof SelectPrimitive.ScrollUpButton>,
37:   React.ComponentPropsWithoutRef<typeof SelectPrimitive.ScrollUpButton>
38: >(({ className, ...props }, ref) => (
39:   <SelectPrimitive.ScrollUpButton
40:     ref={ref}
41:     className={cn(
42:       "flex cursor-default items-center justify-center py-1",
43:       className
44:     )}
45:     {...props}
46:   >
47:     <ChevronUp className="h-4 w-4" />
48:   </SelectPrimitive.ScrollUpButton>
49: ))
50: SelectScrollUpButton.displayName = SelectPrimitive.ScrollUpButton.displayName
51:
52: const SelectScrollDownButton = React.forwardRef<
53:   React.ElementRef<typeof SelectPrimitive.ScrollDownButton>,
54:   React.ComponentPropsWithoutRef<typeof SelectPrimitive.ScrollDownButton>
55: >(({ className, ...props }, ref) => (
56:   <SelectPrimitive.ScrollDownButton
57:     ref={ref}
58:     className={cn(
59:       "flex cursor-default items-center justify-center py-1",
60:       className
61:     )}
62:     {...props}
63:   >
64:     <ChevronDown className="h-4 w-4" />
65:   </SelectPrimitive.ScrollDownButton>
66: ))
67: SelectScrollDownButton.displayName =
68:   SelectPrimitive.ScrollDownButton.displayName
69:
70: const SelectContent = React.forwardRef<
71:   React.ElementRef<typeof SelectPrimitive.Content>,
72:   React.ComponentPropsWithoutRef<typeof SelectPrimitive.Content>
73: >(({ className, children, position = "popper", ...props }, ref) => (
74:   <SelectPrimitive.Portal>
75:     <SelectPrimitive.Content
76:       ref={ref}
77:       className={cn(
78:         "relative z-50 max-h-96 min-w-[8rem] overflow-hidden rounded-md border bg-popover text-popover-forever
79:         position === "popper" &&
80:         "data-[side=bottom]:translate-y-1 data-[side=left]:-translate-x-1 data-[side=right]:translate-x-1
81:         className
82:       )}
83:       position={position}
84:       {...props}
85:     >
86:       <SelectScrollUpButton />
87:       <SelectPrimitive.Viewport
88:         className={cn(
89:           "p-1",
90:           position === "popper" &&
91:           "h-[var(--radix-select-trigger-height)] w-full min-w-[var(--radix-select-trigger-width)]"

```

```

92:         )}
93:     >
94:         {children}
95:     </SelectPrimitive.Viewport>
96:     <SelectScrollDownButton />
97: </SelectPrimitive.Content>
98: </SelectPrimitive.Portal>
99: ))
100: SelectContent.displayName = SelectPrimitive.Content.displayName
101:
102: const SelectLabel = React.forwardRef<
103:   React.ElementRef<typeof SelectPrimitive.Label>,
104:   React.ComponentPropsWithoutRef<typeof SelectPrimitive.Label>
105: >(({ className, ...props }, ref) => (
106:   <SelectPrimitive.Label
107:     ref={ref}
108:     className={cn("py-1.5 pl-8 pr-2 text-sm font-semibold", className)}
109:     {...props}
110:   />
111: ))
112: SelectLabel.displayName = SelectPrimitive.Label.displayName
113:
114: const SelectItem = React.forwardRef<
115:   React.ElementRef<typeof SelectPrimitive.Item>,
116:   React.ComponentPropsWithoutRef<typeof SelectPrimitive.Item>
117: >(({ className, children, ...props }, ref) => (
118:   <SelectPrimitive.Item
119:     ref={ref}
120:     className={cn(
121:       "relative flex w-full cursor-default select-none items-center rounded-sm py-1.5 pl-8 pr-2 text-sm 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 data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0",
35:   {
```

```

36:     variants: {
37:       side: {
38:         top: "inset-x-0 top-0 border-b data-[state=closed]:slide-out-to-top data-[state=open]:slide-in-from-
39:         bottom:
40:           "inset-x-0 bottom-0 border-t data-[state=closed]:slide-out-to-bottom data-[state=open]:slide-in-f
41:         left: "inset-y-0 left-0 h-full w-3/4 border-r data-[state=closed]:slide-out-to-left data-[state=open]
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]
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:     onOpenChange?: (open: boolean) => void
56:   }
57: >((
58:   {
59:     {
60:       defaultOpen = true,
61:       open: openProp,
62:       onOpenChange: setOpenProp,
63:       className,
64:       style,
65:       children,
66:       ...props
67:     },
68:     ref
69:   ) => {
70:     const isMobile = useIsMobile()
71:     const [openMobile, setOpenMobile] = React.useState(false)
72:
73:     // This is the internal state of the sidebar.
74:     // We use openProp and setOpenProp for control from outside the component.
75:     const [_open, _setOpen] = React.useState(defaultOpen)
76:     const open = openProp ?? _open
77:     const setOpen = React.useCallback(
78:       (value: boolean | ((value: boolean) => boolean)) => {
79:         const openState = typeof value === "function" ? value(open) : value
80:         if (setOpenProp) {
81:           setOpenProp(openState)
82:         } else {
83:           _setOpen(openState)
84:         }
85:
86:         // This sets the cookie to keep the sidebar state.
87:         document.cookie = `${SIDEBAR_COOKIE_NAME}=${openState}; path=/; max-age=${SIDEBAR_COOKIE_MAX_AGE}`
88:       },
89:       [setOpenProp, open]
90:     )
91:
92:     // Helper to toggle the sidebar.
93:     const toggleSidebar = React.useCallback(() => {
94:       return isMobile
95:         ? setOpenMobile((open) => !open)
96:         : setOpen((open) => !open)
97:     }, [isMobile, setOpen, setOpenMobile])
98:
99:     // Adds a keyboard shortcut to toggle the sidebar.
100:    React.useEffect(() => {
101:      const handleKeyDown = (event: KeyboardEvent) => {
102:        if (
103:          event.key === SIDEBAR_KEYBOARD_SHORTCUT &&
104:          (event.metaKey || event.ctrlKey)
105:        ) {
106:          event.preventDefault()
107:          toggleSidebar()
108:        }
109:      }

```

```

110:
111:     window.addEventListener("keydown", handleKeyDown)
112:     return () => window.removeEventListener("keydown", handleKeyDown)
113: }, [toggleSidebar])
114:
115: // We add a state so that we can do data-state="expanded" or "collapsed".
116: // This makes it easier to style the sidebar with Tailwind classes.
117: const state = open ? "expanded" : "collapsed"
118:
119: const contextValue = React.useMemo<SidebarContext>(
120:   () => ({
121:     state,
122:     open,
123:     setOpen,
124:     isMobile,
125:     openMobile,
126:     setOpenMobile,
127:     toggleSidebar,
128:   }),
129:   [state, open, setOpen, isMobile, openMobile, setOpenMobile, toggleSidebar]
130: )
131:
132: return (
133:   <SidebarContext.Provider value={contextValue}>
134:     <TooltipProvider delayDuration={0}>
135:       <div
136:         style={
137:           {
138:             "--sidebar-width": SIDEBAR_WIDTH,
139:             "--sidebar-width-icon": SIDEBAR_WIDTH_ICON,
140:             ...style,
141:           } as React.CSSProperties
142:         }
143:         className={cn(
144:           "group/sidebar-wrapper flex min-h-svh w-full has-[[data-variant=inset]]:bg-sidebar",
145:           className
146:         )}
147:         ref={ref}
148:         {...props}
149:       >
150:         {children}
151:       </div>
152:     </TooltipProvider>
153:   </SidebarContext.Provider>
154: )
155: }
156: )
157: SidebarProvider.displayName = "SidebarProvider"
158:
159: const Sidebar = React.forwardRef<
160:   HTMLDivElement,
161:   React.ComponentProps<"div"> & {
162:     side?: "left" | "right"
163:     variant?: "sidebar" | "floating" | "inset"
164:     collapsible?: "offcanvas" | "icon" | "none"
165:   }
166: >((
167:   (
168:     {
169:       side = "left",
170:       variant = "sidebar",
171:       collapsible = "offcanvas",
172:       className,
173:       children,
174:       ...props
175:     },
176:     ref
177:   ) => {
178:     const { isMobile, state, openMobile, setOpenMobile } = useSidebar()
179:
180:     if (collapsible === "none") {
181:       return (
182:         <div

```



```

183:         className={cn(
184:             "flex h-full w-[--sidebar-width] flex-col bg-sidebar text-sidebar-foreground",
185:             className
186:         )}
187:         ref={ref}
188:         {...props}
189:     >
190:         {children}
191:     </div>
192: )
193: }
194:
195: if (isMobile) {
196:     return (
197:         <Sheet open={openMobile} onOpenChange={setOpenMobile} {...props}>
198:             <SheetContent
199:                 data-sidebar="sidebar"
200:                 data-mobile="true"
201:                 className="w-[--sidebar-width] bg-sidebar p-0 text-sidebar-foreground [&button]:hidden"
202:                 style={
203:                     {
204:                         "--sidebar-width": SIDEBAR_WIDTH_MOBILE,
205:                     } as React.CSSProperties
206:                 }
207:                 side={side}
208:             >
209:                 <div className="flex h-full w-full flex-col">{children}</div>
210:             </SheetContent>
211:         </Sheet>
212:     )
213: }
214:
215: return (
216:     <div
217:         ref={ref}
218:         className="group peer hidden md:block text-sidebar-foreground"
219:         data-state={state}
220:         data-collapsible={state === "collapsed" ? collapsible : ""}
221:         data-variant={variant}
222:         data-side={side}
223:     >
224:         {/* This is what handles the sidebar gap on desktop */}
225:         <div
226:             className={cn(
227:                 "duration-200 relative h-svh w-[--sidebar-width] bg-transparent transition-[width] ease-linear"
228:                 "group-data-[collapsible=offcanvas]:w-0",
229:                 "group-data-[side=right]:rotate-180",
230:                 variant === "floating" || variant === "inset"
231:                 ? "group-data-[collapsible=icon]:w-[calc(var(--sidebar-width-icon)+_theme(spacing.4))]"
232:                 : "group-data-[collapsible=icon]:w-[--sidebar-width-icon]"
233:             )}
234:         />
235:         <div
236:             className={cn(
237:                 "duration-200 fixed inset-y-0 z-10 hidden h-svh w-[--sidebar-width] transition-[left,right,width]"
238:                 side === "left"
239:                 ? "left-0 group-data-[collapsible=offcanvas]:left-[calc(var(--sidebar-width)*-1)]"
240:                 : "right-0 group-data-[collapsible=offcanvas]:right-[calc(var(--sidebar-width)*-1)]",
241:                 // Adjust the padding for floating and inset variants.
242:                 variant === "floating" || variant === "inset"
243:                 ? "p-2 group-data-[collapsible=icon]:w-[calc(var(--sidebar-width-icon)+_theme(spacing.4)+2px)]"
244:                 : "group-data-[collapsible=icon]:w-[--sidebar-width-icon] group-data-[side=left]:border-r group-data-[side=right]:border-l"
245:             )}
246:         >
247:             {...props}
248:         >
249:             <div
250:                 data-sidebar="sidebar"
251:                 className="flex h-full w-full flex-col bg-sidebar group-data-[variant=floating]:rounded-lg group-data-[variant=floating]:border group-data-[variant=floating]:shadow"
252:             >
253:                 {children}
254:             </div>
255:         </div>

```

```

256:         </div>
257:     )
258: }
259: )
260: Sidebar.displayName = "Sidebar"
261:
262: const SidebarTrigger = React.forwardRef<
263:   React.ElementRef<typeof Button>,
264:   React.ComponentProps<typeof Button>
265: >(({ className, onClick, ...props }, ref) => {
266:   const { toggleSidebar } = useSidebar()
267:
268:   return (
269:     <Button
270:       ref={ref}
271:       data-sidebar="trigger"
272:       variant="ghost"
273:       size="icon"
274:       className={cn("h-7 w-7", className)}
275:       onClick={(event) => {
276:         onClick?.(event)
277:         toggleSidebar()
278:       }}
279:       {...props}
280:     >
281:       <PanelLeft />
282:       <span className="sr-only">Toggle Sidebar</span>
283:     </Button>
284:   )
285: })
286: SidebarTrigger.displayName = "SidebarTrigger"
287:
288: const SidebarRail = React.forwardRef<
289:   HTMLButtonElement,
290:   React.ComponentProps<"button">
291: >(({ className, ...props }, ref) => {
292:   const { toggleSidebar } = useSidebar()
293:
294:   return (
295:     <button
296:       ref={ref}
297:       data-sidebar="rail"
298:       aria-label="Toggle Sidebar"
299:       tabIndex={-1}
300:       onClick={toggleSidebar}
301:       title="Toggle Sidebar"
302:       className={cn(
303:         "absolute inset-y-0 z-20 hidden w-4 -translate-x-1/2 transition-all ease-linear after:absolute after:
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-ful
307:         "[[data-side=left][data-collapsible=offcanvas]&]:-right-2",
308:         "[[data-side=right][data-collapsible=offcanvas]&]:-left-2",
309:         className
310:       )}
311:       {...props}
312:     />
313:   )
314: })
315: SidebarRail.displayName = "SidebarRail"
316:
317: const SidebarInset = React.forwardRef<
318:   HTMLDivElement,
319:   React.ComponentProps<"main">
320: >(({ className, ...props }, ref) => {
321:   return (
322:     <main
323:       ref={ref}
324:       className={cn(
325:         "relative flex min-h-svh flex-1 flex-col bg-background",
326:         "peer-data-[variant=inset]:min-h-[calc(100svh-theme(spacing.4))] md:peer-data-[variant=inset]:m-2 m
327:         className
328:       )}

```

```

329:     {...props}
330:   />
331: )
332: })
333: SidebarInset.displayName = "SidebarInset"
334:
335: const SidebarInput = React.forwardRef<
336:   React.ElementRef<typeof Input>,
337:   React.ComponentProps<typeof Input>
338: >(({ className, ...props }, ref) => {
339:   return (
340:     <Input
341:       ref={ref}
342:       data-sidebar="input"
343:       className={cn(
344:         "h-8 w-full bg-background shadow-none focus-visible:ring-2 focus-visible:ring-sidebar-ring",
345:         className
346:       )}
347:       {...props}
348:     />
349:   )
350: })
351: SidebarInput.displayName = "SidebarInput"
352:
353: const SidebarHeader = React.forwardRef<
354:   HTMLDivElement,
355:   React.ComponentProps<"div">
356: >(({ className, ...props }, ref) => {
357:   return (
358:     <div
359:       ref={ref}
360:       data-sidebar="header"
361:       className={cn("flex flex-col gap-2 p-2", className)}
362:       {...props}
363:     />
364:   )
365: })
366: SidebarHeader.displayName = "SidebarHeader"
367:
368: const SidebarFooter = React.forwardRef<
369:   HTMLDivElement,
370:   React.ComponentProps<"div">
371: >(({ className, ...props }, ref) => {
372:   return (
373:     <div
374:       ref={ref}
375:       data-sidebar="footer"
376:       className={cn("flex flex-col gap-2 p-2", className)}
377:       {...props}
378:     />
379:   )
380: })
381: SidebarFooter.displayName = "SidebarFooter"
382:
383: const SidebarSeparator = React.forwardRef<
384:   React.ElementRef<typeof Separator>,
385:   React.ComponentProps<typeof Separator>
386: >(({ className, ...props }, ref) => {
387:   return (
388:     <Separator
389:       ref={ref}
390:       data-sidebar="separator"
391:       className={cn("mx-2 w-auto bg-sidebar-border", className)}
392:       {...props}
393:     />
394:   )
395: })
396: SidebarSeparator.displayName = "SidebarSeparator"
397:
398: const SidebarContent = React.forwardRef<
399:   HTMLDivElement,
400:   React.ComponentProps<"div">
401: >(({ className, ...props }, ref) => {

```

```

402:   return (
403:     <div
404:       ref={ref}
405:       data-sidebar="content"
406:       className={cn(
407:         "flex min-h-0 flex-1 flex-col gap-2 overflow-auto group-data-[collapsible=icon]:overflow-hidden",
408:         className
409:       )}
410:       {...props}
411:     />
412:   )
413: })
414: SidebarContent.displayName = "SidebarContent"
415:
416: const SidebarGroup = React.forwardRef<
417:   HTMLDivElement,
418:   React.ComponentProps<"div">
419: >(({ className, ...props }, ref) => {
420:   return (
421:     <div
422:       ref={ref}
423:       data-sidebar="group"
424:       className={cn("relative flex w-full min-w-0 flex-col p-2", className)}
425:       {...props}
426:     />
427:   )
428: })
429: SidebarGroup.displayName = "SidebarGroup"
430:
431: const SidebarGroupLabel = React.forwardRef<
432:   HTMLDivElement,
433:   React.ComponentProps<"div"> & { asChild?: boolean }
434: >(({ className, asChild = false, ...props }, ref) => {
435:   const Comp = asChild ? Slot : "div"
436:
437:   return (
438:     <Comp
439:       ref={ref}
440:       data-sidebar="group-label"
441:       className={cn(
442:         "duration-200 flex h-8 shrink-0 items-center rounded-md px-2 text-xs font-medium text-sidebar-foreground",
443:         "group-data-[collapsible=icon]:-mt-8 group-data-[collapsible=icon]:opacity-0",
444:         className
445:       )}
446:       {...props}
447:     />
448:   )
449: })
450: SidebarGroupLabel.displayName = "SidebarGroupLabel"
451:
452: const SidebarGroupAction = React.forwardRef<
453:   HTMLButtonElement,
454:   React.ComponentProps<"button"> & { asChild?: boolean }
455: >(({ className, asChild = false, ...props }, ref) => {
456:   const Comp = asChild ? Slot : "button"
457:
458:   return (
459:     <Comp
460:       ref={ref}
461:       data-sidebar="group-action"
462:       className={cn(
463:         "absolute right-3 top-3.5 flex aspect-square w-5 items-center justify-center rounded-md p-0 text-si",
464:         // Increases the hit area of the button on mobile.
465:         "after:absolute after:-inset-2 after:md:hidden",
466:         "group-data-[collapsible=icon]:hidden",
467:         className
468:       )}
469:       {...props}
470:     />
471:   )
472: })
473: SidebarGroupAction.displayName = "SidebarGroupAction"
474:

```

```

475: const SidebarGroupContent = React.forwardRef<
476:   HTMLDivElement,
477:   React.ComponentProps<"div">
478: >(({ className, ...props }, ref) => (
479:   <div
480:     ref={ref}
481:     data-sidebar="group-content"
482:     className={cn("w-full text-sm", className)}
483:     {...props}
484:   />
485: ))
486: SidebarGroupContent.displayName = "SidebarGroupContent"
487:
488: const SidebarMenu = React.forwardRef<
489:   HTMLUListElement,
490:   React.ComponentProps<"ul">
491: >(({ className, ...props }, ref) => (
492:   <ul
493:     ref={ref}
494:     data-sidebar="menu"
495:     className={cn("flex w-full min-w-0 flex-col gap-1", className)}
496:     {...props}
497:   />
498: ))
499: SidebarMenu.displayName = "SidebarMenu"
500:
501: const SidebarMenuItem = React.forwardRef<
502:   HTMLLIElement,
503:   React.ComponentProps<"li">
504: >(({ className, ...props }, ref) => (
505:   <li
506:     ref={ref}
507:     data-sidebar="menu-item"
508:     className={cn("group/menu-item relative", className)}
509:     {...props}
510:   />
511: ))
512: SidebarMenuItem.displayName = "SidebarMenuItem"
513:
514: const sidebarMenuButtonVariants = cva(
515:   "peer/menu-button flex w-full items-center gap-2 overflow-hidden rounded-md p-2 text-left text-sm outline",
516:   {
517:     variants: {
518:       variant: {
519:         default: "hover:bg-sidebar-accent hover:text-sidebar-accent-foreground",
520:         outline:
521:           "bg-background shadow-[0_0_1px_hsl(var(--sidebar-border))] hover:bg-sidebar-accent hover:text-s",
522:       },
523:       size: {
524:         default: "h-8 text-sm",
525:         sm: "h-7 text-xs",
526:         lg: "h-12 text-sm group-data-[collapsible=icon]:!p-0",
527:       },
528:     },
529:     defaultVariants: {
530:       variant: "default",
531:       size: "default",
532:     },
533:   }
534: )
535:
536: const SidebarMenuButton = React.forwardRef<
537:   HTMLButtonElement,
538:   React.ComponentProps<"button"> & {
539:     asChild?: boolean
540:     isActive?: boolean
541:     tooltip?: string | React.ComponentProps<typeof TooltipContent>
542:   } & VariantProps<typeof sidebarMenuButtonVariants>
543: >((
544:   {
545:     {
546:       asChild = false,
547:       isActive = false,

```

```

548:     variant = "default",
549:     size = "default",
550:     tooltip,
551:     className,
552:     ...props
553:   },
554:   ref
555: ) => {
556:   const Comp = asChild ? Slot : "button"
557:   const { isMobile, state } = useSidebar()
558:
559:   const button = (
560:     <Comp
561:       ref={ref}
562:       data-sidebar="menu-button"
563:       data-size={size}
564:       data-active={isActive}
565:       className={cn(sidebarMenuButtonVariants({ variant, size })), className)}
566:       {...props}
567:     />
568:   )
569:
570:   if (!tooltip) {
571:     return button
572:   }
573:
574:   if (typeof tooltip === "string") {
575:     tooltip = {
576:       children: tooltip,
577:     }
578:   }
579:
580:   return (
581:     <Tooltip>
582:       <TooltipTrigger asChild>{button}</TooltipTrigger>
583:       <TooltipContent
584:         side="right"
585:         align="center"
586:         hidden={state !== "collapsed" || isMobile}
587:         {...tooltip}
588:       />
589:     </Tooltip>
590:   )
591: }
592: )
593: SidebarMenuButton.displayName = "SidebarMenuButton"
594:
595: const SidebarMenuAction = React.forwardRef<
596:   HTMLButtonElement,
597:   React.ComponentProps<"button"> & {
598:     asChild?: boolean
599:     showOnHover?: boolean
600:   }
601: >(({ className, asChild = false, showOnHover = false, ...props }, ref) => {
602:   const Comp = asChild ? Slot : "button"
603:
604:   return (
605:     <Comp
606:       ref={ref}
607:       data-sidebar="menu-action"
608:       className={cn(
609:         "absolute right-1 top-1.5 flex aspect-square w-5 items-center justify-center rounded-md p-0 text-si
610:         // Increases the hit area of the button on mobile.
611:         "after:absolute after:-inset-2 after:md:hidden",
612:         "peer-data-[size=sm]/menu-button:top-1",
613:         "peer-data-[size=default]/menu-button:top-1.5",
614:         "peer-data-[size=lg]/menu-button:top-2.5",
615:         "group-data-[collapsible=icon]:hidden",
616:         showOnHover &&
617:         "group-focus-within/menu-item:opacity-100 group-hover/menu-item:opacity-100 data-[state=open]:opa
618:       className
619:     )}
620:     {...props}

```

```

621:     />
622:   )
623: })
624: SidebarMenuAction.displayName = "SidebarMenuAction"
625:
626: const SidebarMenuBadge = React.forwardRef<
627:   HTMLDivElement,
628:   React.ComponentProps<"div">
629: >(({ className, ...props }, ref) => (
630:   <div
631:     ref={ref}
632:     data-sidebar="menu-badge"
633:     className={cn(
634:       "absolute right-1 flex h-5 min-w-5 items-center justify-center rounded-md px-1 text-xs font-medium 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]:text-foreground group-[.toaster]:b
```

```

19:         description: "group-[.toast]:text-muted-foreground",
20:         actionButton:
21:             "group-[.toast]:bg-primary group-[.toast]:text-primary-foreground",
22:         cancelButton:
23:             "group-[.toast]:bg-muted group-[.toast]:text-muted-foreground",
24:     },
25: }}
26: {...props}
27: />
28: )
29: }
30:
31: export { Toaster }

```

■ File: components/ui/switch.tsx

```

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

```

■ File: components/ui/table.tsx

```

=====
1: import * as React from "react"
2:
3: import { cn } from "@/lib/utils"
4:
5: const Table = React.forwardRef<
6:   HTMLTableElement,
7:   React.HTMLAttributes<HTMLTableElement>
8: >(({ className, ...props }, ref) => (
9:   <div className="relative w-full overflow-auto">
10:     <table
11:       ref={ref}
12:       className={cn("w-full caption-bottom text-sm", className)}
13:       {...props}
14:     />
15:   </div>
16: ))
17: Table.displayName = "Table"
18:
19: const TableHeader = React.forwardRef<
20:   HTMLTableSectionElement,
21:   React.HTMLAttributes<HTMLTableSectionElement>

```

```

22: >(({ className, ...props }, ref) => (
23:   <thead ref={ref} className={cn("&_tr:border-b", className)} {...props} />
24: ))
25: TableHeader.displayName = "TableHeader"
26:
27: const TableBody = React.forwardRef<
28:   HTMLTableSectionElement,
29:   React.HTMLAttributes<HTMLTableSectionElement>
30: >(({ className, ...props }, ref) => (
31:   <tbody
32:     ref={ref}
33:     className={cn("&_tr:last-child:border-0", className)}
34:     {...props}
35:   />
36: ))
37: TableBody.displayName = "TableBody"
38:
39: const TableFooter = React.forwardRef<
40:   HTMLTableSectionElement,
41:   React.HTMLAttributes<HTMLTableSectionElement>
42: >(({ className, ...props }, ref) => (
43:   <tfoot
44:     ref={ref}
45:     className={cn(
46:       "border-t bg-muted/50 font-medium [&tr]:last:border-b-0",
47:       className
48:     )}
49:     {...props}
50:   />
51: ))
52: TableFooter.displayName = "TableFooter"
53:
54: const TableRow = React.forwardRef<
55:   HTMLTableRowElement,
56:   React.HTMLAttributes<HTMLTableRowElement>
57: >(({ className, ...props }, ref) => (
58:   <tr
59:     ref={ref}
60:     className={cn(
61:       "border-b transition-colors hover:bg-muted/50 data-[state=selected]:bg-muted",
62:       className
63:     )}
64:     {...props}
65:   />
66: ))
67: TableRow.displayName = "TableRow"
68:
69: const TableHead = React.forwardRef<
70:   HTMLTableCellElement,
71:   React.ThHTMLAttributes<HTMLTableCellElement>
72: >(({ className, ...props }, ref) => (
73:   <th
74:     ref={ref}
75:     className={cn(
76:       "h-12 px-4 text-left align-middle font-medium text-muted-foreground [&:has([role=checkbox])]:pr-0",
77:       className
78:     )}
79:     {...props}
80:   />
81: ))
82: TableHead.displayName = "TableHead"
83:
84: const TableCell = React.forwardRef<
85:   HTMLTableCellElement,
86:   React.TdHTMLAttributes<HTMLTableCellElement>
87: >(({ className, ...props }, ref) => (
88:   <td
89:     ref={ref}
90:     className={cn("p-4 align-middle [&:has([role=checkbox])]:pr-0", className)}
91:     {...props}
92:   />
93: ))
94: TableCell.displayName = "TableCell"

```

```

95:
96: const TableCaption = React.forwardRef<
97:   HTMLTableCaptionElement,
98:   React.HTMLAttributes<HTMLTableCaptionElement>
99: >(({ className, ...props }, ref) => (
100:   <caption
101:     ref={ref}
102:     className={cn("mt-4 text-sm text-muted-foreground", className)}
103:     {...props}
104:   />
105: ))
106: TableCaption.displayName = "TableCaption"
107:
108: export {
109:   Table,
110:   TableHeader,
111:   TableBody,
112:   TableFooter,
113:   TableHead,
114:   TableRow,
115:   TableCell,
116:   TableCaption,
117: }

```

■ File: components/ui/tabs.tsx

```

=====
1: "use client"
2:
3: import * as React from "react"
4: import * as TabsPrimitive from "@radix-ui/react-tabs"
5:
6: import { cn } from "@lib/utils"
7:
8: const Tabs = TabsPrimitive.Root
9:
10: const TabsList = React.forwardRef<
11:   React.ElementRef<typeof TabsPrimitive.List>,
12:   React.ComponentPropsWithoutRef<typeof TabsPrimitive.List>
13: >(({ className, ...props }, ref) => (
14:   <TabsPrimitive.List
15:     ref={ref}
16:     className={cn(
17:       "inline-flex h-10 items-center justify-center rounded-md bg-muted p-1 text-muted-foreground",
18:       className
19:     )}
20:     {...props}
21:   />
22: ))
23: TabsList.displayName = TabsPrimitive.List.displayName
24:
25: const TabsTrigger = React.forwardRef<
26:   React.ElementRef<typeof TabsPrimitive.Trigger>,
27:   React.ComponentPropsWithoutRef<typeof TabsPrimitive.Trigger>
28: >(({ className, ...props }, ref) => (
29:   <TabsPrimitive.Trigger
30:     ref={ref}
31:     className={cn(
32:       "inline-flex items-center justify-center whitespace-nowrap rounded-sm px-3 py-1.5 text-sm font-medium",
33:       className
34:     )}
35:     {...props}
36:   />
37: ))
38: TabsTrigger.displayName = TabsPrimitive.Trigger.displayName
39:
40: const TabsContent = React.forwardRef<
41:   React.ElementRef<typeof TabsPrimitive.Content>,
42:   React.ComponentPropsWithoutRef<typeof TabsPrimitive.Content>
43: >(({ className, ...props }, ref) => (
44:   <TabsPrimitive.Content
45:     ref={ref}

```

```

46:     className={cn(
47:       "mt-2 ring-offset-background focus-visible:outline-none focus-visible:ring-2 focus-visible:ring-ring",
48:       className
49:     )}
50:     {...props}
51:   />
52: ))
53: TabsContent.displayName = TabsPrimitive.Content.displayName
54:
55: export { Tabs, TabsList, TabsTrigger, TabsContent }

```

■ File: components\ui\textarea.tsx

```

1: import * as React from "react"
2:
3: import { cn } from "@/lib/utils"
4:
5: const Textarea = React.forwardRef<
6:   HTMLTextAreaElement,
7:   React.ComponentProps<"textarea">
8: >(({ className, ...props }, ref) => {
9:   return (
10:     <textarea
11:       className={cn(
12:         "flex min-h-[80px] w-full rounded-md border border-input bg-background px-3 py-2 text-base ring-off",
13:         className
14:       )}
15:       ref={ref}
16:       {...props}
17:     />
18:   )
19: })
20: Textarea.displayName = "Textarea"
21:
22: 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 { X } from "lucide-react"
7:
8: import { cn } from "@/lib/utils"
9:
10: const ToastProvider = ToastPrimitives.Provider
11:
12: const ToastViewport = React.forwardRef<
13:   React.ElementRef<typeof ToastPrimitives.Viewport>,
14:   React.ComponentPropsWithoutRef<typeof ToastPrimitives.Viewport>
15: >(({ className, ...props }, ref) => (
16:   <ToastPrimitives.Viewport
17:     ref={ref}
18:     className={cn(
19:       "fixed top-0 z-[100] flex max-h-screen w-full flex-col-reverse p-4 sm:bottom-0 sm:right-0 sm:top-auto",
20:       className,
21:     )}
22:     {...props}
23:   />
24: ))
25: ToastViewport.displayName = ToastPrimitives.Viewport.displayName
26:
27: const toastVariants = cva(
28:   "group pointer-events-auto relative flex w-full items-center justify-between space-x-4 overflow-hidden rounded",
29:   {
30:     variants: {
31:       variant: {

```

```

32:         default: "border bg-background text-foreground",
33:         destructive: "destructive group border-destructive bg-destructive text-destructive-foreground",
34:         success: "border-green-500 bg-green-500 text-white", // Added success variant
35:         warning: "border-yellow-500 bg-yellow-500 text-white", // Added warning variant
36:     },
37: },
38:     defaultVariants: {
39:         variant: "default",
40:     },
41: },
42: )
43:
44: const Toast = React.forwardRef<
45:   React.ElementRef<typeof ToastPrimitives.Root>,
46:   React.ComponentPropsWithoutRef<typeof ToastPrimitives.Root> & VariantProps<typeof toastVariants>
47: >(({ className, variant, ...props }, ref) => {
48:   return <ToastPrimitives.Root ref={ref} className={cn(toastVariants({ variant }), className)} {...props} />
49: })
50: Toast.displayName = ToastPrimitives.Root.displayName
51:
52: const ToastAction = React.forwardRef<
53:   React.ElementRef<typeof ToastPrimitives.Action>,
54:   React.ComponentPropsWithoutRef<typeof ToastPrimitives.Action>
55: >(({ className, ...props }, ref) => (
56:   <ToastPrimitives.Action
57:     ref={ref}
58:     className={cn(
59:       "inline-flex h-8 shrink-0 items-center justify-center rounded-md border bg-transparent px-3 text-sm font-medium",
60:       className,
61:     )}
62:     {...props}
63:   />
64: ))
65: ToastAction.displayName = ToastPrimitives.Action.displayName
66:
67: const ToastClose = React.forwardRef<
68:   React.ElementRef<typeof ToastPrimitives.Close>,
69:   React.ComponentPropsWithoutRef<typeof ToastPrimitives.Close>
70: >(({ className, ...props }, ref) => (
71:   <ToastPrimitives.Close
72:     ref={ref}
73:     className={cn(
74:       "absolute right-1 top-1 rounded-md p-1 text-foreground/50 opacity-0 transition-opacity hover:text-foreground",
75:       className,
76:     )}
77:     toast-close=""
78:     {...props}
79:   >
80:     <X className="h-4 w-4" />
81:   </ToastPrimitives.Close>
82: ))
83: ToastClose.displayName = ToastPrimitives.Close.displayName
84:
85: const ToastTitle = React.forwardRef<
86:   React.ElementRef<typeof ToastPrimitives.Title>,
87:   React.ComponentPropsWithoutRef<typeof ToastPrimitives.Title>
88: >(({ className, ...props }, ref) => (
89:   <ToastPrimitives.Title ref={ref} className={cn("text-sm font-semibold", className)} {...props} />
90: ))
91: ToastTitle.displayName = ToastPrimitives.Title.displayName
92:
93: const ToastDescription = React.forwardRef<
94:   React.ElementRef<typeof ToastPrimitives.Description>,
95:   React.ComponentPropsWithoutRef<typeof ToastPrimitives.Description>
96: >(({ className, ...props }, ref) => (
97:   <ToastPrimitives.Description ref={ref} className={cn("text-sm opacity-90", className)} {...props} />
98: ))
99: ToastDescription.displayName = ToastPrimitives.Description.displayName
100:
101: type ToastProps = React.ComponentPropsWithoutRef<typeof Toast>
102:
103: type ToastActionElement = React.ReactElement<typeof ToastAction>
104:

```

```

105: export {
106:   type ToastProps,
107:   type ToastActionElement,
108:   ToastProvider,
109:   ToastViewport,
110:   Toast,
111:   ToastTitle,
112:   ToastDescription,
113:   ToastClose,
114:   ToastAction,
115: }

```

■ File: components\ui\toaster.tsx

```

1: "use client"
2:
3: import { Toast, ToastClose, ToastDescription, ToastProvider, ToastTitle, ToastViewport } from "@components"
4: import { useToast } from "@components/ui/use-toast"
5:
6: export function Toaster() {
7:   const { toasts } = useToast()
8:
9:   return (
10:     <ToastProvider>
11:       {toasts.map(({ id, title, description, action, ...props }) => (
12:         <Toast key={id} {...props}>
13:           <div className="grid gap-1">
14:             {title && <ToastTitle>{title}</ToastTitle>}
15:             {description && <ToastDescription>{description}</ToastDescription>}
16:           </div>
17:           {action}
18:         <ToastClose />
19:       </Toast>
20:     )}
21:     <ToastViewport />
22:   </ToastProvider>
23: )
24: }

```

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

```

1: "use client"
2:
3: import * as React from "react"
4: import * as ToggleGroupPrimitive from "@radix-ui/react-toggle-group"
5: import { type VariantProps } from "class-variance-authority"
6:
7: import { cn } from "@lib/Utils"
8: import { toggleVariants } from "@components/ui/toggle"
9:
10: const ToggleGroupContext = React.createContext<
11:   VariantProps<typeof toggleVariants>
12: >({
13:   size: "default",
14:   variant: "default",
15: })
16:
17: const ToggleGroup = React.forwardRef<
18:   React.ElementRef<typeof ToggleGroupPrimitive.Root>,
19:   React.ComponentPropsWithoutRef<typeof ToggleGroupPrimitive.Root> &
20:   VariantProps<typeof toggleVariants>
21: >(({ className, variant, size, children, ...props }, ref) => (
22:   <ToggleGroupPrimitive.Root
23:     ref={ref}
24:     className={cn("flex items-center justify-center gap-1", className)}
25:     {...props}
26:   >
27:     <ToggleGroupContext.Provider value={{ variant, size }}>
28:       {children}

```

```

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

```

■ File: components/ui/toggle.tsx

```

1: "use client"
2:
3: import * as React from "react"
4: import * as TogglePrimitive from "@radix-ui/react-toggle"
5: import { cva, type VariantProps } from "class-variance-authority"
6:
7: import { cn } from "@lib/Utils"
8:
9: const toggleVariants = cva(
10:   "inline-flex items-center justify-center rounded-md text-sm font-medium ring-offset-background transition-colors focus-visible:outline focus-visible:outline-2 focus-visible:outline-offset-2",
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:
5: import type React from "react"
6:
7: import { create } from "zustand"
8: import type { ToastProps } from "../toast"
9:
10: type Toast = ToastProps & {
11:   id: string
12:   title?: React.ReactNode
13:   description?: React.ReactNode
14:   action?: React.ReactNode
15: }
16:
17: const TOAST_LIMIT = 1
18: const TOAST_REMOVE_DELAY = 1000000 // Keep toasts on screen for a long time for demo purposes
19:
20: type State = {
21:   toasts: Toast[]
22: }
23:
24: type Actions = {
25:   addToast: (toast: Toast) => void
26:   updateToast: (id: string, toast: Partial<Toast>) => void
27:   dismissToast: (id?: string) => void
28:   removeToast: (id?: string) => void
29: }
30:
31: const toastStore = create<State & Actions>((set, get) => ({
32:   toasts: [],
33:   addToast: (toast) => {
34:     set((state) => ({
35:       toasts: [toast, ...state.toasts].slice(0, TOAST_LIMIT),
36:     }))
37:   },
38:   updateToast: (id, toast) => {
39:     set((state) => ({
40:       toasts: state.toasts.map((t) => (t.id === id ? { ...t, ...toast } : t)),
41:     }))
42:   },
43:   dismissToast: (id) => {
44:     set((state) => ({
45:       toasts: state.toasts.map((t) => (t.id === id || id === undefined ? { ...t, open: false } : t)),
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>) => toastStore.getState().updateToast(id, newOpts)
68:   const dismiss = () => toastStore.getState().dismissToast(id)
69:
70:   toastStore.getState().addToast({
71:     id,
```

```

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

```

■ 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: libutils.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": "my-v0-project",
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: },
11: "dependencies": {
12:   "@hello-pangea/dnd": "latest",
13:   "@hookform/resolvers": "^3.9.1",
14:   "@radix-ui/react-accordion": "1.2.2",
15:   "@radix-ui/react-alert-dialog": "1.1.4",
16:   "@radix-ui/react-aspect-ratio": "1.1.1",
17:   "@radix-ui/react-avatar": "1.1.2",
18:   "@radix-ui/react-checkbox": "1.1.3",
19:   "@radix-ui/react-collapsible": "1.1.2",
20:   "@radix-ui/react-context-menu": "2.2.4",
21:   "@radix-ui/react-dialog": "1.1.4",
22:   "@radix-ui/react-dropdown-menu": "2.1.4",
23:   "@radix-ui/react-hover-card": "1.1.4",
24:   "@radix-ui/react-label": "2.1.1",
25:   "@radix-ui/react-menubar": "1.1.4",
26:   "@radix-ui/react-navigation-menu": "1.2.3",
27:   "@radix-ui/react-popover": "1.1.4",
28:   "@radix-ui/react-progress": "1.1.1",
29:   "@radix-ui/react-radio-group": "1.2.2",
30:   "@radix-ui/react-scroll-area": "1.2.2",
31:   "@radix-ui/react-select": "2.1.4",
32:   "@radix-ui/react-separator": "1.1.1",
33:   "@radix-ui/react-slider": "1.2.2",
34:   "@radix-ui/react-slot": "1.1.1",
35:   "@radix-ui/react-switch": "1.1.2",
36:   "@radix-ui/react-tabs": "1.1.2",
37:   "@radix-ui/react-toast": "latest",
38:   "@radix-ui/react-toggle": "1.1.1",
39:   "@radix-ui/react-toggle-group": "1.1.1",
40:   "@radix-ui/react-tooltip": "1.1.6",
41:   "autoprefixer": "^10.4.20",
42:   "class-variance-authority": "^0.7.1",
43:   "clsx": "^2.1.1",
44:   "cmdk": "1.0.4",
45:   "date-fns": "4.1.0",
46:   "embla-carousel-react": "8.5.1",
47:   "fs": "latest",
48:   "immer": "latest",
49:   "input-otp": "1.4.1",
50:   "lucide-react": "^0.454.0",
51:   "next": "15.2.4",
52:   "next-themes": "latest",
53:   "path": "latest",
54:   "react": "^19",
55:   "react-day-picker": "8.10.1",
56:   "react-dom": "^19",
57:   "react-hook-form": "^7.54.1",
58:   "react-resizable-panels": "^2.1.7",
59:   "recharts": "2.15.0",
60:   "sonner": "^1.7.1",
61:   "tailwind-merge": "^2.5.5",
62:   "tailwindcss-animate": "^1.0.7",
63:   "use-sync-external-store": "latest",
64:   "uuid": "latest",
65:   "vaul": "^0.9.6",
66:   "zod": "^3.24.1",
67:   "zustand": "latest"
68: },
69: "devDependencies": {
70:   "@types/node": "^22",
71:   "@types/react": "^19",
72:   "@types/react-dom": "^19",
73:   "postcss": "^8.5",
74:   "tailwindcss": "^3.4.17",
75:   "typescript": "^5"
76: }
77: }

```

■ 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 PatientInfo {
2:   name: string;
3:   gender: string;
4:   birthDate: string;
5:   sampleCode: string;
6:   sampleDate: string;
7:   reportDate: string;
8:   reportAuth: string;
9:   genomicAnalytics: string;

```

```

10:   checkedBy: string;
11:   scientificContent: string;
12:   disclaimer: string;
13:
14: }
15:
16: export interface ReportContent {
17:   introduction: string;
18:   genomicsExplanation: string;
19:   genesHealthImpact: string;
20:   fundamentalsPRS: string;
21:   utilityDoctors: string;
22:   microarrayExplanation: string;
23:   microarrayData: string;
24: }
25:
26: export interface ReportSettings {
27:   title: string;
28:   subtitle: string;
29:   companyName: string;
30:   headerColor: string;
31:   accentColor: string;
32:   fonts: {
33:     primary: string;
34:     secondary: string;
35:     mono: string;
36:   };
37: }
38:
39: // NEW: Interface for dynamically defined diet analysis fields
40: export interface DynamicDietFieldDefinition {
41:   _uuid: string; // For DND library internal use
42:   id: string; // Unique identifier for the field (e.g., "carbohydrate_sensitivity")
43:   label: string; // Display name (e.g., "Carbohydrate Sensitivity")
44:   category: string; // Category (e.g., "Macronutrients", "Food Sensitivities")
45:   min: number; // Min score for the field
46:   max: number; // Max score for the field
47:   highRecommendation: string; // Recommendation for high score
48:   normalRecommendation: string; // Recommendation for normal score
49:   lowRecommendation: string; // Recommendation for low score
50: }
51:
52: // NEW: Interface for a patient's specific result for a dynamic diet field
53: export interface PatientDietAnalysisResult {
54:   fieldId: string; // References DynamicDietFieldDefinition.id
55:   score: number;
56:   level: string; // e.g., "HIGH", "NORMAL", "LOW", "SUITABLE"
57:   recommendation: string; // Patient-specific recommendation
58: }
59:
60: // OLD DietAnalysis interface is removed.
61: // The data it contained will now be managed via DynamicDietFieldDefinition and PatientDietAnalysisResult.
62:
63: export interface NutrientData {
64:   score: number;
65:   healthImpact: string;
66:   intakeLevel: string;
67:   source: string;
68: }
69:
70: export interface NutritionData {
71:   vitamins: {
72:     vitaminA: NutrientData;
73:     vitaminB12: NutrientData;
74:     vitaminB2: NutrientData;
75:     vitaminB6: NutrientData;
76:     vitaminB9: NutrientData;
77:     vitaminC: NutrientData;
78:     vitaminD: NutrientData;
79:     vitaminE: NutrientData;
80:   };
81:   fattyAcids: {
82:     omegaFattyAcids: NutrientData;

```

```

83:     };
84:     elements: {
85:         calcium: NutrientData;
86:         iodine: NutrientData;
87:         iron: NutrientData;
88:         magnesium: NutrientData;
89:         phosphorus: NutrientData;
90:         selenium: NutrientData;
91:         zinc: NutrientData;
92:     };
93:     complexNutrients: {
94:         antiInflammatory: NutrientData;
95:         antioxidants: NutrientData;
96:     };
97: }
98:
99: export interface ExerciseData {
100:     level: string;
101:     description: string;
102: }
103:
104: export interface SportsAndFitness {
105:     exerciseType: {
106:         endurancePotential: ExerciseData;
107:         powerPotential: ExerciseData;
108:         strengthProfile: ExerciseData;
109:         exerciseTime: ExerciseData;
110:     };
111:     performance: {
112:         enduranceCapacity: ExerciseData;
113:         fatigueResistance: ExerciseData;
114:         bloodFlow: ExerciseData;
115:         oxygenEfficiency: ExerciseData;
116:         waterLoss: ExerciseData;
117:         lactateClearing: ExerciseData;
118:         recoveryEfficiency: ExerciseData;
119:         bodyComposition: ExerciseData;
120:         injuryRisk: ExerciseData;
121:         achillesTendonInjury: ExerciseData;
122:         ligamentInjury: ExerciseData;
123:         rupture: ExerciseData;
124:         muscleCramps: ExerciseData;
125:     };
126: }
127:
128: export interface HealthConditionStatus {
129:     status: "strength" | "improvement";
130: }
131:
132: export interface LifestyleConditions {
133:     heartVascularHealth: {
134:         hypertension: HealthConditionStatus;
135:         leftVentricularHypertrophy: HealthConditionStatus;
136:         atherosclerosis: HealthConditionStatus;
137:         familialHypercholesterolemia: HealthConditionStatus;
138:         cardiomyopathy: HealthConditionStatus;
139:         cholesterolImbalance: HealthConditionStatus;
140:         stroke: HealthConditionStatus;
141:     };
142:     diabetes: {
143:         obesity: HealthConditionStatus;
144:         diabetesType2: HealthConditionStatus;
145:     };
146:     liverHealth: {
147:         nafld: HealthConditionStatus;
148:         liverCirrhosis: HealthConditionStatus;
149:     };
150:     boneHealth: {
151:         osteoporosis: HealthConditionStatus;
152:         osteoarthritis: HealthConditionStatus;
153:     };
154:     gutHealth: {
155:         irritableBowelSyndrome: HealthConditionStatus;

```

```

156:     celiacDisease: HealthConditionStatus;
157: };
158: hormoneSystem: {
159:     hypothyroidism: HealthConditionStatus;
160: };
161: }
162:
163: export interface MetabolicGeneData {
164:     genes: string[];
165:     genotype: string;
166:     impact: string;
167:     advice: string;
168: }
169:
170: export interface MetabolicCore {
171:     methylation: MetabolicGeneData;
172:     inflammation: MetabolicGeneData;
173:     oxidativeStress: MetabolicGeneData;
174:     detoxification: MetabolicGeneData;
175:     vitaminDMetabolism: MetabolicGeneData;
176:     lipidMetabolism: MetabolicGeneData;
177:     insulinResistance: MetabolicGeneData;
178:     circadianRhythm: MetabolicGeneData;
179:     exerciseResponsiveness: MetabolicGeneData;
180:     snackingTendency: MetabolicGeneData;
181:     oestrogenElimination: MetabolicGeneData;
182:     powerVsEndurance: MetabolicGeneData;
183:     injuryRecovery: MetabolicGeneData;
184:     fluidLossTendency: MetabolicGeneData;
185:     vo2Max: MetabolicGeneData;
186: }
187:
188: export interface DigestiveHealthData {
189:     level: string;
190:     description: string;
191: }
192:
193: export interface DigestiveHealth {
194:     gutHealth: DigestiveHealthData;
195:     glutenIntolerance: DigestiveHealthData;
196:     celiacDiseaseTendency: DigestiveHealthData;
197:     impactOnBonesJoints: DigestiveHealthData;
198:     lactoseIntolerance: DigestiveHealthData;
199:     probioticsRequirement: DigestiveHealthData;
200: }
201:
202: export interface AddictionData {
203:     tendency: string;
204:     description: string;
205: }
206:
207: export interface GenesAndAddiction {
208:     alcoholAddiction: AddictionData;
209:     smokingAddiction: AddictionData;
210:     snackingHabits: AddictionData;
211:     altruism: AddictionData;
212: }
213:
214: export interface SleepData {
215:     impact: string;
216:     intervention: string;
217: }
218:
219: export interface SleepAndRest {
220:     sleepCycle: SleepData;
221:     sleepApnea: SleepData;
222:     sleepDepth: SleepData;
223:     stressSensitivity: SleepData;
224: }
225:
226: export interface AllergyData {
227:     tendency: string;
228: }

```

```

229:
230: export interface AllergiesAndSensitivity {
231:   dustAllergy: AllergyData;
232:   pesticideSensitivity: AllergyData;
233:   smokeSensitivity: AllergyData;
234:   automobileSmokeSensitivity: AllergyData;
235:   generalAdvice: string;
236: }
237:
238: export interface GeneTestResult {
239:   geneName: string;
240:   variation: string;
241:   result: string;
242: }
243:
244: export interface MetabolicSummary {
245:   strengths: string[];
246:   weaknesses: string[];
247: }
248:
249: export interface PreventiveHealth {
250:   diagnosticTests: {
251:     halfYearly: string[];
252:     yearly: string[];
253:   };
254:   nutritionalSupplements: Array<{
255:     supplement: string;
256:     needed: boolean;
257:   }>;
258: }
259:
260: export interface FamilyGeneticImpact {
261:   gene: string;
262:   normalAlleles: string;
263:   yourResult: string;
264:   healthImpact: string;
265: }
266:
267: export interface ReportSummaries {
268:   nutrigenomicsSummary: string;
269:   exerciseGenomicsSummary: string;
270: }
271:
272: export interface ComprehensiveReportData {
273:   patientInfo: PatientInfo;
274:   content: ReportContent;
275:   settings: ReportSettings;
276:   // NEW: Dynamic diet field definitions
277:   dynamicDietFieldDefinitions: DynamicDietFieldDefinition[];
278:   // NEW: Patient-specific results for dynamic diet fields
279:   patientDietAnalysisResults: PatientDietAnalysisResult[];
280:   nutritionData: NutritionData;
281:   sportsAndFitness: SportsAndFitness;
282:   lifestyleConditions: LifestyleConditions;
283:   metabolicCore: MetabolicCore;
284:   digestiveHealth: DigestiveHealth;
285:   genesAndAddiction: GenesAndAddiction;
286:   sleepAndRest: SleepAndRest;
287:   allergiesAndSensitivity: AllergiesAndSensitivity;
288:   geneTestResults: GeneTestResult[];
289:   metabolicSummary: MetabolicSummary;
290:   preventiveHealth: PreventiveHealth;
291:   familyGeneticImpact: FamilyGeneticImpact[];
292:   summaries: ReportSummaries;
293: }

```
