

■ Project - Frontend Code Export

■ Frontend JavaScript/JSX Files:

```
[JS] postcss.config.js
[JSX] src\App.jsx
[JS] src\api\axios.js
[JSX] src\components\auth\LoginForm.jsx
[JSX] src\components\auth\RegisterForm.jsx
[JSX] src\components\common\Button.jsx
[JSX] src\components\common\Card.jsx
[JSX] src\components\common\Input.jsx
[JSX] src\components\common\Loader.jsx
[JSX] src\components\common\Modal.jsx
[JSX] src\components\common\Toast.jsx
[JSX] src\components\diagnosis\ConfidenceGauge.jsx
[JSX] src\components\diagnosis\DiagnosisResult.jsx
[JSX] src\components\diagnosis\ImageUploader.jsx
[JSX] src\components\diagnosis\RecommendationCard.jsx
[JSX] src\components\history\SessionCard.jsx
[JSX] src\components\history\SessionList.jsx
[JSX] src\components\layout\Footer.jsx
[JSX] src\components\layout\Navbar.jsx
[JSX] src\contexts\AuthContext.jsx
[JSX] src\contexts\ToastContext.jsx
[JS] src\hooks\useAuth.js
[JS] src\hooks\useToast.js
[JSX] src\main.jsx
[JSX] src\pages\AdminDashboard.jsx
[JSX] src\pages\Dashboard.jsx
[JSX] src\pages\Home.jsx
[JSX] src\pages>Login.jsx
[JSX] src\pages\Profile.jsx
[JSX] src\pages\Register.jsx
[JS] src\services\api.js
[JS] src\services\authService.js
[JS] src\services\diagnosisService.js
[JS] src\services\historyService.js
[JS] src\services\systemService.js
[JS] src\utils\constants.js
[JS] src\utils\helpers.js
[JS] src\utils\pdfGenerator.js
[JS] src\utils\validators.js
[JS] tailwind.config.js
[JS] vite.config.js
```

■ File: postcss.config.js

```
=====
1: export default {
2:   plugins: {
3:     tailwindcss: {},
4:     autoprefixer: {},
5:   },
6: }
```

■ File: src\App.jsx

```
=====
1: // src/App.jsx - COMPLETE WITH ADMIN DASHBOARD
2: import { BrowserRouter as Router, Routes, Route, Navigate } from 'react-router-dom'
3: import { AuthProvider } from '../contexts/AuthContext'
4: import { ToastProvider } from '../contexts/ToastContext'
5: import { useAuth } from '../hooks/useAuth'
6:
7: // Pages
8: import Home from '../pages/Home'
9: import Login from '../pages/Login'
10: import Register from '../pages/Register'
11: import Dashboard from '../pages/Dashboard'
12: import Profile from '../pages/Profile'
13: import AdminDashboard from '../pages/AdminDashboard'
14:
15: // Protected Route Component
16: const ProtectedRoute = ({ children }) => {
17:   const { user, loading } = useAuth()
18:
19:   if (loading) {
20:     return (
21:       <div className="min-h-screen flex items-center justify-center bg-gradient-to-br from-primary-50 to-fo
22:       <div className="animate-spin rounded-full h-12 w-12 border-b-2 border-primary-600"></div>
23:     </div>
24:   )
25: }
26:
27: return user ? children : <Navigate to="/login" />
28: }
29:
30: function App() {
31:   return (
32:     <Router>
33:       <AuthProvider>
34:         <ToastProvider>
35:           <Routes>
36:             { /* Public route - NO LOGIN REQUIRED */ }
37:             <Route path="/" element={<Home />} />
38:
39:             { /* Auth routes */ }
40:             <Route path="/login" element={<Login />} />
41:             <Route path="/register" element={<Register />} />
42:
43:             { /* Protected routes - HISTORY TRACKING */ }
44:             <Route
45:               path="/dashboard"
46:               element={
47:                 <ProtectedRoute>
48:                   <Dashboard />
49:                 </ProtectedRoute>
50:               }
51:             />
52:             <Route
53:               path="/profile"
54:               element={
55:                 <ProtectedRoute>
56:                   <Profile />
57:                 </ProtectedRoute>
58:               }
59:             />
60:           </Routes>
61:         </AuthProvider>
62:       </Router>
63:     )
64:   }
65: }
```

```

61:         { /* Admin/System Monitoring - Protected */ }
62:         <Route
63:             path="/admin"
64:             element={
65:                 <ProtectedRoute>
66:                     <AdminDashboard />
67:                 </ProtectedRoute>
68:             }
69:         />
70:     </Routes>
71: </ToastProvider>
72: </AuthProvider>
73: </Router>
74: )
75: }
76:
77: export default App

```

■ File: src\api\axios.js

```

=====
1: // =====
2: // 1. src/api/axios.js - CORRECTED & ALIGNED
3: // =====
4: import axios from "axios";
5:
6: const API_BASE_URL =
7:   import.meta.env.VITE_API_URL || "http://localhost:5000/api";
8:
9: const api = axios.create({
10:   baseURL: API_BASE_URL,
11:   withCredentials: true, // CRITICAL: Send cookies for refresh token
12:   headers: {
13:     "Content-Type": "application/json",
14:   },
15:   timeout: 30000, // Match backend REQUEST_TIMEOUT
16: });
17:
18: // Request Interceptor - Add access token
19: api.interceptors.request.use(
20:   (config) => {
21:     const token = localStorage.getItem("accessToken");
22:     if (token) {
23:       config.headers.Authorization = `Bearer ${token}`;
24:     }
25:     return config;
26:   },
27:   (error) => Promise.reject(error)
28: );
29:
30: // Response Interceptor - Handle token refresh
31: api.interceptors.response.use(
32:   (response) => response,
33:   async (error) => {
34:     const originalRequest = error.config;
35:
36:     // Handle TOKEN_EXPIRED (match backend error code exactly)
37:     if (
38:       error.response?.status === 401 &&
39:       error.response?.data?.code === "TOKEN_EXPIRED" &&
40:       !originalRequest._retry
41:     ) {
42:       originalRequest._retry = true;
43:
44:       try {
45:         // Call refresh endpoint (uses httpOnly cookie)
46:         const { data } = await axios.post(
47:           `${API_BASE_URL}/auth/refresh`,
48:           {},
49:           { withCredentials: true }
50:         );
51:

```

```

52:         // Store new access token
53:         const newToken = data.data?.accessToken || data.accessToken;
54:         const expiresIn = data.data?.expiresIn || data.expiresIn || 900; // Default 15min
55:
56:         localStorage.setItem("accessToken", newToken);
57:         localStorage.setItem("tokenExpiry", Date.now() + expiresIn * 1000);
58:
59:         // Retry original request with new token
60:         originalRequest.headers.Authorization = `Bearer ${newToken}`;
61:         return api(originalRequest);
62:     } catch (refreshError) {
63:         // Refresh failed - logout user
64:         localStorage.removeItem("accessToken");
65:         localStorage.removeItem("tokenExpiry");
66:         window.location.href = "/login";
67:         return Promise.reject(refreshError);
68:     }
69: }
70:
71: return Promise.reject(error);
72: }
73: );
74:
75: export default api;

```

File: src\components\auth\LoginForm.jsx

```

=====
1: import { useState } from 'react'
2: import { Mail, Lock } from 'lucide-react'
3: import { useNavigate, Link } from 'react-router-dom'
4: import Input from '../common/Input'
5: import Button from '../common/Button'
6: import { useAuth } from '@/hooks/useAuth'
7: import { useToast } from '@/hooks/useToast'
8: import { validateEmail } from '@/utils/validators'
9:
10: const LoginForm = () => {
11:   const [formData, setFormData] = useState({ email: '', password: '' })
12:   const [errors, setErrors] = useState({})
13:   const [loading, setLoading] = useState(false)
14:   const { login } = useAuth()
15:   const { addToast } = useToast()
16:   const navigate = useNavigate()
17:
18:   const handleChange = (e) => {
19:     const { name, value } = e.target
20:     setFormData(prev => ({ ...prev, [name]: value }))
21:     if (errors[name]) {
22:       setErrors(prev => ({ ...prev, [name]: '' }))
23:     }
24:   }
25:
26:   const validate = () => {
27:     const newErrors = {}
28:     if (!formData.email) {
29:       newErrors.email = 'Email is required'
30:     } else if (!validateEmail(formData.email)) {
31:       newErrors.email = 'Invalid email address'
32:     }
33:     if (!formData.password) {
34:       newErrors.password = 'Password is required'
35:     }
36:     return newErrors
37:   }
38:
39:   const handleSubmit = async (e) => {
40:     e.preventDefault()
41:     const newErrors = validate()
42:
43:     if (Object.keys(newErrors).length > 0) {
44:       setErrors(newErrors)

```

```

45:     return
46:   }
47:
48:   setLoading(true)
49:   try {
50:     await login(formData.email, formData.password)
51:     addToast('Login successful!', 'success')
52:     navigate('/dashboard')
53:   } catch (error) {
54:     addToast(error.response?.data?.error || 'Login failed', 'error')
55:   } finally {
56:     setLoading(false)
57:   }
58: }
59:
60: return (
61:   <form onSubmit={handleSubmit} className="space-y-6">
62:     <Input
63:       label="Email Address"
64:       name="email"
65:       type="email"
66:       value={formData.email}
67:       onChange={handleChange}
68:       error={errors.email}
69:       icon={<Mail size={20} />}
70:       placeholder="your.email@example.com"
71:     />
72:
73:     <Input
74:       label="Password"
75:       name="password"
76:       type="password"
77:       value={formData.password}
78:       onChange={handleChange}
79:       error={errors.password}
80:       icon={<Lock size={20} />}
81:       placeholder="???????"
82:     />
83:
84:     <Button type="submit" loading={loading} className="w-full">
85:       Sign In
86:     </Button>
87:
88:     <p className="text-center text-sm text-gray-600">
89:       Don't have an account?{' '}
90:       <Link to="/register" className="text-primary-600 font-medium hover:text-primary-700 hover:underline">
91:         Sign up
92:       </Link>
93:     </p>
94:   </form>
95: )
96: }
97:
98: export default LoginForm

```

■ File: src/components/auth/RegisterForm.jsx

```

=====
1: import { useState } from 'react'
2: import { Mail, Lock, User as UserIcon } from 'lucide-react'
3: import { useNavigate, Link } from 'react-router-dom'
4: import Input from '../common/Input'
5: import Button from '../common/Button'
6: import { useAuth } from '@/hooks/useAuth'
7: import { useToast } from '@/hooks/useToast'
8: import { validateEmail, validatePassword, validateUsername, getPasswordStrength } from '@/utils/validators'
9:
10: const RegisterForm = () => {
11:   const [formData, setFormData] = useState({ username: '', email: '', password: '', confirmPassword: '' })
12:   const [errors, setErrors] = useState({})
13:   const [loading, setLoading] = useState(false)
14:   const { register } = useAuth()

```

```

15:   const { addToast } = useToast()
16:   const navigate = useNavigate()
17:
18:   const passwordStrength = formData.password ? getPasswordStrength(formData.password) : null
19:
20:   const handleChange = (e) => {
21:     const { name, value } = e.target
22:     setFormData(prev => ({ ...prev, [name]: value }))
23:     if (errors[name]) {
24:       setErrors(prev => ({ ...prev, [name]: '' }))
25:     }
26:   }
27:
28:   const validate = () => {
29:     const newErrors = {}
30:
31:     if (!formData.username) {
32:       newErrors.username = 'Username is required'
33:     } else if (!validateUsername(formData.username)) {
34:       newErrors.username = 'Username must be 3-30 characters (letters, numbers, underscore)'
35:     }
36:
37:     if (!formData.email) {
38:       newErrors.email = 'Email is required'
39:     } else if (!validateEmail(formData.email)) {
40:       newErrors.email = 'Invalid email address'
41:     }
42:
43:     if (!formData.password) {
44:       newErrors.password = 'Password is required'
45:     } else if (!validatePassword(formData.password)) {
46:       newErrors.password = 'Password must be 8+ chars with uppercase, lowercase, number and special char'
47:     }
48:
49:     if (formData.password !== formData.confirmPassword) {
50:       newErrors.confirmPassword = 'Passwords do not match'
51:     }
52:
53:     return newErrors
54:   }
55:
56:   const handleSubmit = async (e) => {
57:     e.preventDefault()
58:     const newErrors = validate()
59:
60:     if (Object.keys(newErrors).length > 0) {
61:       setErrors(newErrors)
62:       return
63:     }
64:
65:     setLoading(true)
66:     try {
67:       await register(formData.username, formData.email, formData.password)
68:       addToast('Registration successful! Please login.', 'success')
69:       navigate('/login')
70:     } catch (error) {
71:       addToast(error.response?.data?.error || 'Registration failed', 'error')
72:     } finally {
73:       setLoading(false)
74:     }
75:   }
76:
77:   return (
78:     <form onSubmit={handleSubmit} className="space-y-6">
79:       <Input
80:         label="Username"
81:         name="username"
82:         value={formData.username}
83:         onChange={handleChange}
84:         error={errors.username}
85:         icon={<UserIcon size={20} />}
86:         placeholder="johndoe"
87:       />

```

```

88:
89:     <Input
90:       label="Email Address"
91:       name="email"
92:       type="email"
93:       value={formData.email}
94:       onChange={handleChange}
95:       error={errors.email}
96:       icon={<Mail size={20} />}
97:       placeholder="your.email@example.com"
98:     />
99:
100:   <div>
101:     <Input
102:       label="Password"
103:       name="password"
104:       type="password"
105:       value={formData.password}
106:       onChange={handleChange}
107:       error={errors.password}
108:       icon={<Lock size={20} />}
109:       placeholder="???????"
110:     />
111:     {passwordStrength && (
112:       <div className="mt-2">
113:         <div className="flex items-center gap-2 mb-1">
114:           <div className="flex-1 bg-gray-200 rounded-full h-2 overflow-hidden">
115:             <div
116:               className={`h-full ${passwordStrength.color} transition-all duration-300`}
117:               style={{ width: passwordStrength.width }}
118:             />
119:           </div>
120:           <span className="text-xs font-medium text-gray-600">{passwordStrength.label}</span>
121:         </div>
122:       </div>
123:     )}
124:   </div>
125:
126:   <Input
127:     label="Confirm Password"
128:     name="confirmPassword"
129:     type="password"
130:     value={formData.confirmPassword}
131:     onChange={handleChange}
132:     error={errors.confirmPassword}
133:     icon={<Lock size={20} />}
134:     placeholder="???????"
135:   />
136:
137:   <Button type="submit" loading={loading} className="w-full">
138:     Create Account
139:   </Button>
140:
141:   <p className="text-center text-sm text-gray-600">
142:     Already have an account?{' '}
143:     <Link to="/login" className="text-primary-600 font-medium hover:text-primary-700 hover:underline">
144:       Sign in
145:     </Link>
146:   </p>
147: </form>
148: )
149: }
150:
151: export default RegisterForm

```

■ File: src\components\common\Button.jsx

```

=====
1: import clsx from 'clsx'
2: import { motion } from 'framer-motion'
3:
4: const Button = ({

```

```

5:   children,
6:   variant = 'primary',
7:   size = 'md',
8:   className = '',
9:   disabled = false,
10:  loading = false,
11:  icon = null,
12:  ...props
13: }) => {
14:   const baseClasses = 'font-medium rounded-xl transition-all duration-300 flex items-center justify-center
15:
16:   const variants = {
17:     primary: 'bg-gradient-to-r from-primary-600 to-forest-600 text-white shadow-lg hover:shadow-xl hover:sc
18:     secondary: 'bg-white text-primary-700 border-2 border-primary-200 hover:bg-primary-50 hover:border-prim
19:     outline: 'border-2 border-primary-500 text-primary-600 hover:bg-primary-50',
20:     ghost: 'text-primary-600 hover:bg-primary-50',
21:     danger: 'bg-red-600 text-white hover:bg-red-700 shadow-lg',
22:   }
23:
24:   const sizes = {
25:     sm: 'px-4 py-2 text-sm',
26:     md: 'px-6 py-3 text-base',
27:     lg: 'px-8 py-4 text-lg',
28:   }
29:
30:   return (
31:     <motion.button
32:       whileHover={{ scale: disabled ? 1 : 1.02 }}
33:       whileTap={{ scale: disabled ? 1 : 0.98 }}
34:       className={clsx(
35:         baseClasses,
36:         variants[variant],
37:         sizes[size],
38:         className
39:       )}
40:       disabled={disabled || loading}
41:       {...props}
42:     >
43:       {loading ? (
44:         <>
45:           <div className="animate-spin rounded-full h-5 w-5 border-b-2 border-current"></div>
46:           Loading...
47:         </>
48:       ) : (
49:         <>
50:           {icon}
51:           {children}
52:         </>
53:       )}
54:     </motion.button>
55:   )
56: }
57:
58: export default Button

```

File: src\components\common\Card.jsx

```

=====
1: import { motion } from 'framer-motion'
2: import clsx from 'clsx'
3:
4: const Card = ({
5:   children,
6:   className = '',
7:   bordered = false,
8:   hover = true,
9:   ...props
10: }) => {
11:   return (
12:     <motion.div
13:       initial={{ opacity: 0, y: 20 }}
14:       animate={{ opacity: 1, y: 0 }}

```



```

15:     transition={{ duration: 0.3 }}
16:     whileHover={hover ? { y: -4 } : {}}
17:     className={clsx(
18:       'card',
19:       bordered && 'card-bordered',
20:       className
21:     )}
22:     {...props}
23:   >
24:     {children}
25:   </motion.div>
26: )
27: }
28:
29: export default Card

```

■ File: src\components\common\Input.jsx

```

1: import clsx from 'clsx'
2: import { useState } from 'react'
3: import { Eye, EyeOff } from 'lucide-react'
4:
5: const Input = ({
6:   label,
7:   error,
8:   type = 'text',
9:   icon = null,
10:  helperText = null,
11:  className = '',
12:  ...props
13: }) => {
14:   const [showPassword, setShowPassword] = useState(false)
15:   const isPassword = type === 'password'
16:   const inputType = isPassword && showPassword ? 'text' : type
17:
18:   return (
19:     <div className={clsx('space-y-2', className)}>
20:       <label && (
21:         <label className="block text-sm font-medium text-gray-700">
22:           {label}
23:         </label>
24:       )>
25:       <div className="relative">
26:         <div && (
27:           <div className="absolute left-3 top-1/2 -translate-y-1/2 text-gray-400">
28:             {icon}
29:           </div>
30:         )>
31:         <input
32:           type={inputType}
33:           className={clsx(
34:             'input-field',
35:             icon && 'pl-10',
36:             isPassword && 'pr-10',
37:             error && 'border-red-500 focus:border-red-500 focus:ring-red-100'
38:           )}
39:           {...props}
40:         />
41:         <div && (
42:           <button
43:             type="button"
44:             onClick={() => setShowPassword(!showPassword)}
45:             className="absolute right-3 top-1/2 -translate-y-1/2 text-gray-400 hover:text-gray-600"
46:           >
47:             {showPassword ? <EyeOff size={20} /> : <Eye size={20} />}
48:           </button>
49:         )>
50:       </div>
51:       <div && (
52:         <p className="text-sm text-red-600 animate-slide-down">{error}</p>
53:       )>

```

```

54:         {helperText && !error && (
55:             <p className="text-sm text-gray-500">{helperText}</p>
56:         )}
57:     </div>
58: )
59: }
60:
61: export default Input

```

■ File: src\components\common\Loader.jsx

```

=====
1: import { motion } from 'framer-motion'
2:
3: const Loader = ({ size = 'md', fullscreen = false }) => {
4:     const sizes = {
5:         sm: 'h-8 w-8',
6:         md: 'h-12 w-12',
7:         lg: 'h-16 w-16',
8:     }
9:
10:    const loader = (
11:        <div className="flex flex-col items-center gap-4">
12:            <motion.div
13:                animate={{
14:                    rotate: 360,
15:                }}
16:                transition={{
17:                    duration: 1,
18:                    repeat: Infinity,
19:                    ease: 'linear',
20:                }}
21:                className={` ${sizes[size]} border-4 border-primary-200 border-t-primary-600 rounded-full` }
22:            />
23:            <p className="text-gray-600 font-medium">Loading...</p>
24:        </div>
25:    )
26:
27:    if (fullscreen) {
28:        return (
29:            <div className="min-h-screen flex items-center justify-center bg-gradient-to-br from-primary-50 to-fo
30:                {loader}
31:            </div>
32:        )
33:    }
34:
35:    return <div className="flex items-center justify-center p-8">{loader}</div>
36: }
37:
38: export default Loader

```

■ File: src\components\common\Modal.jsx

```

=====
1: import { motion, AnimatePresence } from 'framer-motion'
2: import { X } from 'lucide-react'
3: import { useEffect } from 'react'
4:
5: const Modal = ({ isOpen, onClose, title, children, size = 'md' }) => {
6:     useEffect(() => {
7:         if (isOpen) {
8:             document.body.style.overflow = 'hidden'
9:         } else {
10:            document.body.style.overflow = 'unset'
11:        }
12:        return () => {
13:            document.body.style.overflow = 'unset'
14:        }
15:    }, [isOpen])
16:
17:    const sizes = {

```

```

18:     sm: 'max-w-md',
19:     md: 'max-w-2xl',
20:     lg: 'max-w-4xl',
21:     full: 'max-w-7xl',
22:   }
23:
24:   return (
25:     <AnimatePresence>
26:       {isOpen && (
27:         <>
28:           <motion.div
29:             initial={{ opacity: 0 }}
30:             animate={{ opacity: 1 }}
31:             exit={{ opacity: 0 }}
32:             className="fixed inset-0 bg-black/50 backdrop-blur-sm z-40"
33:             onClick={onClose}
34:           />
35:           <div className="fixed inset-0 z-50 overflow-y-auto">
36:             <div className="flex min-h-full items-center justify-center p-4">
37:               <motion.div
38:                 initial={{ opacity: 0, scale: 0.95, y: 20 }}
39:                 animate={{ opacity: 1, scale: 1, y: 0 }}
40:                 exit={{ opacity: 0, scale: 0.95, y: 20 }}
41:                 className={`relative bg-white rounded-2xl shadow-xl w-full ${sizes[size]} p-6`}
42:                 onClick={e => e.stopPropagation()}
43:               >
44:                 <div className="flex items-center justify-between mb-4">
45:                   <h2 className="text-2xl font-bold text-gray-900">{title}</h2>
46:                   <button
47:                     onClick={onClose}
48:                     className="text-gray-400 hover:text-gray-600 transition-colors p-2 rounded-lg hover:bg-gray-50"
49:                   >
50:                     <X size={24} />
51:                   </button>
52:                 </div>
53:                 {children}
54:               </motion.div>
55:             </div>
56:           </div>
57:         </>
58:       )}
59:     </AnimatePresence>
60:   )
61: }
62:
63: export default Modal

```

■ File: src\components\common\Toast.jsx

```

1: import { motion } from 'framer-motion'
2: import { CheckCircle, XCircle, AlertCircle, Info, X } from 'lucide-react'
3:
4: const Toast = ({ message, type = 'info', onClose }) => {
5:   const config = {
6:     success: {
7:       icon: <CheckCircle size={20} />,
8:       bg: 'bg-green-50',
9:       border: 'border-green-200',
10:      text: 'text-green-800',
11:      iconColor: 'text-green-600',
12:    },
13:    error: {
14:      icon: <XCircle size={20} />,
15:      bg: 'bg-red-50',
16:      border: 'border-red-200',
17:      text: 'text-red-800',
18:      iconColor: 'text-red-600',
19:    },
20:    warning: {
21:      icon: <AlertCircle size={20} />,
22:      bg: 'bg-yellow-50',

```

```

23:     border: 'border-yellow-200',
24:     text: 'text-yellow-800',
25:     iconColor: 'text-yellow-600',
26:   },
27:   info: {
28:     icon: <Info size={20} />,
29:     bg: 'bg-blue-50',
30:     border: 'border-blue-200',
31:     text: 'text-blue-800',
32:     iconColor: 'text-blue-600',
33:   },
34: }
35:
36: const { icon, bg, border, text, iconColor } = config[type]
37:
38: return (
39:   <motion.div
40:     initial={{ opacity: 0, x: 100 }}
41:     animate={{ opacity: 1, x: 0 }}
42:     exit={{ opacity: 0, x: 100 }}
43:     className={` ${bg} ${border} ${text} border-2 rounded-xl px-4 py-3 shadow-lg min-w-[300px] max-w-md`}
44:   >
45:     <div className="flex items-start gap-3">
46:       <div className={iconColor}>{icon}</div>
47:       <p className="flex-1 text-sm font-medium">{message}</p>
48:       <button
49:         onClick={onClose}
50:         className="text-gray-400 hover:text-gray-600 transition-colors"
51:       >
52:         <X size={16} />
53:       </button>
54:     </div>
55:   </motion.div>
56: )
57: }
58:
59: export default Toast

```

■ File: src\components\diagnosis\ConfidenceGauge.jsx

```

1: import { motion } from 'framer-motion'
2: import { getConfidenceLevel } from '@/utils/constants'
3:
4: const ConfidenceGauge = ({ confidence }) => {
5:   const percentage = (confidence * 100).toFixed(1)
6:   const level = getConfidenceLevel(confidence)
7:
8:   return (
9:     <div className="space-y-4">
10:       <div className="flex items-end gap-4">
11:         <div className="text-6xl font-bold bg-gradient-to-r from-primary-600 to-forest-600 bg-clip-text text-transparent">
12:           {percentage}%
13:         </div>
14:         <div className={`text-2xl font-semibold ${level.color} mb-2`} >
15:           {level.label}
16:         </div>
17:       </div>
18:
19:       <div className="relative h-6 bg-gray-200 rounded-full overflow-hidden">
20:         <motion.div
21:           initial={{ width: 0 }}
22:           animate={{ width: `${percentage}%` }}
23:           transition={{ duration: 1, ease: 'easeOut' }}
24:           className="absolute h-full bg-gradient-to-r from-primary-500 to-forest-600 rounded-full"
25:         />
26:       </div>
27:
28:       <div className="grid grid-cols-4 gap-2 text-xs">
29:         <div className="text-center">
30:           <div className="h-2 bg-red-500 rounded mb-1" />
31:           <span className="text-gray-600">Low</span>

```

```

32:         </div>
33:         <div className="text-center">
34:             <div className="h-2 bg-yellow-500 rounded mb-1" />
35:             <span className="text-gray-600">Moderate</span>
36:         </div>
37:         <div className="text-center">
38:             <div className="h-2 bg-green-500 rounded mb-1" />
39:             <span className="text-gray-600">High</span>
40:         </div>
41:         <div className="text-center">
42:             <div className="h-2 bg-green-700 rounded mb-1" />
43:             <span className="text-gray-600">Very High</span>
44:         </div>
45:     </div>
46: </div>
47: )
48: }
49:
50: export default ConfidenceGauge

```

■ File: src\components\diagnosis\DiagnosisResult.jsx

```

=====
1: import { Download, RotateCcw, AlertTriangle } from "lucide-react";
2: import { motion } from "framer-motion";
3: import Card from "../common/Card";
4: import Button from "../common/Button";
5: import ConfidenceGauge from "../ConfidenceGauge";
6: import RecommendationCard from "../RecommendationCard";
7: import { DISEASE_INFO } from "@/utils/constants";
8: import { generatePDFReport } from "@/utils/pdfGenerator";
9: import { useToast } from "@/hooks/useToast";
10: import { useAuth } from "@/hooks/useAuth";
11:
12: const DiagnosisResult = ({ result, imagePreview, onNewDiagnosis }) => {
13:     const { addToast } = useToast();
14:     const { user } = useAuth();
15:
16:     // CRITICAL FIX: Handle not_plant and fallback
17:     const diseaseInfo =
18:         DISEASE_INFO[result.class] ||
19:         DISEASE_INFO.not_plant ||
20:         DISEASE_INFO.Healthy;
21:     const isNotPlant = result.class === "not_plant";
22:
23:     const handleDownloadPDF = async () => {
24:         try {
25:             await generatePDFReport(result, imagePreview);
26:             addToast("Report downloaded successfully!", "success");
27:         } catch (error) {
28:             addToast("Failed to generate PDF", "error");
29:         }
30:     };
31:
32:     return (
33:         <motion.div
34:             initial={{ opacity: 0, y: 20 }}
35:             animate={{ opacity: 1, y: 0 }}
36:             className="space-y-6"
37:         >
38:             { /* SPECIAL WARNING for not_plant */ }
39:             { isNotPlant && (
40:                 <Card className="p-6 bg-gradient-to-r from-yellow-50 to-orange-50 border-2 border-yellow-300">
41:                     <div className="flex items-start gap-4">
42:                         <AlertTriangle
43:                             className="text-yellow-600 flex-shrink-0"
44:                             size={32}
45:                         />
46:                         <div>
47:                             <h3 className="text-xl font-bold text-yellow-900 mb-2">
48:                                 ?? Not a Plant Detected
49:                             </h3>

```

```

50:         <p className="text-yellow-800 text-lg">
51:             Our AI detected that this image does not contain a plant. Please
52:             upload a clear photo of plant leaves or affected areas for
53:             accurate diagnosis.
54:         </p>
55:     </div>
56: </div>
57: </Card>
58: )}
59:
60: { /* Action Buttons */}
61: <div className="flex flex-wrap gap-4 justify-center">
62:     {!isNotPlant && (
63:         <Button
64:             onClick={handleDownloadPDF}
65:             icon={<Download size={20} />}
66:             size="lg"
67:         >
68:             Download PDF Report
69:         </Button>
70:     )}
71:     <Button
72:         onClick={onNewDiagnosis}
73:         variant="secondary"
74:         icon={<RotateCcw size={20} />}
75:         size="lg"
76:     >
77:         {isNotPlant ? "Try Again with Plant Image" : "New Diagnosis"}
78:     </Button>
79: </div>
80:
81: { /* Main Result Card */}
82: <Card className={`p-6 ${diseaseInfo.borderColor} border-2`} >
83:     <div className="flex items-start gap-4">
84:         <div className="text-6xl">{diseaseInfo.icon}</div>
85:         <div className="flex-1">
86:             <h2 className={`text-3xl font-bold ${diseaseInfo.color} mb-2`} >
87:                 {diseaseInfo.title}
88:             </h2>
89:             <p className="text-gray-700 text-lg mb-4">
90:                 {diseaseInfo.description}
91:             </p>
92:
93:             {!isNotPlant && (
94:                 <div className="flex flex-wrap gap-3">
95:                     <div
96:                         className={`px-4 py-2 ${diseaseInfo.bgColor} ${diseaseInfo.borderColor} border-2 rounded-
97:                     >
98:                         <span className="text-sm font-medium text-gray-600">
99:                             Category:{" "}
100:                        </span>
101:                        <span className={`font-bold ${diseaseInfo.color}`} >
102:                            {result.category || "N/A"}
103:                        </span>
104:                    </div>
105:                    {result.subtype && (
106:                        <div
107:                            className={`px-4 py-2 ${diseaseInfo.bgColor} ${diseaseInfo.borderColor} border-2 rounded-
108:                        >
109:                            <span className="text-sm font-medium text-gray-600">
110:                                Subtype:{" "}
111:                            </span>
112:                            <span className={`font-bold ${diseaseInfo.color}`} >
113:                                {result.subtype}
114:                            </span>
115:                        </div>
116:                    )}
117:                </div>
118:            )}
119:        </div>
120:    </div>
121: </Card>
122:

```

```

123:     { /* Uploaded Image */ }
124:     { imagePreview && (
125:         <Card className="p-4">
126:             <h3 className="text-lg font-semibold text-gray-900 mb-3">
127:                 Analyzed Image
128:             </h3>
129:             <img
130:                 src={imagePreview}
131:                 alt="Analyzed"
132:                 className="w-full max-h-96 object-contain rounded-xl shadow-lg"
133:             />
134:         </Card>
135:     ) }
136:
137:     { /* Confidence - Only show for plant images */ }
138:     { !isNotPlant && (
139:         <Card className="p-6">
140:             <h3 className="text-lg font-semibold text-gray-900 mb-4">
141:                 Confidence Level
142:             </h3>
143:             <ConfidenceGauge confidence={result.confidence} />
144:         </Card>
145:     ) }
146:
147:     { /* Recommendations */ }
148:     { diseaseInfo.recommendations &&
149:         diseaseInfo.recommendations.length > 0 && (
150:         <Card className="p-6">
151:             <h3 className="text-lg font-semibold text-gray-900 mb-4">
152:                 {isNotPlant
153:                     ? "? Tips for Better Results"
154:                     : "Recommended Actions"}
155:             </h3>
156:             <div className="space-y-3">
157:                 {diseaseInfo.recommendations.map((rec, index) => (
158:                     <RecommendationCard
159:                         key={index}
160:                         recommendation={rec}
161:                         index={index}
162:                     />
163:                 ))}
164:             </div>
165:         </Card>
166:     ) }
167:
168:     { /* Privacy Notice - Only for valid plant diagnoses */ }
169:     { !isNotPlant && (
170:         <div
171:             className={`p-4 border-2 rounded-xl ${
172:                 user ? "bg-green-50 border-green-200" : "bg-blue-50 border-blue-200"
173:             }`}
174:         >
175:             <p className={`text-sm ${user ? "text-green-800" : "text-blue-800"} `}>
176:                 {user ? (
177:                     <>
178:                         <strong>Saved to History:</strong> This diagnosis has been
179:                         automatically saved to your account. View it in your{" "}
180:                         <a
181:                             href="/dashboard"
182:                             className="underline font-semibold hover:opacity-80"
183:                         >
184:                             Dashboard
185:                         </a>
186:                     .
187:                     </>
188:                 ) : (
189:                     <>
190:                         <strong>Privacy Notice:</strong> This diagnosis was not
191:                         saved. Download the PDF to keep your results. Want to track your
192:                         plant health history?{" "}
193:                         <a
194:                             href="/register"
195:                             className="underline font-semibold hover:text-blue-900"

```

```

196:         >
197:         Create a free account
198:       </a>
199:     .
200:   </>
201:   )}
202: </p>
203: </div>
204: )}
205: </motion.div>
206: );
207: };
208:
209: export default DiagnosisResult;

```

■ File: src\components\diagnosis\ImageUploader.jsx

```

1: import { useCallback } from 'react'
2: import { useDropzone } from 'react-dropzone'
3: import { Upload, X, Image as ImageIcon } from 'lucide-react'
4: import { motion, AnimatePresence } from 'framer-motion'
5:
6: const ImageUploader = ({ onImageSelect, selectedImage, imagePreview, onRemove }) => {
7:   const onDrop = useCallback((acceptedFiles) => {
8:     const file = acceptedFiles[0]
9:     if (file) {
10:       onImageSelect(file)
11:     }
12:   }, [onImageSelect])
13:
14:   const { getRootProps, getInputProps, isDragActive } = useDropzone({
15:     onDrop,
16:     accept: {
17:       'image/jpeg': ['.jpg', '.jpeg'],
18:       'image/png': ['.png']
19:     },
20:     maxFiles: 1,
21:     maxSize: 5242880, // 5MB
22:   })
23:
24:   return (
25:     <div className="space-y-4">
26:       <AnimatePresence mode="wait">
27:         {!imagePreview ? (
28:           <motion.div
29:             key="dropzone"
30:             initial={{ opacity: 0 }}
31:             animate={{ opacity: 1 }}
32:             exit={{ opacity: 0 }}
33:             {...getRootProps()}
34:             className={`border-3 border-dashed rounded-2xl p-16 text-center cursor-pointer transition-all duration-300
35:               ${isDragActive
36:                 ? 'border-primary-500 bg-primary-50 scale-105'
37:                 : 'border-gray-300 hover:border-primary-400 hover:bg-primary-50/50'}
38:             `}
39:           >
40:             <input {...getInputProps()} />
41:             <div className="flex flex-col items-center gap-4">
42:               <div className="p-6 rounded-full bg-primary-100">
43:                 <Upload className="text-primary-600" size={48} />
44:               </div>
45:               <div>
46:                 <p className="text-2xl font-semibold text-gray-900 mb-2">
47:                   {isDragActive ? 'Drop your image here' : 'Upload Plant Image'}
48:                 </p>
49:                 <p className="text-gray-600 mb-1">
50:                   Drag & drop or click to browse
51:                 </p>
52:                 <p className="text-sm text-gray-500">
53:                   Supports: JPG, JPEG, PNG (Max 5MB)
54:                 </p>

```



```

55:         </div>
56:     </div>
57: </motion.div>
58: ) : (
59:     <motion.div
60:       key="preview"
61:       initial={{ opacity: 0, scale: 0.9 }}
62:       animate={{ opacity: 1, scale: 1 }}
63:       exit={{ opacity: 0, scale: 0.9 }}
64:       className="relative rounded-2xl overflow-hidden bg-gray-100 shadow-lg"
65:     >
66:       <img
67:         src={imagePreview}
68:         alt="Preview"
69:         className="w-full h-96 object-contain"
70:       />
71:       <button
72:         onClick={onRemove}
73:         className="absolute top-4 right-4 p-3 rounded-full bg-red-500 text-white hover:bg-red-600 sha
74:       >
75:         <X size={24} />
76:       </button>
77:     </motion.div>
78:   )}
79: </AnimatePresence>
80:
81:   {imagePreview && (
82:     <div className="flex items-center gap-2 text-sm text-gray-600 bg-green-50 border-2 border-green-200
83:       <ImageIcon size={20} className="text-green-600" />
84:       <span className="font-medium">Image ready for diagnosis</span>
85:     </div>
86:   )}
87: </div>
88: )
89: }
90:
91: export default ImageUploader

```

■ File: src\components\diagnosis\RecommendationCard.jsx

```

1: import { motion } from 'framer-motion'
2: import { CheckCircle } from 'lucide-react'
3:
4: const RecommendationCard = ({ recommendation, index }) => {
5:   return (
6:     <motion.div
7:       initial={{ opacity: 0, x: -20 }}
8:       animate={{ opacity: 1, x: 0 }}
9:       transition={{ delay: index * 0.1 }}
10:      className="flex items-start gap-3 p-4 bg-primary-50 border-2 border-primary-200 rounded-xl hover:bg-p
11:    >
12:      <CheckCircle className="text-primary-600 flex-shrink-0 mt-0.5" size={20} />
13:      <p className="text-gray-700 font-medium">{recommendation}</p>
14:    </motion.div>
15:  )
16: }
17:
18: export default RecommendationCard

```

■ File: src\components\history\SessionCard.jsx

```

1: import { Calendar, Trash2 } from 'lucide-react'
2: import Card from '../common/Card'
3: import { formatDate } from '@/utils/helpers'
4: import { DISEASE_INFO } from '@/utils/constants'
5:
6: const SessionCard = ({ session, onClick, onDelete }) => {
7:   const lastPrediction = session.last_prediction
8:   const diseaseInfo = lastPrediction ? DISEASE_INFO[lastPrediction.predicted_class] : null

```

```

9:
10:   return (
11:     <Card className="p-6 cursor-pointer group hover:shadow-xl" onClick={onClick}>
12:       <div className="flex items-center justify-between">
13:         <div className="flex-1">
14:           <h3 className="text-lg font-semibold text-gray-900 group-hover:text-primary-600 transition-colors">
15:             {session.title}
16:           </h3>
17:
18:           <div className="flex items-center gap-4 text-sm text-gray-600">
19:             <div className="flex items-center gap-1">
20:               <Calendar size={16} />
21:               <span>{formatDate(session.created_at)}</span>
22:             </div>
23:             {lastPrediction && diseaseInfo && (
24:               <div className={`flex items-center gap-2 px-3 py-1 ${diseaseInfo.bgColor} ${diseaseInfo.borderColor}`>
25:                 <span className="text-xl">{diseaseInfo.icon}</span>
26:                 <span className={`font-medium ${diseaseInfo.color}`}>
27:                   {lastPrediction.predicted_class.replace(/_/g, ' ')}
28:                 </span>
29:               </div>
30:             )}
31:           </div>
32:         </div>
33:
34:         <button
35:           onClick={(e) => {
36:             e.stopPropagation()
37:             onDelete()
38:           }}
39:           className="p-2 rounded-lg text-gray-400 hover:text-red-600 hover:bg-red-50 transition-all"
40:         >
41:           <Trash2 size={20} />
42:         </button>
43:       </div>
44:     </Card>
45:   )
46: }
47:
48: export default SessionCard

```

■ File: src\components\history\SessionList.jsx

```

=====
1: import { motion } from 'framer-motion'
2: import SessionCard from './SessionCard'
3: import Loader from '../common/Loader'
4:
5: const SessionList = ({ sessions, loading, onSessionClick, onDeleteSession }) => {
6:   if (loading) {
7:     return <Loader />
8:   }
9:
10:  if (!sessions || sessions.length === 0) {
11:    return (
12:      <div className="text-center py-16">
13:        <div className="text-8xl mb-6">?</div>
14:        <h3 className="text-2xl font-semibold text-gray-700 mb-3">No sessions yet</h3>
15:        <p className="text-gray-500 mb-8">Start diagnosing plants to create your first session</p>
16:      </div>
17:    )
18:  }
19:
20:  return (
21:    <div className="grid gap-4">
22:      {sessions.map((session, index) => (
23:        <motion.div
24:          key={session.id}
25:          initial={{ opacity: 0, y: 20 }}
26:          animate={{ opacity: 1, y: 0 }}
27:          transition={{ delay: index * 0.05 }}
28:        >

```

```

29:         <SessionCard
30:             session={session}
31:             onClick={() => onSessionClick(session.id)}
32:             onDelete={() => onDeleteSession(session.id)}
33:         />
34:     </motion.div>
35:   )}
36: </div>
37: )
38: }
39:
40: export default SessionList

```

■ File: src\components\layout\Footer.jsx

```

=====
1: import { Leaf, Github, Twitter, Mail } from 'lucide-react'
2: import { Link } from 'react-router-dom'
3:
4: const Footer = () => {
5:   return (
6:     <footer className="bg-white border-t-2 border-primary-100 mt-auto">
7:       <div className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8 py-12">
8:         <div className="grid grid-cols-1 md:grid-cols-4 gap-8">
9:           {/* Brand */}
10:          <div className="space-y-4">
11:            <div className="flex items-center gap-2">
12:              <div className="bg-gradient-to-br from-primary-500 to-forest-600 p-2 rounded-xl">
13:                <Leaf className="text-white" size={24} />
14:              </div>
15:              <span className="text-xl font-bold bg-gradient-to-r from-primary-700 to-forest-700 bg-clip-text">
16:                Plant Health
17:              </span>
18:            </div>
19:            <p className="text-gray-600 text-sm">
20:              AI-powered plant disease detection. Keep your plants healthy with advanced technology.
21:            </p>
22:          </div>
23:
24:          {/* Quick Links */}
25:          <div>
26:            <h3 className="font-semibold text-gray-900 mb-4">Quick Links</h3>
27:            <ul className="space-y-2">
28:              <li><Link to="/" className="text-gray-600 hover:text-primary-600 text-sm transition-colors">Home</li>
29:              <li><Link to="/dashboard" className="text-gray-600 hover:text-primary-600 text-sm transition-colors">Dashboard</li>
30:              <li><Link to="/login" className="text-gray-600 hover:text-primary-600 text-sm transition-colors">Login</li>
31:              <li><Link to="/register" className="text-gray-600 hover:text-primary-600 text-sm transition-colors">Register</li>
32:            </ul>
33:          </div>
34:
35:          {/* Features */}
36:          <div>
37:            <h3 className="font-semibold text-gray-900 mb-4">Features</h3>
38:            <ul className="space-y-2">
39:              <li className="text-gray-600 text-sm">Instant Diagnosis</li>
40:              <li className="text-gray-600 text-sm">PDF Reports</li>
41:              <li className="text-gray-600 text-sm">History Tracking</li>
42:              <li className="text-gray-600 text-sm">Privacy First</li>
43:            </ul>
44:          </div>
45:
46:          {/* Social */}
47:          <div>
48:            <h3 className="font-semibold text-gray-900 mb-4">Connect</h3>
49:            <div className="flex gap-4">
50:              <a href="#" className="p-2 rounded-lg bg-primary-50 text-primary-600 hover:bg-primary-100 transition-colors">
51:                <Github size={20} />
52:              </a>
53:              <a href="#" className="p-2 rounded-lg bg-primary-50 text-primary-600 hover:bg-primary-100 transition-colors">
54:                <Twitter size={20} />
55:              </a>
56:              <a href="#" className="p-2 rounded-lg bg-primary-50 text-primary-600 hover:bg-primary-100 transition-colors">

```

```

57:         <Mail size={20} />
58:     </a>
59: </div>
60: </div>
61: </div>
62:
63: <div className="border-t-2 border-primary-100 mt-8 pt-8 text-center">
64:     <p className="text-gray-600 text-sm">
65:         © 2024 Plant Health Detector. All rights reserved.
66:     </p>
67:     <p className="text-gray-500 text-xs mt-2">
68:         Made with ? for plant lovers everywhere
69:     </p>
70: </div>
71: </div>
72: </footer>
73: )
74: }
75:
76: export default Footer

```

■ File: src\components\layout\Navbar.jsx

```

1: // src/components/layout/Navbar.jsx - COMPLETE WITH ADMIN LINK
2: import { Link, useNavigate } from "react-router-dom";
3: import { Leaf, LogOut, User, Menu, X, History, Settings } from "lucide-react";
4: import { useAuth } from "@/hooks/useAuth";
5: import { useToast } from "@/hooks/useToast";
6: import { useState } from "react";
7: import Button from "../common/Button";
8: import { motion, AnimatePresence } from "framer-motion";
9:
10: const Navbar = () => {
11:     const { user, logout } = useAuth();
12:     const { addToast } = useToast();
13:     const navigate = useNavigate();
14:     const [mobileMenuOpen, setMobileMenuOpen] = useState(false);
15:
16:     const handleLogout = async () => {
17:         try {
18:             await logout();
19:             addToast("Logged out successfully", "success");
20:             navigate("/");
21:         } catch (error) {
22:             addToast("Failed to logout", "error");
23:         }
24:     };
25:
26:     return (
27:         <nav className="bg-white/90 backdrop-blur-lg border-b-2 border-primary-100 sticky top-0 z-50 shadow-lg">
28:             <div className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">
29:                 <div className="flex items-center justify-between h-16">
30:                     {/* Logo */}
31:                     <Link to="/" className="flex items-center gap-2 group">
32:                         <div className="bg-gradient-to-br from-primary-500 to-forest-600 p-2 rounded-xl group-hover:scale-105">
33:                             <Leaf className="text-white" size={24} />
34:                         </div>
35:                         <span className="text-xl font-bold bg-gradient-to-r from-primary-700 to-forest-700 bg-clip-text">
36:                             Leafora
37:                         </span>
38:                     </Link>
39:
40:                     {/* Desktop Menu */}
41:                     <div className="hidden md:flex items-center gap-4">
42:                         {user ? (
43:                             <>
44:                                 <Link to="/dashboard">
45:                                     <Button variant="ghost" icon={<History size={20} />}>
46:                                         My History
47:                                     </Button>
48:                                 </Link>

```

```

49:         <Link to="/admin">
50:             <Button variant="ghost" icon={<Settings size={20} />}>
51:                 System
52:             </Button>
53:         </Link>
54:         <Link to="/profile">
55:             <Button variant="ghost" icon={<User size={20} />}>
56:                 {user.username}
57:             </Button>
58:         </Link>
59:         <Button
60:             variant="outline"
61:             onClick={handleLogout}
62:             icon={<LogOut size={20} />}
63:         >
64:             Logout
65:         </Button>
66:     </>
67: ) : (
68:     <>
69:         <Link to="/login">
70:             <Button variant="ghost">Sign In</Button>
71:         </Link>
72:         <Link to="/register">
73:             <Button variant="primary">Create Account</Button>
74:         </Link>
75:     </>
76: )}
77: </div>
78:
79: { /* Mobile Menu Button */}
80: <button
81:     onClick={() => setMobileMenuOpen(!mobileMenuOpen)}
82:     className="md:hidden p-2 rounded-lg hover:bg-primary-50 transition-colors"
83: >
84:     {mobileMenuOpen ? <X size={24} /> : <Menu size={24} />}
85: </button>
86: </div>
87: </div>
88:
89: { /* Mobile Menu */}
90: <AnimatePresence>
91:     {mobileMenuOpen && (
92:         <motion.div
93:             initial={{ opacity: 0, height: 0 }}
94:             animate={{ opacity: 1, height: "auto" }}
95:             exit={{ opacity: 0, height: 0 }}
96:             className="md:hidden border-t-2 border-primary-100 bg-white"
97:         >
98:             <div className="px-4 py-4 space-y-2">
99:                 {user ? (
100:                     <>
101:                         <Link
102:                             to="/dashboard"
103:                             onClick={() => setMobileMenuOpen(false)}
104:                         >
105:                             <Button
106:                                 variant="ghost"
107:                                 className="w-full justify-start"
108:                                 icon={<History size={20} />}
109:                             >
110:                                 My History
111:                             </Button>
112:                         </Link>
113:                         <Link to="/admin" onClick={() => setMobileMenuOpen(false)}>
114:                             <Button
115:                                 variant="ghost"
116:                                 className="w-full justify-start"
117:                                 icon={<Settings size={20} />}
118:                             >
119:                                 System
120:                             </Button>
121:                         </Link>

```

```

122:         <Link to="/profile" onClick={() => setMobileMenuOpen(false)}>
123:             <Button
124:                 variant="ghost"
125:                 className="w-full justify-start"
126:                 icon={<User size={20} />}
127:             >
128:                 Profile
129:             </Button>
130:         </Link>
131:         <Button
132:             variant="outline"
133:             onClick={() => {
134:                 handleLogout();
135:                 setMobileMenuOpen(false);
136:             }}
137:             className="w-full justify-start"
138:             icon={<LogOut size={20} />}
139:         >
140:             Logout
141:         </Button>
142:     </>
143: ) : (
144:     <>
145:         <Link to="/login" onClick={() => setMobileMenuOpen(false)}>
146:             <Button variant="ghost" className="w-full">
147:                 Sign In
148:             </Button>
149:         </Link>
150:         <Link to="/register" onClick={() => setMobileMenuOpen(false)}>
151:             <Button variant="primary" className="w-full">
152:                 Create Account
153:             </Button>
154:         </Link>
155:     </>
156: )}
157: </div>
158: </motion.div>
159: )}
160: </AnimatePresence>
161: </nav>
162: );
163: };
164:
165: export default Navbar;

```

■ File: src\contexts\AuthContext.jsx

```

1: import { createContext, useState, useEffect } from 'react'
2: import { authService } from '../services/authService'
3:
4: export const AuthContext = createContext()
5:
6: export const AuthProvider = ({ children }) => {
7:   const [user, setUser] = useState(null)
8:   const [loading, setLoading] = useState(true)
9:
10:  useEffect(() => {
11:    checkAuth()
12:  }, [])
13:
14:  const checkAuth = async () => {
15:    const token = localStorage.getItem('accessToken')
16:    if (token) {
17:      try {
18:        const userData = await authService.getProfile()
19:        setUser(userData)
20:      } catch (error) {
21:        localStorage.removeItem('accessToken')
22:      }
23:    }
24:    setLoading(false)

```

```

25:   }
26:
27:   const login = async (email, password) => {
28:     const data = await authService.login(email, password)
29:     setUser(data.user)
30:     return data
31:   }
32:
33:   const register = async (username, email, password) => {
34:     const data = await authService.register(username, email, password)
35:     return data
36:   }
37:
38:   const logout = async () => {
39:     await authService.logout()
40:     setUser(null)
41:   }
42:
43:   return (
44:     <AuthContext.Provider value={{ user, loading, login, register, logout, checkAuth }}>
45:       {children}
46:     </AuthContext.Provider>
47:   )
48: }

```

■ File: src\contexts\ToastContext.jsx

```

=====
1: import { createContext, useState, useCallback } from 'react'
2: import Toast from '../components/common/Toast'
3:
4: export const ToastContext = createContext()
5:
6: export const ToastProvider = ({ children }) => {
7:   const [toasts, setToasts] = useState([])
8:
9:   const addToast = useCallback((message, type = 'info') => {
10:     const id = Date.now()
11:     setToasts((prev) => [...prev, { id, message, type }])
12:
13:     // Auto-remove after 5 seconds
14:     setTimeout(() => {
15:       setToasts((prev) => prev.filter((toast) => toast.id !== id))
16:     }, 5000)
17:   }, [])
18:
19:   const removeToast = useCallback((id) => {
20:     setToasts((prev) => prev.filter((toast) => toast.id !== id))
21:   }, [])
22:
23:   return (
24:     <ToastContext.Provider value={{ addToast }}>
25:       {children}
26:       <div className="fixed top-4 right-4 z-50 space-y-2">
27:         {toasts.map((toast) => (
28:           <Toast
29:             key={toast.id}
30:             message={toast.message}
31:             type={toast.type}
32:             onClose={() => removeToast(toast.id)}
33:           />
34:         ))}
35:       </div>
36:     </ToastContext.Provider>
37:   )
38: }
=====

```

■ File: src\hooks\useAuth.js

```
=====
1: import { useContext } from 'react'
2: import { AuthContext } from '../contexts/AuthContext'
3:
4: export const useAuth = () => {
5:   const context = useContext(AuthContext)
6:   if (!context) {
7:     throw new Error('useAuth must be used within AuthProvider')
8:   }
9:   return context
10: }
```

■ File: src\hooks\useToast.js

```
=====
1: import { useContext } from 'react'
2: import { ToastContext } from '../contexts/ToastContext'
3:
4: export const useToast = () => {
5:   const context = useContext(ToastContext)
6:   if (!context) {
7:     throw new Error('useToast must be used within ToastProvider')
8:   }
9:   return context
10: }
```

■ File: src\main.jsx

```
=====
1: import React from 'react'
2: import ReactDOM from 'react-dom/client'
3: import App from './App'
4: import './styles/index.css'
5:
6: ReactDOM.createRoot(document.getElementById('root')).render(
7:   <React.StrictMode>
8:     <App />
9:   </React.StrictMode>,
10: )
```

■ File: src\pages\AdminDashboard.jsx

```
=====
1: // src/pages/AdminDashboard.jsx - NEW FILE
2: // Comprehensive system monitoring dashboard
3:
4: import { useState, useEffect } from "react";
5: import {
6:   Activity,
7:   Database,
8:   HardDrive,
9:   Shield,
10:  RefreshCw,
11:  Trash2,
12:  CheckCircle,
13:  XCircle,
14: } from "lucide-react";
15: import { motion } from "framer-motion";
16: import Navbar from "../components/layout/Navbar";
17: import Footer from "../components/layout/Footer";
18: import Card from "../components/common/Card";
19: import Button from "../components/common/Button";
20: import { useToast } from "@/hooks/useToast";
21: import { systemService } from "../services/systemService";
22:
23: const AdminDashboard = () => {
24:   const { addToast } = useToast();
25:
26:   const [loading, setLoading] = useState(true);
```



```

27:   const [refreshing, setRefreshing] = useState(false);
28:
29:   const [systemHealth, setSystemHealth] = useState(null);
30:   const [redisStatus, setRedisStatus] = useState(null);
31:   const [guestStats, setGuestStats] = useState(null);
32:   const [databaseStats, setDatabaseStats] = useState(null);
33:   const [storageStats, setStorageStats] = useState(null);
34:
35:   useEffect(() => {
36:     loadAllData();
37:   }, []);
38:
39:   const loadAllData = async () => {
40:     setLoading(true);
41:     try {
42:       await Promise.all([
43:         loadSystemHealth(),
44:         loadRedisStatus(),
45:         loadGuestStats(),
46:         loadDatabaseStats(),
47:         loadStorageStats(),
48:       ]);
49:     } catch (error) {
50:       addToast("Failed to load system data", "error");
51:     } finally {
52:       setLoading(false);
53:     }
54:   };
55:
56:   const loadSystemHealth = async () => {
57:     try {
58:       const data = await systemService.getHealth();
59:       setSystemHealth(data);
60:     } catch (error) {
61:       console.error("Failed to load health:", error);
62:     }
63:   };
64:
65:   const loadRedisStatus = async () => {
66:     try {
67:       const data = await systemService.getRedisStatus();
68:       setRedisStatus(data);
69:     } catch (error) {
70:       console.error("Failed to load Redis status:", error);
71:     }
72:   };
73:
74:   const loadGuestStats = async () => {
75:     try {
76:       const data = await systemService.getGuestStats();
77:       setGuestStats(data);
78:     } catch (error) {
79:       console.error("Failed to load guest stats:", error);
80:     }
81:   };
82:
83:   const loadDatabaseStats = async () => {
84:     try {
85:       const data = await systemService.getDatabaseStats();
86:       setDatabaseStats(data);
87:     } catch (error) {
88:       console.error("Failed to load database stats:", error);
89:     }
90:   };
91:
92:   const loadStorageStats = async () => {
93:     try {
94:       const data = await systemService.getStorageStats();
95:       setStorageStats(data);
96:     } catch (error) {
97:       console.error("Failed to load storage stats:", error);
98:     }
99:   };

```

```

100:
101: const handleRefresh = async () => {
102:   setRefreshing(true);
103:   await loadAllData();
104:   setRefreshing(false);
105:   addToast("Data refreshed", "success");
106: };
107:
108: const handleClearGuestStore = async () => {
109:   if (
110:     !window.confirm(
111:       "Clear guest memory store? This will reset all guest rate limits.",
112:     )
113:   ) {
114:     return;
115:   }
116:
117:   try {
118:     await systemService.clearGuestStore();
119:     addToast("Guest store cleared", "success");
120:     await loadGuestStats();
121:   } catch (error) {
122:     addToast("Failed to clear guest store", "error");
123:   }
124: };
125:
126: const StatusBadge = ({ status }) => {
127:   const isHealthy =
128:     status === "healthy" ||
129:     status === "operational" ||
130:     status === "connected";
131:
132:   return (
133:     <div
134:       className={`inline-flex items-center gap-2 px-3 py-1 rounded-full ${
135:         isHealthy
136:           ? "bg-green-100 text-green-700"
137:           : "bg-yellow-100 text-yellow-700"
138:       }`}
139:     >
140:       {isHealthy ? <CheckCircle size={16} /> : <XCircle size={16} />}
141:       <span className="font-medium">{status}</span>
142:     </div>
143:   );
144: };
145:
146: if (loading) {
147:   return (
148:     <div className="min-h-screen flex flex-col">
149:       <Navbar />
150:       <main className="flex-1 flex items-center justify-center">
151:         <div className="animate-spin rounded-full h-12 w-12 border-b-2 border-primary-600"></div>
152:       </main>
153:       <Footer />
154:     </div>
155:   );
156: }
157:
158: return (
159:   <div className="min-h-screen flex flex-col bg-gray-50">
160:     <Navbar />
161:
162:     <main className="flex-1 max-w-7xl mx-auto px-4 sm:px-6 lg:px-8 py-8 w-full">
163:       </* Header */>
164:       <div className="flex items-center justify-between mb-8">
165:         <div>
166:           <h1 className="text-4xl font-bold text-gray-900 mb-2">
167:             System Dashboard
168:           </h1>
169:           <p className="text-gray-600">
170:             Monitor system health and performance
171:           </p>
172:         </div>

```

```

173:         <Button
174:             onClick={handleRefresh}
175:             loading={refreshing}
176:             icon={<RefreshCw size={20} />}
177:         >
178:             Refresh
179:         </Button>
180:     </div>
181:
182:     { /* System Health Overview */}
183:     {systemHealth && (
184:         <Card className="p-6 mb-6">
185:             <div className="flex items-center justify-between mb-6">
186:                 <h2 className="text-2xl font-bold text-gray-900">
187:                     System Health
188:                 </h2>
189:                 <StatusBadge status={systemHealth.status} />
190:             </div>
191:
192:             <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-4">
193:                 { /* Database */}
194:                 <div className="p-4 bg-gradient-to-br from-blue-50 to-blue-100 rounded-xl border-2 border-blue-200">
195:                     <div className="flex items-center gap-3 mb-2">
196:                         <Database size={24} className="text-blue-600" />
197:                         <h3 className="font-semibold text-gray-900">Database</h3>
198:                     </div>
199:                     <StatusBadge status={systemHealth.services?.database?.status} />
200:                     <div className="mt-3 text-sm text-gray-700">
201:                         <p>Type: {systemHealth.services?.database?.type}</p>
202:                         <p>
203:                             Connected: { " " }
204:                             {systemHealth.services?.database?.connected ? "Yes" : "No"}
205:                         </p>
206:                     </div>
207:                 </div>
208:
209:                 { /* Redis */}
210:                 <div className="p-4 bg-gradient-to-br from-red-50 to-red-100 rounded-xl border-2 border-red-200">
211:                     <div className="flex items-center gap-3 mb-2">
212:                         <Shield size={24} className="text-red-600" />
213:                         <h3 className="font-semibold text-gray-900">Redis</h3>
214:                     </div>
215:                     <StatusBadge status={systemHealth.services?.redis?.status} />
216:                     <div className="mt-3 text-sm text-gray-700">
217:                         <p>Guest: {systemHealth.services?.redis?.guest_limiter}</p>
218:                         <p>Rate: {systemHealth.services?.redis?.rate_limiter}</p>
219:                     </div>
220:                 </div>
221:
222:                 { /* AI Service */}
223:                 <div className="p-4 bg-gradient-to-br from-purple-50 to-purple-100 rounded-xl border-2 border-purple-200">
224:                     <div className="flex items-center gap-3 mb-2">
225:                         <Activity size={24} className="text-purple-600" />
226:                         <h3 className="font-semibold text-gray-900">AI Service</h3>
227:                     </div>
228:                     <StatusBadge
229:                         status={systemHealth.services?.ai_service?.status}
230:                     />
231:                     <div className="mt-3 text-sm text-gray-700">
232:                         <p>Workers: {systemHealth.services?.ai_service?.workers}</p>
233:                         <p>
234:                             Success: {systemHealth.services?.ai_service?.success_rate}
235:                         </p>
236:                     </div>
237:                 </div>
238:
239:                 { /* Storage */}
240:                 <div className="p-4 bg-gradient-to-br from-green-50 to-green-100 rounded-xl border-2 border-green-200">
241:                     <div className="flex items-center gap-3 mb-2">
242:                         <HardDrive size={24} className="text-green-600" />
243:                         <h3 className="font-semibold text-gray-900">Storage</h3>
244:                     </div>
245:                     <StatusBadge status={systemHealth.services?.storage?.status} />

```

```

246:         <div className="mt-3 text-sm text-gray-700">
247:             <p>Files: {systemHealth.services?.storage?.files}</p>
248:             <p>Size: {systemHealth.services?.storage?.total_size}</p>
249:         </div>
250:     </div>
251: </div>
252: </Card>
253: )}
254:
255: {/* Guest Statistics */}
256: {guestStats && (
257:     <Card className="p-6 mb-6">
258:         <div className="flex items-center justify-between mb-6">
259:             <h2 className="text-2xl font-bold text-gray-900">
260:                 Guest Predictions
261:             </h2>
262:             <Button
263:                 onClick={handleClearGuestStore}
264:                 variant="outline"
265:                 icon={<Trash2 size={18} />}
266:                 size="sm"
267:             >
268:                 Clear Store
269:             </Button>
270:         </div>
271:
272:         <div className="grid grid-cols-1 md:grid-cols-3 gap-4">
273:             <div className="p-4 bg-gray-50 rounded-xl">
274:                 <p className="text-sm text-gray-600 mb-1">Storage Type</p>
275:                 <p className="text-2xl font-bold text-gray-900">
276:                     {guestStats.storage_type}
277:                 </p>
278:             </div>
279:
280:             <div className="p-4 bg-gray-50 rounded-xl">
281:                 <p className="text-sm text-gray-600 mb-1">Redis Connected</p>
282:                 <p className="text-2xl font-bold text-gray-900">
283:                     {guestStats.redis_connected ? "Yes" : "No"}
284:                 </p>
285:             </div>
286:
287:             {guestStats.memory_store_stats && (
288:                 <div className="p-4 bg-gray-50 rounded-xl">
289:                     <p className="text-sm text-gray-600 mb-1">Memory Store</p>
290:                     <div className="text-sm text-gray-700 mt-2">
291:                         <p>
292:                             Fingerprints:{" " }
293:                             {guestStats.memory_store_stats.fingerprints || 0}
294:                         </p>
295:                         <p>IPs: {guestStats.memory_store_stats.ips || 0}</p>
296:                         <p>
297:                             Sessions: {guestStats.memory_store_stats.sessions || 0}
298:                         </p>
299:                     </div>
300:                 </div>
301:             )}
302:         </div>
303:     </Card>
304: )}
305:
306: {/* Database Statistics */}
307: {databaseStats && (
308:     <Card className="p-6 mb-6">
309:         <h2 className="text-2xl font-bold text-gray-900 mb-6">
310:             Database Statistics
311:         </h2>
312:
313:         <div className="grid grid-cols-1 md:grid-cols-2 gap-6">
314:             {/* Pool Stats */}
315:             {databaseStats.pool && (
316:                 <div>
317:                     <h3 className="font-semibold text-gray-900 mb-3">
318:                         Connection Pool

```

```

319:         </h3>
320:         <div className="space-y-2 text-sm">
321:             <div className="flex justify-between">
322:                 <span className="text-gray-600">Dialect:</span>
323:                 <span className="font-medium">
324:                     {databaseStats.pool.dialect}
325:                 </span>
326:             </div>
327:             {databaseStats.pool.size !== undefined && (
328:                 <>
329:                     <div className="flex justify-between">
330:                         <span className="text-gray-600">Pool Size:</span>
331:                         <span className="font-medium">
332:                             {databaseStats.pool.size}
333:                         </span>
334:                     </div>
335:                     <div className="flex justify-between">
336:                         <span className="text-gray-600">Available:</span>
337:                         <span className="font-medium">
338:                             {databaseStats.pool.available}
339:                         </span>
340:                     </div>
341:                     <div className="flex justify-between">
342:                         <span className="text-gray-600">In Use:</span>
343:                         <span className="font-medium">
344:                             {databaseStats.pool.using}
345:                         </span>
346:                     </div>
347:                 </>
348:             )}
349:         </div>
350:     </div>
351: )}
352:
353: {/* Query Stats */}
354: {databaseStats.queries && (
355:     <div>
356:         <h3 className="font-semibold text-gray-900 mb-3">
357:             Query Performance
358:         </h3>
359:         <div className="space-y-2 text-sm">
360:             <div className="flex justify-between">
361:                 <span className="text-gray-600">Total Queries:</span>
362:                 <span className="font-medium">
363:                     {databaseStats.queries.totalQueries}
364:                 </span>
365:             </div>
366:             <div className="flex justify-between">
367:                 <span className="text-gray-600">Slow Queries:</span>
368:                 <span className="font-medium">
369:                     {databaseStats.queries.slowQueries}
370:                 </span>
371:             </div>
372:             <div className="flex justify-between">
373:                 <span className="text-gray-600">Avg Time:</span>
374:                 <span className="font-medium">
375:                     {databaseStats.queries.averageQueryTime}
376:                 </span>
377:             </div>
378:         </div>
379:     </div>
380: )}
381: </div>
382: </Card>
383: )}
384:
385: {/* Storage Statistics */}
386: {storageStats && (
387:     <Card className="p-6">
388:         <h2 className="text-2xl font-bold text-gray-900 mb-6">
389:             Storage Statistics
390:         </h2>
391:

```

```

392:         <div className="grid grid-cols-1 md:grid-cols-3 gap-4">
393:             <div className="p-4 bg-gray-50 rounded-xl">
394:                 <p className="text-sm text-gray-600 mb-1">Total Files</p>
395:                 <p className="text-2xl font-bold text-gray-900">
396:                     {storageStats.fileCount}
397:                 </p>
398:             </div>
399:
400:             <div className="p-4 bg-gray-50 rounded-xl">
401:                 <p className="text-sm text-gray-600 mb-1">Total Size</p>
402:                 <p className="text-2xl font-bold text-gray-900">
403:                     {storageStats.totalSizeMB} MB
404:                 </p>
405:             </div>
406:
407:             <div className="p-4 bg-gray-50 rounded-xl">
408:                 <p className="text-sm text-gray-600 mb-1">Pending Cleanups</p>
409:                 <p className="text-2xl font-bold text-gray-900">
410:                     {storageStats.pendingCleanups}
411:                 </p>
412:             </div>
413:         </div>
414:     </Card>
415: }}
416: </main>
417:
418: <Footer />
419: </div>
420: );
421: };
422:
423: export default AdminDashboard;

```

■ File: src\pages\Dashboard.jsx

```

1: import { useState, useEffect } from "react";
2: import { Link } from "react-router-dom";
3: import {
4:   Activity,
5:   TrendingUp,
6:   Plus,
7:   Search,
8:   Calendar,
9:   Trash2,
10:  AlertCircle,
11:  FileText,
12: } from "lucide-react";
13: import { motion } from "framer-motion";
14: import Navbar from "../components/layout/Navbar";
15: import Footer from "../components/layout/Footer";
16: import Card from "../components/common/Card";
17: import Button from "../components/common/Button";
18: import Input from "../components/common/Input";
19: import Modal from "../components/common/Modal";
20: import Loader from "../components/common/Loader";
21: import DiagnosisResult from "../components/diagnosis/DiagnosisResult";
22: import { useAuth } from "@/hooks/useAuth";
23: import { useToast } from "@/hooks/useToast";
24: import { historyService } from "../services/historyService";
25: import { formatDate } from "@/utils/helpers";
26: import { DISEASE_INFO } from "@/utils/constants";
27:
28: const Dashboard = () => {
29:   const { user } = useAuth();
30:   const { addToast } = useToast();
31:   const [sessions, setSessions] = useState([]);
32:   const [loading, setLoading] = useState(true);
33:   const [searchTerm, setSearchTerm] = useState("");
34:   const [modalOpen, setModalOpen] = useState(false);
35:   const [sessionPredictions, setSessionPredictions] = useState([]);
36:

```

```

37:   useEffect(() => {
38:     loadSessions();
39:   }, []);
40:
41:   const loadSessions = async () => {
42:     try {
43:       const data = await historyService.getSessions(100, 0);
44:       setSessions(data.sessions || []);
45:     } catch (error) {
46:       addToast("Failed to load sessions", "error");
47:     } finally {
48:       setLoading(false);
49:     }
50:   };
51:
52:   const loadSessionPredictions = async (sessionId) => {
53:     try {
54:       const data = await historyService.getSessionPredictions(sessionId);
55:       setSessionPredictions(data.predictions || []);
56:       setModalOpen(true);
57:     } catch (error) {
58:       addToast("Failed to load predictions", "error");
59:     }
60:   };
61:
62:   const handleDeleteSession = async (sessionId) => {
63:     if (!window.confirm("Are you sure you want to delete this session?")) {
64:       return;
65:     }
66:
67:     try {
68:       await historyService.deleteSession(sessionId);
69:       setSessions(sessions.filter((s) => s.id !== sessionId));
70:       addToast("Session deleted successfully", "success");
71:     } catch (error) {
72:       addToast("Failed to delete session", "error");
73:     }
74:   };
75:
76:   const filteredSessions = sessions.filter((session) =>
77:     session.title.toLowerCase().includes(searchTerm.toLowerCase())
78:   );
79:
80:   const stats = [
81:     {
82:       icon: <Activity className="text-primary-600" size={28} />,
83:       label: "Total Diagnoses",
84:       value: sessions.length,
85:       bgColor: "from-primary-50 to-primary-100",
86:       borderColor: "border-primary-200",
87:     },
88:     {
89:       icon: <TrendingUp className="text-forest-600" size={28} />,
90:       label: "This Week",
91:       value: sessions.filter((s) => {
92:         const date = new Date(s.created_at);
93:         const now = new Date();
94:         const diffTime = Math.abs(now - date);
95:         const diffDays = Math.ceil(diffTime / (1000 * 60 * 60 * 24));
96:         return diffDays <= 7;
97:       }).length,
98:       bgColor: "from-forest-50 to-forest-100",
99:       borderColor: "border-forest-200",
100:     },
101:   ];
102:
103:   return (
104:     <div className="min-h-screen flex flex-col bg-gray-50">
105:       <Navbar />
106:
107:       <main className="flex-1 max-w-7xl mx-auto px-4 sm:px-6 lg:px-8 py-6 md:py-8 lg:py-10 w-full">
108:         <motion.div
109:           initial={{ opacity: 0, y: 20 }}

```

```

110:         animate={{ opacity: 1, y: 0 }}
111:         className="mb-6 md:mb-8"
112:     >
113:     <h1 className="text-2xl sm:text-3xl md:text-4xl font-bold text-gray-900 mb-2">
114:         Welcome back, {user?.username}
115:     </h1>
116:     <p className="text-sm sm:text-base text-gray-600">
117:         View your diagnosis history and track plant health over time
118:     </p>
119: </motion.div>
120:
121: {/* Two Column Layout - 50/50 Split */}
122: <div className="grid grid-cols-1 lg:grid-cols-2 gap-6">
123:     {/* Left Column - Stats & Actions */}
124:     <div className="space-y-6">
125:         {/* Stats */}
126:         {stats.map((stat, index) => (
127:             <motion.div
128:                 key={index}
129:                 initial={{ opacity: 0, y: 20 }}
130:                 animate={{ opacity: 1, y: 0 }}
131:                 transition={{ delay: index * 0.1 }}
132:             >
133:                 <Card
134:                     className={`p-6 bg-gradient-to-br ${stat.bgColor} border-2 ${stat.borderColor} hover:shadow-2xl`
135:                 >
136:                     <div className="flex items-center justify-between">
137:                         <div>
138:                             <p className="text-xs font-medium mb-1 uppercase tracking-wide text-gray-600">
139:                                 {stat.label}
140:                             </p>
141:                             <p className="text-3xl sm:text-4xl font-bold text-gray-900">
142:                                 {stat.value}
143:                             </p>
144:                         </div>
145:                         <div className="bg-white p-4 rounded-xl shadow-lg">
146:                             {stat.icon}
147:                         </div>
148:                     </div>
149:                 </Card>
150:             </motion.div>
151:         )))
152:
153:         {/* Quick Action */}
154:         <motion.div
155:             initial={{ opacity: 0, y: 20 }}
156:             animate={{ opacity: 1, y: 0 }}
157:             transition={{ delay: 0.2 }}
158:         >
159:             <Link to="/">
160:                 <Card className="p-6 bg-gradient-to-br from-primary-500 to-forest-600 text-white hover:shadow-2xl`
161:                     <div className="flex items-center justify-between">
162:                         <div>
163:                             <h3 className="text-xl sm:text-2xl font-bold mb-1">
164:                                 New Diagnosis
165:                             </h3>
166:                             <p className="text-primary-100 text-sm">
167:                                 Upload an image to diagnose your plant
168:                             </p>
169:                         </div>
170:                         <div className="bg-white/20 p-3 rounded-xl backdrop-blur-sm">
171:                             <Plus size={32} />
172:                         </div>
173:                     </div>
174:                 </Card>
175:             </Link>
176:         </motion.div>
177:     </div>
178:
179:     {/* Right Column - History */}
180:     <div className="space-y-6">
181:         {/* Search */}
182:         <motion.div

```



```

183:     initial={{ opacity: 0, y: 20 }}
184:     animate={{ opacity: 1, y: 0 }}
185:     transition={{ delay: 0.1 }}
186:   >
187:     <Card className="p-4 bg-white shadow-md">
188:       <Input
189:         placeholder="Search sessions..."
190:         value={searchTerm}
191:         onChange={(e) => setSearchTerm(e.target.value)}
192:         icon={<Search size={18} />}
193:         className="text-sm"
194:       />
195:     </Card>
196:   </motion.div>
197:
198:   { /* Section Header */ }
199:   <div className="flex items-center justify-between">
200:     <h2 className="text-xl sm:text-2xl font-bold text-gray-900">
201:       Diagnosis History
202:     </h2>
203:     {filteredSessions.length > 0 && (
204:       <div className="flex items-center gap-2 text-xs text-gray-600 bg-white px-3 py-1.5 rounded-
205:         <FileText size={14} />
206:         <span>{filteredSessions.length}</span>
207:       </div>
208:     )}
209:   </div>
210:
211:   { /* Sessions List */ }
212:   <div className="max-h-[calc(100vh-400px)] overflow-y-auto pr-2">
213:     {loading ? (
214:       <div className="flex items-center justify-center py-16">
215:         <Loader size="lg" />
216:       </div>
217:     ) : filteredSessions.length === 0 ? (
218:       <Card className="p-8 bg-white">
219:         <div className="text-center">
220:           <div className="inline-flex items-center justify-center w-16 h-16 bg-gradient-to-br from
221:             {searchTerm ? (
222:               <AlertCircle size={32} className="text-gray-400" />
223:             ) : (
224:               <Activity size={32} className="text-primary-600" />
225:             )}
226:           </div>
227:           <h3 className="text-lg font-bold text-gray-900 mb-2">
228:             {searchTerm ? "No matching sessions" : "No diagnoses yet"}
229:           </h3>
230:           <p className="text-gray-600 mb-6 text-sm">
231:             {searchTerm
232:               ? "Try a different search term"
233:               : "Start by diagnosing your first plant"}
234:           </p>
235:           {!searchTerm && (
236:             <Link to="/">
237:               <Button size="md" icon={<Plus size={18} />}>
238:                 Diagnose Your First Plant
239:               </Button>
240:             </Link>
241:           )}
242:         </div>
243:       </Card>
244:     ) : (
245:       <div className="space-y-4">
246:         {filteredSessions.map((session, index) => {
247:           const lastPrediction = session.last_prediction;
248:           const diseaseInfo = lastPrediction
249:             ? DISEASE_INFO[lastPrediction.predicted_class]
250:             : null;
251:
252:           return (
253:             <motion.div
254:               key={session.id}
255:               initial={{ opacity: 0, y: 20 }}

```

```

256:         animate={{ opacity: 1, y: 0 }}
257:         transition={{ delay: index * 0.05 }}
258:       >
259:         <Card
260:           className="p-4 cursor-pointer group hover:shadow-xl hover:border-primary-200 tran
261:           onClick={() => loadSessionPredictions(session.id)}
262:         >
263:           <div className="flex items-start justify-between gap-3">
264:             <div className="flex-1 min-w-0">
265:               <h3 className="text-sm sm:text-base font-semibold text-gray-900 group-hover:t
266:                 {session.title}
267:               </h3>
268:
269:               <div className="flex flex-col gap-2 text-xs text-gray-600">
270:                 <div className="flex items-center gap-1.5">
271:                   <Calendar
272:                     size={14}
273:                     className="text-gray-400"
274:                   />
275:                   <span>{formatDate(session.created_at)}</span>
276:                 </div>
277:                 {lastPrediction && diseaseInfo && (
278:                   <div
279:                     className={`inline-flex items-center gap-1.5 px-2 py-1 ${diseaseInfo.bg
280:                   >
281:                     <span className="text-base">
282:                       {diseaseInfo.icon}
283:                     </span>
284:                     <span
285:                       className={`font-medium ${diseaseInfo.color} truncate text-xs`}
286:                     >
287:                       {lastPrediction.predicted_class.replace(
288:                         /_/g,
289:                         " "
290:                       )}
291:                     </span>
292:                   </div>
293:                 )}
294:               </div>
295:             </div>
296:
297:             <button
298:               onClick={(e) => {
299:                 e.stopPropagation();
300:                 handleDeleteSession(session.id);
301:               }}
302:               className="p-2 rounded-lg text-gray-400 hover:text-red-600 hover:bg-red-50 tr
303:               aria-label="Delete session"
304:             >
305:               <Trash2 size={18} />
306:             </button>
307:           </div>
308:         </Card>
309:       </motion.div>
310:     );
311:   }}
312: </div>
313: )}
314: </div>
315: </div>
316: </div>
317: </main>
318:
319: { /* Session Details Modal */ }
320: <Modal
321:   isOpen={modalOpen}
322:   onClose={() => setModalOpen(false)}
323:   title="Session Predictions"
324:   size="lg"
325: >
326:   {sessionPredictions.length > 0 ? (
327:     <div className="space-y-6 max-h-[60vh] md:max-h-[70vh] overflow-y-auto pr-2">
328:       {sessionPredictions.map((pred) => (

```

```

329:         <Card key={pred.id} className="p-4 bg-white">
330:             <DiagnosisResult
331:                 result={pred}
332:                 imagePreview={null}
333:                 onNewDiagnosis={() => setModalOpen(false)}
334:             />
335:         </Card>
336:     )}
337: </div>
338: ) : (
339:     <div className="text-center py-12">
340:         <div className="inline-flex items-center justify-center w-16 h-16 bg-gray-100 rounded-full mb-4">
341:             <AlertCircle size={28} className="text-gray-400" />
342:         </div>
343:         <p className="text-gray-600 text-sm">
344:             No predictions found in this session
345:         </p>
346:     </div>
347: )}
348: </Modal>
349:
350: <Footer />
351: </div>
352: );
353: };
354:
355: export default Dashboard;

```

■ File: src\pages\Home.jsx

```

1: // src/pages/Home.jsx - ENHANCED WITH COMPLETE GUEST MODE SUPPORT
2: import { useState } from "react";
3: import { Link } from "react-router-dom";
4: import {
5:   Leaf,
6:   Upload,
7:   Download,
8:   History,
9:   Shield,
10:  Zap,
11:  CheckCircle,
12:  Info,
13:  AlertCircle,
14: } from "lucide-react";
15: import { motion } from "framer-motion";
16: import Navbar from "../components/layout/Navbar";
17: import Footer from "../components/layout/Footer";
18: import Card from "../components/common/Card";
19: import Button from "../components/common/Button";
20: import Loader from "../components/common/Loader";
21: import ImageUploader from "../components/diagnosis/ImageUploader";
22: import DiagnosisResult from "../components/diagnosis/DiagnosisResult";
23: import { diagnosisService } from "../services/diagnosisService";
24: import { useToast } from "../hooks/useToast";
25: import { useAuth } from "../hooks/useAuth";
26: import { ERROR_CODES, ERROR_MESSAGES } from "../utils/constants";
27:
28: const Home = () => {
29:   const [selectedImage, setSelectedImage] = useState(null);
30:   const [imagePreview, setImagePreview] = useState(null);
31:   const [diagnosing, setDiagnosing] = useState(false);
32:   const [result, setResult] = useState(null);
33:   const [guestMetadata, setGuestMetadata] = useState(null);
34:   const { addToast } = useToast();
35:   const { user } = useAuth();
36:
37:   const handleImageSelect = (file) => {
38:     setSelectedImage(file);
39:     setResult(null);
40:     setGuestMetadata(null);
41:

```

```

42:     const reader = new FileReader();
43:     reader.onload = () => setImagePreview(reader.result);
44:     reader.readAsDataURL(file);
45: };
46:
47: const handleRemoveImage = () => {
48:     setSelectedImage(null);
49:     setImagePreview(null);
50:     setResult(null);
51:     setGuestMetadata(null);
52: };
53:
54: const handleDiagnose = async () => {
55:     if (!selectedImage) {
56:         addToast("Please upload an image first", "warning");
57:         return;
58:     }
59:
60:     setDiagnosing(true);
61:     try {
62:         let data;
63:         if (user) {
64:             // Authenticated mode
65:             data = await diagnosisService.diagnoseWithHistory(selectedImage);
66:             setResult(data.prediction);
67:             addToast("Diagnosis completed and saved!", "success");
68:         } else {
69:             // Guest mode
70:             data = await diagnosisService.diagnoseStateless(selectedImage);
71:             setResult(data.prediction);
72:             setGuestMetadata({
73:                 usage: data.usage,
74:                 loginPrompt: data.login_prompt,
75:                 features: data.features,
76:                 canDownloadPdf: data.can_download_pdf,
77:             });
78:             addToast("Diagnosis completed!", "success");
79:         }
80:     } catch (error) {
81:         const errorCode = error.response?.data?.code;
82:         const errorMessage =
83:             ERROR_MESSAGES[errorCode] ||
84:             error.response?.data?.error ||
85:             "Diagnosis failed";
86:
87:         // Handle specific error codes
88:         if (errorCode === ERROR_CODES.GUEST_SESSION_LIMIT_EXCEEDED) {
89:             addToast(
90:                 "Session limit reached. Sign up to continue analyzing plants!",
91:                 "error",
92:             );
93:         } else if (errorCode === ERROR_CODES.GUEST_DAILY_LIMIT_EXCEEDED) {
94:             addToast(
95:                 "Daily limit reached. Create a free account for 50 predictions per day!",
96:                 "error",
97:             );
98:         } else if (errorCode === ERROR_CODES.SUSPICIOUS_ACTIVITY_DETECTED) {
99:             addToast(
100:                 "Too many rapid requests. Please slow down or create an account.",
101:                 "warning",
102:             );
103:         } else if (errorCode?.includes("RATE_LIMIT")) {
104:             addToast("Too many requests. Please wait and try again.", "warning");
105:         } else if (errorCode === ERROR_CODES.INVALID_FILE_TYPE) {
106:             addToast("Invalid file type. Use JPG or PNG images.", "error");
107:         } else if (errorCode === ERROR_CODES.FILE_TOO_LARGE) {
108:             addToast("File too large. Maximum size is 5MB.", "error");
109:         } else {
110:             addToast(errorMessage, "error");
111:         }
112:     } finally {
113:         setDiagnosing(false);
114:     }

```

```

115:   };
116:
117:   const features = [
118:     {
119:       icon: <Zap className="text-primary-600" size={32} />,
120:       title: "Instant Results",
121:       description:
122:         "Get AI-powered diagnosis in seconds. No waiting, no signup required.",
123:     },
124:     {
125:       icon: <Shield className="text-forest-600" size={32} />,
126:       title: "Privacy First",
127:       description:
128:         "Your images are analyzed instantly and never stored without permission.",
129:     },
130:     {
131:       icon: <Download className="text-sage-600" size={32} />,
132:       title: "Download Reports",
133:       description: "Get detailed PDF reports with treatment recommendations.",
134:     },
135:     {
136:       icon: <History className="text-primary-600" size={32} />,
137:       title: "Optional History",
138:       description:
139:         "Create an account to save and track your diagnoses over time.",
140:     },
141:     {
142:       icon: <Upload className="text-forest-600" size={32} />,
143:       title: "Easy Upload",
144:       description:
145:         "Simple drag and drop interface. Works on mobile and desktop.",
146:     },
147:     {
148:       icon: <Leaf className="text-sage-600" size={32} />,
149:       title: "Multiple Conditions",
150:       description:
151:         "Detect diseases, pests, and nutrient deficiencies accurately.",
152:     },
153:   ];
154:
155:   const tips = [
156:     { icon: <CheckCircle size={18} />, text: "Take photos in good lighting" },
157:     { icon: <CheckCircle size={18} />, text: "Focus on affected areas" },
158:     { icon: <CheckCircle size={18} />, text: "Avoid blurry images" },
159:     { icon: <CheckCircle size={18} />, text: "Include leaves clearly" },
160:   ];
161:
162:   return (
163:     <div className="min-h-screen flex flex-col bg-gray-50">
164:       <Navbar />
165:
166:       <main className="flex-1">
167:         { /* Hero Section */ }
168:         <section className="relative py-16 md:py-24 lg:py-32 px-4 overflow-hidden bg-gradient-to-br from-pr
169:           <div className="absolute inset-0 bg-[url('
170:
171:           <div className="max-w-7xl mx-auto relative z-10">
172:             <motion.div
173:               initial={{ opacity: 0, y: 30 }}
174:               animate={{ opacity: 1, y: 0 }}
175:               className="text-center mb-12 lg:mb-16"
176:             >
177:               <h1 className="text-4xl md:text-5xl lg:text-7xl font-bold mb-6 leading-tight">
178:                 <span className="bg-gradient-to-r from-primary-600 via-forest-600 to-sage-600 bg-clip-text
179:                   AI Plant Health
180:                 </span>
181:               <br />
182:               <span className="text-gray-900">Diagnosis</span>
183:             </h1>
184:
185:             <p className="text-lg md:text-xl text-gray-700 mb-6 max-w-3xl mx-auto leading-relaxed">
186:               Instant plant disease detection powered by AI. No signup
187:               required.

```

```

188:         </p>
189:
190:     <div className="max-w-2xl mx-auto bg-white rounded-lg shadow-sm border border-gray-200 p-4">
191:         <div className="flex items-start gap-3">
192:             <Info
193:                 size={20}
194:                 className="text-primary-600 flex-shrink-0 mt-0.5"
195:             />
196:             <div className="text-left">
197:                 <p className="text-sm text-gray-700">
198:                     <span className="font-semibold">Privacy First:</span> Your
199:                     images are analyzed instantly and nothing is stored.
200:                     {!user && (
201:                         <>
202:                             { " " }
203:                             <Link
204:                                 to="/register"
205:                                 className="text-primary-600 hover:text-primary-700 hover:underline font-semibol
206:                             >
207:                                 Create an account
208:                             </Link>{ " " }
209:                             if you want to save your diagnosis history.
210:                         </>
211:                     )}
212:                 </p>
213:             </div>
214:         </div>
215:     </div>
216: </motion.div>
217:
218: { /* Diagnosis Area */ }
219: <motion.div
220:     initial={{ opacity: 0, y: 20 }}
221:     animate={{ opacity: 1, y: 0 }}
222:     transition={{ delay: 0.2 }}
223:     className="max-w-4xl mx-auto"
224: >
225:     {diagnosing ? (
226:         <Card className="p-8 md:p-12 bg-white shadow-xl">
227:             <Loader size="lg" />
228:             <p className="text-center text-gray-600 mt-6 text-base md:text-lg">
229:                 Analyzing your plant... This may take a few seconds
230:             </p>
231:         </Card>
232:     ) : result ? (
233:         <div className="space-y-6">
234:             <DiagnosisResult
235:                 result={result}
236:                 imagePreview={imagePreview}
237:                 onNewDiagnosis={handleRemoveImage}
238:             />
239:
240:             { /* Guest Usage Tracking */ }
241:             {!user && guestMetadata?.usage && (
242:                 <Card className="p-6 bg-gradient-to-br from-blue-50 to-indigo-50 border-2 border-blue-2
243:                 <div className="flex items-start gap-4">
244:                     <Info
245:                         size={24}
246:                         className="text-blue-600 flex-shrink-0"
247:                     />
248:                     <div className="flex-1">
249:                         <h3 className="text-lg font-semibold text-blue-900 mb-2">
250:                             Usage Tracking
251:                         </h3>
252:                         <div className="grid grid-cols-3 gap-3 text-sm">
253:                             <div className="text-center p-2 bg-white rounded-lg">
254:                                 <p className="text-gray-600 text-xs">
255:                                     Predictions Used
256:                                 </p>
257:                                 <p className="text-xl font-bold text-blue-900">
258:                                     {guestMetadata.usage.predictions_used}
259:                                 </p>
260:                             </div>

```

```

261:         <div className="text-center p-2 bg-white rounded-lg">
262:             <p className="text-gray-600 text-xs">
263:                 Session Left
264:             </p>
265:             <p className="text-xl font-bold text-blue-900">
266:                 {guestMetadata.usage.session_remaining}
267:             </p>
268:         </div>
269:         <div className="text-center p-2 bg-white rounded-lg">
270:             <p className="text-gray-600 text-xs">
271:                 Daily Left
272:             </p>
273:             <p className="text-xl font-bold text-blue-900">
274:                 {guestMetadata.usage.daily_remaining}
275:             </p>
276:         </div>
277:     </div>
278: </div>
279: </div>
280: </Card>
281: )}
282:
283: {/* Login Prompt */}
284: {!user && guestMetadata?.loginPrompt?.show && (
285:     <Card className="p-6 md:p-8 bg-gradient-to-br from-orange-50 to-red-50 border-2 border-
286:         <div className="text-center">
287:             <AlertCircle
288:                 size={40}
289:                 className="mx-auto mb-4 text-orange-600"
290:             />
291:             <h3 className="text-lg md:text-xl font-semibold text-orange-900 mb-2">
292:                 {guestMetadata.loginPrompt.title}
293:             </h3>
294:             <p className="text-sm md:text-base text-orange-700 mb-6">
295:                 {guestMetadata.loginPrompt.message}
296:             </p>
297:             <div className="flex flex-col sm:flex-row gap-3 justify-center">
298:                 <Link to="/register" className="w-full sm:w-auto">
299:                     <Button className="w-full">
300:                         Create Free Account
301:                     </Button>
302:                 </Link>
303:                 <Link to="/login" className="w-full sm:w-auto">
304:                     <Button variant="secondary" className="w-full">
305:                         Sign In
306:                     </Button>
307:                 </Link>
308:             </div>
309:         </div>
310:     </Card>
311: )}
312:
313: {/* Feature Comparison */}
314: {!user && guestMetadata?.features && (
315:     <Card className="p-6">
316:         <h3 className="text-xl font-bold text-gray-900 mb-4">
317:             Unlock Full Features
318:         </h3>
319:         <div className="grid md:grid-cols-2 gap-4">
320:             <div>
321:                 <p className="font-semibold text-gray-700 mb-2">
322:                     Current (Guest):
323:                 </p>
324:                 <ul className="space-y-1 text-sm text-gray-600">
325:                     {guestMetadata.features.current_access.map(
326:                         (feature, i) => (
327:                             <li key={i}>? {feature}</li>
328:                         ),
329:                     )}
330:                 </ul>
331:             </div>
332:             <div className="bg-primary-50 rounded-lg p-4 border-2 border-primary-200">
333:                 <p className="font-semibold text-primary-900 mb-2">

```

```

334:             With Free Account:
335:         </p>
336:         <ul className="space-y-1 text-sm text-primary-700">
337:             {guestMetadata.features.with_account.map(
338:                 (feature, i) => (
339:                     <li key={i}>? {feature}</li>
340:                 ),
341:             )}
342:         </ul>
343:     </div>
344: </div>
345: </Card>
346: )}
347: </div>
348: ) : (
349:     <Card className="p-6 md:p-8 lg:p-10 bg-white shadow-xl">
350:         <ImageUploader
351:             onSelect={handleImageSelect}
352:             selectedImage={selectedImage}
353:             imagePreview={imagePreview}
354:             onRemove={handleRemoveImage}
355:         />
356:
357:         {selectedImage && (
358:             <motion.div
359:                 initial={{ opacity: 0, y: 20 }}
360:                 animate={{ opacity: 1, y: 0 }}
361:                 className="mt-8 flex justify-center"
362:             >
363:                 <Button
364:                     onClick={handleDiagnose}
365:                     size="lg"
366:                     icon={<Upload size={20} />}
367:                     className="w-full sm:w-auto"
368:                 >
369:                     Diagnose Plant
370:                 </Button>
371:             </motion.div>
372:         )}
373:
374:         <div className="mt-8 p-6 bg-gradient-to-br from-primary-50 via-forest-50 to-sage-50 rounded"
375:             <div className="flex items-center gap-2 mb-4">
376:                 <Info size={20} className="text-primary-600" />
377:                 <h3 className="font-semibold text-gray-900 text-base md:text-lg">
378:                     Tips for Best Results
379:                 </h3>
380:             </div>
381:             <ul className="space-y-3">
382:                 {tips.map((tip, index) => (
383:                     <li
384:                         key={index}
385:                         className="flex items-start gap-3 text-sm md:text-base text-gray-700"
386:                     >
387:                         <span className="text-primary-600 flex-shrink-0 mt-0.5">
388:                             {tip.icon}
389:                         </span>
390:                         <span>{tip.text}</span>
391:                     </li>
392:                 ))}
393:             </ul>
394:         </div>
395:     </Card>
396: )}
397: </motion.div>
398: </div>
399: </section>
400:
401: { /* Features Section */ }
402: <section className="py-16 md:py-20 lg:py-24 px-4 bg-white">
403:     <div className="max-w-7xl mx-auto">
404:         <div className="text-center mb-12 md:mb-16">
405:             <h2 className="text-3xl md:text-4xl lg:text-5xl font-bold text-gray-900 mb-4">
406:                 Why Use Plant Health Detector?

```



```

407:         </h2>
408:         <p className="text-base md:text-lg text-gray-600 max-w-2xl mx-auto">
409:             Advanced AI technology combined with user-friendly design for
410:             accurate plant diagnostics
411:         </p>
412:     </div>
413:
414:     <div className="grid sm:grid-cols-2 lg:grid-cols-3 gap-6 md:gap-8">
415:         {features.map((feature, index) => (
416:             <motion.div
417:                 key={index}
418:                 initial={{ opacity: 0, y: 20 }}
419:                 whileInView={{ opacity: 1, y: 0 }}
420:                 viewport={{ once: true }}
421:                 transition={{ delay: index * 0.1 }}
422:             >
423:                 <Card className="p-6 md:p-8 h-full bg-gradient-to-br from-white to-gray-50 border border-
424:                     <div className="bg-gradient-to-br from-primary-100 to-forest-100 p-3 md:p-4 rounded-xl
425:                         {feature.icon}
426:                     </div>
427:                     <h3 className="text-lg md:text-xl font-semibold text-gray-900 mb-3">
428:                         {feature.title}
429:                     </h3>
430:                     <p className="text-sm md:text-base text-gray-600 leading-relaxed">
431:                         {feature.description}
432:                     </p>
433:                 </Card>
434:             </motion.div>
435:         ))}
436:     </div>
437: </div>
438: </section>
439:
440: { /* CTA Section */}
441: {!user && (
442:     <section className="relative py-16 md:py-20 lg:py-24 px-4 bg-gradient-to-br from-primary-600 via-
443:         <div className="absolute inset-0 bg-[url('
444:
445:         <div className="max-w-4xl mx-auto text-center relative z-10">
446:             <History size={48} className="mx-auto mb-6 text-white" />
447:             <h2 className="text-3xl md:text-4xl lg:text-5xl font-bold text-white mb-6">
448:                 Want to Track Your Plants?
449:             </h2>
450:             <p className="text-lg md:text-xl text-primary-50 mb-8 md:mb-10 leading-relaxed">
451:                 Create a free account to save your diagnosis history and
452:                 re-download reports anytime.
453:             </p>
454:             <div className="flex flex-col sm:flex-row gap-4 justify-center">
455:                 <Link to="/register" className="w-full sm:w-auto">
456:                     <Button
457:                         size="lg"
458:                         className="w-full bg-white text-primary-600 hover:bg-gray-100"
459:                     >
460:                         Create Free Account
461:                     </Button>
462:                 </Link>
463:                 <Link to="/login" className="w-full sm:w-auto">
464:                     <Button
465:                         variant="secondary"
466:                         size="lg"
467:                         className="w-full bg-transparent border-2 border-white text-white hover:bg-white/10"
468:                     >
469:                         Sign In
470:                     </Button>
471:                 </Link>
472:             </div>
473:         </div>
474:     </section>
475: )}
476: </main>
477:
478: <Footer />
479: </div>

```

```

480:   });
481: };
482:
483: export default Home;

```

■ File: src\pages\Login.jsx

```

=====
1: import { Leaf } from 'lucide-react'
2: import { Link } from 'react-router-dom'
3: import { motion } from 'framer-motion'
4: import LoginForm from '../components/auth/LoginForm'
5: import Card from '../components/common/Card'
6:
7: const Login = () => {
8:   return (
9:     <div className="min-h-screen flex items-center justify-center p-4 bg-gradient-to-br from-primary-50 to-
10:     <div className="max-w-md w-full">
11:       <motion.div
12:         initial={{ opacity: 0, y: 20 }}
13:         animate={{ opacity: 1, y: 0 }}
14:         transition={{ duration: 0.5 }}
15:       >
16:         { /* Logo */ }
17:         <Link to="/" className="flex items-center justify-center gap-2 mb-8 group">
18:           <div className="bg-gradient-to-br from-primary-500 to-forest-600 p-3 rounded-xl group-hover:scale-105">
19:             <Leaf className="text-white" size={32} />
20:           </div>
21:           <span className="text-2xl font-bold bg-gradient-to-r from-primary-700 to-forest-700 bg-clip-text">
22:             Plant Health
23:           </span>
24:         </Link>
25:
26:         { /* Login Card */ }
27:         <Card className="p-8">
28:           <div className="text-center mb-8">
29:             <h1 className="text-3xl font-bold text-gray-900 mb-2">
30:               Welcome Back
31:             </h1>
32:             <p className="text-gray-600">
33:               Sign in to access your diagnosis history
34:             </p>
35:           </div>
36:
37:           <LoginForm />
38:         </Card>
39:
40:         { /* Back to Home */ }
41:         <div className="text-center mt-6">
42:           <Link to="/" className="text-gray-600 hover:text-primary-600 text-sm font-medium transition-colors">
43:             ? Back to Home
44:           </Link>
45:         </div>
46:       </motion.div>
47:     </div>
48:   </div>
49: )
50: }
51:
52: export default Login

```

■ File: src\pages\Profile.jsx

```

=====
1: // src/pages/Profile.jsx - ENHANCED WITH SESSION MANAGEMENT
2: import { useState, useEffect } from "react";
3: import {
4:   User,
5:   Mail,
6:   Calendar,
7:   LogOut,

```

```

8:   Shield,
9:   Trash2,
10:  Monitor,
11: } from "lucide-react";
12: import { motion } from "framer-motion";
13: import { useNavigate } from "react-router-dom";
14: import Navbar from "../components/layout/Navbar";
15: import Footer from "../components/layout/Footer";
16: import Card from "../components/common/Card";
17: import Button from "../components/common/Button";
18: import Modal from "../components/common/Modal";
19: import { useAuth } from "@/hooks/useAuth";
20: import { useToast } from "@/hooks/useToast";
21: import { authService } from "../services/authService";
22: import { historyService } from "../services/historyService";
23:
24: const Profile = () => {
25:   const { user, logout } = useAuth();
26:   const { addToast } = useToast();
27:   const navigate = useNavigate();
28:
29:   const [stats, setStats] = useState({
30:     totalSessions: 0,
31:     totalPredictions: 0,
32:   });
33:
34:   const [sessions, setSessions] = useState([]);
35:   const [loadingSessions, setLoadingSessions] = useState(false);
36:   const [showSessionsModal, setShowSessionsModal] = useState(false);
37:
38:   useEffect(() => {
39:     loadStats();
40:   }, []);
41:
42:   const loadStats = async () => {
43:     try {
44:       const data = await historyService.getSessions(1000, 0);
45:       setStats({
46:         totalSessions: data.total || 0,
47:         totalPredictions: data.sessions?.length || 0,
48:       });
49:     } catch (error) {
50:       console.error("Failed to load stats:", error);
51:     }
52:   };
53:
54:   const loadActiveSessions = async () => {
55:     setLoadingSessions(true);
56:     try {
57:       const data = await authService.getSessions();
58:       setSessions(data.sessions || []);
59:       setShowSessionsModal(true);
60:     } catch (error) {
61:       addToast("Failed to load sessions", "error");
62:     } finally {
63:       setLoadingSessions(false);
64:     }
65:   };
66:
67:   const handleRevokeSession = async (sessionId) => {
68:     if (!window.confirm("Are you sure you want to revoke this session?")) {
69:       return;
70:     }
71:
72:     try {
73:       await authService.revokeSession(sessionId);
74:       addToast("Session revoked successfully", "success");
75:
76:       // Reload sessions
77:       const data = await authService.getSessions();
78:       setSessions(data.sessions || []);
79:     } catch (error) {
80:       addToast("Failed to revoke session", "error");

```

```

81:     }
82:   };
83:
84:   const handleLogoutAll = async () => {
85:     if (!window.confirm("This will log you out from all devices. Continue?")) {
86:       return;
87:     }
88:
89:     try {
90:       await authService.logoutAll();
91:       addToast("Logged out from all devices", "success");
92:       navigate("/");
93:     } catch (error) {
94:       addToast("Failed to logout from all devices", "error");
95:     }
96:   };
97:
98:   const handleLogout = async () => {
99:     try {
100:       await logout();
101:       addToast("Logged out successfully", "success");
102:       navigate("/");
103:     } catch (error) {
104:       addToast("Failed to logout", "error");
105:     }
106:   };
107:
108:   const formatDate = (dateString) => {
109:     return new Date(dateString).toLocaleString("en-US", {
110:       year: "numeric",
111:       month: "short",
112:       day: "numeric",
113:       hour: "2-digit",
114:       minute: "2-digit",
115:     });
116:   };
117:
118:   return (
119:     <div className="min-h-screen flex flex-col">
120:       <Navbar />
121:
122:       <main className="flex-1 max-w-4xl mx-auto px-4 sm:px-6 lg:px-8 py-8">
123:         <motion.div
124:           initial={{ opacity: 0, y: 20 }}
125:           animate={{ opacity: 1, y: 0 }}
126:           className="mb-8"
127:         >
128:           <h1 className="text-4xl font-bold text-gray-900 mb-2">
129:             Profile Settings
130:           </h1>
131:           <p className="text-gray-600 text-lg">
132:             Manage your account and view your statistics
133:           </p>
134:         </motion.div>
135:
136:         { /* Profile Info */ }
137:         <Card className="p-8 mb-6">
138:           <div className="flex items-center gap-6 mb-8">
139:             <div className="w-24 h-24 rounded-full bg-gradient-to-br from-primary-500 to-forest-600 flex items-center justify-center">
140:               {user?.username?.charAt(0).toUpperCase()}
141:             </div>
142:             <div className="flex-1">
143:               <h2 className="text-3xl font-bold text-gray-900 mb-1">
144:                 {user?.username}
145:               </h2>
146:               <p className="text-gray-600 flex items-center gap-2">
147:                 <Mail size={16} />
148:                 {user?.email}
149:               </p>
150:             </div>
151:           </div>
152:
153:           <div className="grid grid-cols-1 md:grid-cols-2 gap-4">

```

```

154:         <div className="p-4 bg-gradient-to-br from-primary-50 to-primary-100 rounded-xl border-2 border-
155:             <p className="text-gray-600 text-sm font-medium mb-1">
156:                 Total Sessions
157:             </p>
158:             <p className="text-3xl font-bold text-primary-700">
159:                 {stats.totalSessions}
160:             </p>
161:         </div>
162:         <div className="p-4 bg-gradient-to-br from-forest-50 to-forest-100 rounded-xl border-2 border-f
163:             <p className="text-gray-600 text-sm font-medium mb-1">
164:                 Total Diagnoses
165:             </p>
166:             <p className="text-3xl font-bold text-forest-700">
167:                 {stats.totalPredictions}
168:             </p>
169:         </div>
170:     </div>
171: </Card>
172:
173:     { /* Account Information */ }
174:     <Card className="p-8 mb-6">
175:         <h3 className="text-xl font-bold text-gray-900 mb-6">
176:             Account Information
177:         </h3>
178:
179:         <div className="space-y-4">
180:             <div className="flex items-center justify-between p-4 bg-gray-50 rounded-xl">
181:                 <div className="flex items-center gap-3">
182:                     <User className="text-gray-600" size={20} />
183:                     <div>
184:                         <p className="text-sm text-gray-600">Username</p>
185:                         <p className="font-semibold text-gray-900">
186:                             {user?.username}
187:                         </p>
188:                     </div>
189:                 </div>
190:             </div>
191:
192:             <div className="flex items-center justify-between p-4 bg-gray-50 rounded-xl">
193:                 <div className="flex items-center gap-3">
194:                     <Mail className="text-gray-600" size={20} />
195:                     <div>
196:                         <p className="text-sm text-gray-600">Email Address</p>
197:                         <p className="font-semibold text-gray-900">{user?.email}</p>
198:                     </div>
199:                 </div>
200:             </div>
201:
202:             <div className="flex items-center justify-between p-4 bg-gray-50 rounded-xl">
203:                 <div className="flex items-center gap-3">
204:                     <Calendar className="text-gray-600" size={20} />
205:                     <div>
206:                         <p className="text-sm text-gray-600">Member Since</p>
207:                         <p className="font-semibold text-gray-900">
208:                             {user?.created_at &&
209:                                 new Date(user.created_at).toLocaleDateString("en-US", {
210:                                     year: "numeric",
211:                                     month: "long",
212:                                     day: "numeric",
213:                                 })}
214:                         </p>
215:                     </div>
216:                 </div>
217:             </div>
218:         </div>
219:     </Card>
220:
221:     { /* Security & Sessions */ }
222:     <Card className="p-8 mb-6">
223:         <h3 className="text-xl font-bold text-gray-900 mb-6">
224:             Security & Sessions
225:         </h3>
226:

```

```

227:     <div className="space-y-3">
228:         <Button
229:             variant="outline"
230:             className="w-full justify-start"
231:             icon={<Monitor size={20} />}
232:             onClick={loadActiveSessions}
233:             loading={loadingSessions}
234:         >
235:             View Active Sessions
236:         </Button>
237:
238:         <Button
239:             variant="outline"
240:             className="w-full justify-start"
241:             icon={<Shield size={20} />}
242:             onClick={handleLogoutAll}
243:         >
244:             Logout from All Devices
245:         </Button>
246:
247:         <Button
248:             variant="outline"
249:             className="w-full justify-start"
250:             icon={<LogOut size={20} />}
251:             onClick={handleLogout}
252:         >
253:             Sign Out
254:         </Button>
255:     </div>
256: </Card>
257:
258:     { /* Version Info */ }
259:     <div className="text-center mt-8 text-sm text-gray-500">
260:         <p>Plant Health Detector v2.0.0</p>
261:         <p>© 2024 All rights reserved</p>
262:     </div>
263: </main>
264:
265: { /* Active Sessions Modal */ }
266: <Modal
267:     isOpen={showSessionsModal}
268:     onClose={() => setShowSessionsModal(false)}
269:     title="Active Sessions"
270:     size="lg"
271: >
272:     {sessions.length === 0 ? (
273:         <div className="text-center py-8">
274:             <Shield size={48} className="mx-auto text-gray-400 mb-4" />
275:             <p className="text-gray-600">No active sessions</p>
276:         </div>
277:     ) : (
278:         <div className="space-y-3">
279:             {sessions.map((session) => (
280:                 <div
281:                     key={session.id}
282:                     className={`p-4 rounded-xl border-2 ${
283:                         session.isCurrent
284:                             ? "bg-primary-50 border-primary-300"
285:                             : "bg-gray-50 border-gray-200"
286:                     }`}
287:                 >
288:                     <div className="flex items-start justify-between">
289:                         <div className="flex-1">
290:                             <div className="flex items-center gap-2 mb-2">
291:                                 <Monitor size={16} className="text-gray-600" />
292:                                 <p className="font-semibold text-gray-900">
293:                                     {session.userAgent || "Unknown Device"}
294:                                 </p>
295:                                 {session.isCurrent && (
296:                                     <span className="px-2 py-1 bg-primary-600 text-white text-xs rounded-full">
297:                                         Current
298:                                     </span>
299:                                 )}

```

```

300:                 </div>
301:                 <div className="text-sm text-gray-600 space-y-1">
302:                     <p>IP: {session.ipAddress || "Unknown"}</p>
303:                     <p>Last used: {formatDate(session.lastUsedAt)}</p>
304:                     <p>Expires: {formatDate(session.expiresAt)}</p>
305:                 </div>
306:             </div>
307:             {!session.isCurrent && (
308:                 <button
309:                     onClick={() => handleRevokeSession(session.id)}
310:                     className="p-2 rounded-lg text-gray-400 hover:text-red-600 hover:bg-red-50 transition
311:                 >
312:                     <Trash2 size={18} />
313:                 </button>
314:             )}
315:         </div>
316:     </div>
317: ))}
318: </div>
319: )}
320: </Modal>
321:
322:     <Footer />
323: </div>
324: );
325: };
326:
327: export default Profile;

```

■ File: src\pages\Register.jsx

```

=====
1: import { Leaf } from 'lucide-react'
2: import { Link } from 'react-router-dom'
3: import { motion } from 'framer-motion'
4: import RegisterForm from '../components/auth/RegisterForm'
5: import Card from '../components/common/Card'
6:
7: const Register = () => {
8:     return (
9:         <div className="min-h-screen flex items-center justify-center p-4 bg-gradient-to-br from-primary-50 to-
10:         <div className="max-w-md w-full">
11:             <motion.div
12:                 initial={{ opacity: 0, y: 20 }}
13:                 animate={{ opacity: 1, y: 0 }}
14:                 transition={{ duration: 0.5 }}
15:             >
16:                 { /* Logo */ }
17:                 <Link to="/" className="flex items-center justify-center gap-2 mb-8 group">
18:                     <div className="bg-gradient-to-br from-primary-500 to-forest-600 p-3 rounded-xl group-hover:sc
19:                         <Leaf className="text-white" size={32} />
20:                     </div>
21:                     <span className="text-2xl font-bold bg-gradient-to-r from-primary-700 to-forest-700 bg-clip-text
22:                         Plant Health
23:                     </span>
24:                 </Link>
25:
26:                 { /* Register Card */ }
27:                 <Card className="p-8">
28:                     <div className="text-center mb-8">
29:                         <h1 className="text-3xl font-bold text-gray-900 mb-2">
30:                             Create Account
31:                         </h1>
32:                         <p className="text-gray-600">
33:                             Save your diagnosis history and track plant health over time
34:                         </p>
35:                     </div>
36:
37:                     <RegisterForm />
38:                 </Card>
39:
40:                 { /* Back to Home */ }

```

```

41:         <div className="text-center mt-6">
42:             <Link to="/" className="text-gray-600 hover:text-primary-600 text-sm font-medium transition-colors">
43:                 ? Back to Home
44:             </Link>
45:         </div>
46:     </motion.div>
47: </div>
48: </div>
49: )
50: }
51:
52: export default Register

```

■ File: src/services/api.js

```

=====
1: import axios from 'axios'
2:
3: const API_URL = import.meta.env.VITE_API_URL || 'http://localhost:5000/api'
4:
5: const api = axios.create({
6:   baseURL: API_URL,
7:   withCredentials: true,
8:   headers: {
9:     'Content-Type': 'application/json',
10:  },
11: })
12:
13: // Request interceptor - Add token to requests
14: api.interceptors.request.use(
15:   (config) => {
16:     const token = localStorage.getItem('accessToken')
17:     if (token) {
18:       config.headers.Authorization = `Bearer ${token}`
19:     }
20:     return config
21:   },
22:   (error) => Promise.reject(error)
23: )
24:
25: // Response interceptor - Handle token refresh
26: api.interceptors.response.use(
27:   (response) => response,
28:   async (error) => {
29:     const originalRequest = error.config
30:
31:     // Handle token expiration (401 with TOKEN_EXPIRED code)
32:     if (
33:       error.response?.status === 401 &&
34:       error.response?.data?.code === 'TOKEN_EXPIRED' &&
35:       !originalRequest._retry
36:     ) {
37:       originalRequest._retry = true
38:
39:       try {
40:         // Try to refresh the token
41:         const { data } = await axios.post(
42:           `${API_URL}/auth/refresh`,
43:           {},
44:           { withCredentials: true }
45:         )
46:
47:         // Store new access token
48:         localStorage.setItem('accessToken', data.accessToken)
49:
50:         // Update the failed request with new token
51:         originalRequest.headers.Authorization = `Bearer ${data.accessToken}`
52:
53:         // Retry the original request
54:         return api(originalRequest)
55:       } catch (refreshError) {
56:         // Refresh failed - clear token and redirect to login

```



```

57:         localStorage.removeItem('accessToken')
58:         window.location.href = '/login'
59:         return Promise.reject(refreshError)
60:     }
61: }
62:
63:     return Promise.reject(error)
64: }
65: )
66:
67: export default api

```

■ File: src\services\authService.js

```

=====
1: // src/services/authService.js - ENHANCED WITH SESSION MANAGEMENT
2: import api from "../api/axios";
3:
4: export const authService = {
5:     /**
6:      * Register new user
7:      */
8:     register: async (username, email, password) => {
9:         const response = await api.post("/auth/register", {
10:             username,
11:             email,
12:             password,
13:         });
14:         return response.data.data || response.data;
15:     },
16:
17:     /**
18:      * Login user
19:      */
20:     login: async (email, password) => {
21:         const response = await api.post("/auth/login", { email, password });
22:
23:         // Backend returns: { success: true, data: { accessToken, expiresIn, user } }
24:         const { accessToken, expiresIn, user } =
25:             response.data.data || response.data;
26:
27:         // Store token with expiry
28:         localStorage.setItem("accessToken", accessToken);
29:         localStorage.setItem("tokenExpiry", Date.now() + (expiresIn || 900) * 1000);
30:
31:         return { user, accessToken, expiresIn };
32:     },
33:
34:     /**
35:      * Logout user
36:      */
37:     logout: async () => {
38:         try {
39:             await api.post("/auth/logout");
40:         } finally {
41:             localStorage.removeItem("accessToken");
42:             localStorage.removeItem("tokenExpiry");
43:         }
44:     },
45:
46:     /**
47:      * Logout from all devices
48:      */
49:     logoutAll: async () => {
50:         try {
51:             const response = await api.post("/auth/logout-all");
52:             return response.data.data || response.data;
53:         } finally {
54:             localStorage.removeItem("accessToken");
55:             localStorage.removeItem("tokenExpiry");
56:         }
57:     },

```

```

58:
59:  /**
60:   * Get current user profile
61:   */
62:  getProfile: async () => {
63:    const response = await api.get("/auth/profile");
64:    return response.data.data?.user || response.data.user;
65:  },
66:
67:  /**
68:   * Refresh access token
69:   */
70:  refreshToken: async () => {
71:    const response = await api.post("/auth/refresh");
72:    const { accessToken, expiresIn } = response.data.data || response.data;
73:
74:    localStorage.setItem("accessToken", accessToken);
75:    localStorage.setItem("tokenExpiry", Date.now() + (expiresIn || 900) * 1000);
76:
77:    return { accessToken, expiresIn };
78:  },
79:
80:  /**
81:   * ENHANCEMENT: Get all active sessions
82:   */
83:  getSessions: async () => {
84:    const response = await api.get("/auth/sessions");
85:    return response.data.data || response.data;
86:  },
87:
88:  /**
89:   * ENHANCEMENT: Revoke specific session
90:   */
91:  revokeSession: async (sessionId) => {
92:    const response = await api.delete(`/auth/sessions/${sessionId}`);
93:    return response.data;
94:  },
95:
96:  /**
97:   * Check if token is expired
98:   */
99:  isTokenExpired: () => {
100:    const expiry = localStorage.getItem("tokenExpiry");
101:    if (!expiry) return true;
102:
103:    return Date.now() >= parseInt(expiry);
104:  },
105:
106:  /**
107:   * Get time until token expires (in seconds)
108:   */
109:  getTimeUntilExpiry: () => {
110:    const expiry = localStorage.getItem("tokenExpiry");
111:    if (!expiry) return 0;
112:
113:    const remaining = parseInt(expiry) - Date.now();
114:    return Math.max(0, Math.floor(remaining / 1000));
115:  },
116: };

```

■ File: src/services/diagnosisService.js

```

=====
1: // src/services/diagnosisService.js - FULLY ALIGNED WITH BACKEND
2: import api from "../api/axios";
3:
4: export const diagnosisService = {
5:   /**
6:    * GUEST MODE - No authentication required
7:    * POST /api/guest/predict
8:    */
9:   diagnoseStateless: async (file) => {

```

```

10:     const formData = new FormData();
11:     formData.append("image", file);
12:
13:     const response = await api.post("/guest/predict", formData, {
14:       headers: { "Content-Type": "multipart/form-data" },
15:     });
16:
17:     return response.data.data || response.data;
18:   },
19:
20:   /**
21:    * GUEST PDF DOWNLOAD - Returns feature gate error
22:    * POST /api/guest/predict/download-pdf
23:    */
24:   downloadGuestPDF: async (prediction) => {
25:     const response = await api.post(
26:       "/guest/predict/download-pdf",
27:       { prediction },
28:       { responseType: "blob" },
29:     );
30:
31:     // This will throw 403 with login prompt from backend
32:     const blob = new Blob([response.data], { type: "application/pdf" });
33:     const url = window.URL.createObjectURL(blob);
34:     const link = document.createElement("a");
35:     link.href = url;
36:     link.download = `plant_diagnosis_${Date.now()}.pdf`;
37:     document.body.appendChild(link);
38:     link.click();
39:     link.remove();
40:     window.URL.revokeObjectURL(url);
41:
42:     return true;
43:   },
44:
45:   /**
46:    * AUTHENTICATED MODE - Saves to history
47:    * POST /api/predict
48:    */
49:   diagnoseWithHistory: async (file, sessionId = null) => {
50:     const formData = new FormData();
51:     formData.append("image", file);
52:     if (sessionId) {
53:       formData.append("session_id", sessionId);
54:     }
55:
56:     const response = await api.post("/predict", formData, {
57:       headers: { "Content-Type": "multipart/form-data" },
58:     });
59:
60:     return response.data.data || response.data;
61:   },
62:
63:   /**
64:    * Create new session
65:    * POST /api/predict/session
66:    */
67:   createSession: async (title = null) => {
68:     const response = await api.post("/predict/session", { title });
69:     return response.data.data || response.data;
70:   },
71:
72:   /**
73:    * Check AI health
74:    * GET /api/predict/health
75:    */
76:   checkHealth: async () => {
77:     const response = await api.get("/predict/health");
78:     return response.data;
79:   },
80: };

```

■ File: src/services/historyService.js

```
=====
1: // =====
2: // 4. src/services/historyService.js - ALIGNED
3: // =====
4: import api from "../api/axios";
5:
6: export const historyService = {
7:   getSessions: async (limit = 50, offset = 0) => {
8:     const response = await api.get("/history/sessions", {
9:       params: { limit, offset },
10:    });
11:    return response.data.data || response.data;
12:  },
13:
14:   getSessionPredictions: async (sessionId, limit = 100, offset = 0) => {
15:     const response = await api.get(
16:       `/history/sessions/${sessionId}/predictions`,
17:       { params: { limit, offset } }
18:     );
19:     return response.data.data || response.data;
20:   },
21:
22:   updateSessionTitle: async (sessionId, title) => {
23:     const response = await api.patch(`/history/sessions/${sessionId}`, {
24:       title,
25:     });
26:     return response.data.data || response.data;
27:   },
28:
29:   deleteSession: async (sessionId) => {
30:     const response = await api.delete(`/history/sessions/${sessionId}`);
31:     return response.data;
32:   },
33:
34:   getPredictionDetails: async (predictionId) => {
35:     const response = await api.get(`/history/predictions/${predictionId}`);
36:     return response.data.data || response.data;
37:   },
38: };
=====
```

■ File: src/services/systemService.js

```
=====
1: // src/services/systemService.js - NEW FILE
2: // Comprehensive system monitoring service aligned with backend
3:
4: import api from "../api/axios";
5:
6: export const systemService = {
7:   /**
8:    * Get complete system health
9:    * GET /api/system/health
10:   */
11:   getHealth: async () => {
12:     const response = await api.get("/system/health");
13:     return response.data.data || response.data;
14:   },
15:
16:   /**
17:    * Get Redis connection status
18:    * GET /api/system/redis-status
19:   */
20:   getRedisStatus: async () => {
21:     const response = await api.get("/system/redis-status");
22:     return response.data.data || response.data;
23:   },
24:
25:   /**
26:    * Get guest rate limiting statistics
27:    * GET /api/system/guest-stats
28:   */
=====
```

```

29:   getGuestStats: async () => {
30:     const response = await api.get("/system/guest-stats");
31:     return response.data.data || response.data;
32:   },
33:
34:   /**
35:    * Clear guest memory store (requires auth)
36:    * POST /api/system/clear-guest-store
37:    */
38:   clearGuestStore: async () => {
39:     const response = await api.post("/system/clear-guest-store");
40:     return response.data;
41:   },
42:
43:   /**
44:    * Get database statistics (requires auth)
45:    * GET /api/system/database-stats
46:    */
47:   getDatabaseStats: async () => {
48:     const response = await api.get("/system/database-stats");
49:     return response.data.data || response.data;
50:   },
51:
52:   /**
53:    * Get storage statistics (requires auth)
54:    * GET /api/system/storage-stats
55:    */
56:   getStorageStats: async () => {
57:     const response = await api.get("/system/storage-stats");
58:     return response.data.data || response.data;
59:   },
60:
61:   /**
62:    * Get CSP violation statistics (requires auth)
63:    * GET /api/system/csp-stats
64:    */
65:   getCspStats: async () => {
66:     const response = await api.get("/system/csp-stats");
67:     return response.data.data || response.data;
68:   },
69:
70:   /**
71:    * Test guest rate limiting
72:    * POST /api/system/test-guest-limit
73:    */
74:   testGuestLimit: async () => {
75:     const response = await api.post("/system/test-guest-limit");
76:     return response.data.data || response.data;
77:   },
78: };

```

■ File: src\utils\constants.js

```

=====
1: // src/utils/constants.js - FULLY ALIGNED WITH BACKEND
2: export const API_URL =
3:   import.meta.env.VITE_API_URL || "http://localhost:5000/api";
4:
5: // =====
6: // EXACT MATCH: Backend Error Codes from src/config/constants.js
7: // =====
8: export const ERROR_CODES = {
9:   // Authentication Errors
10:  TOKEN_EXPIRED: "TOKEN_EXPIRED",
11:  REFRESH_TOKEN_EXPIRED: "REFRESH_TOKEN_EXPIRED",
12:  TOKEN_REVOKED: "TOKEN_REVOKED",
13:  INVALID_TOKEN: "INVALID_TOKEN",
14:  TOKEN_REQUIRED: "TOKEN_REQUIRED",
15:  REFRESH_TOKEN_REQUIRED: "REFRESH_TOKEN_REQUIRED",
16:
17:  // User Errors
18:  USER_NOT_FOUND: "USER_NOT_FOUND",

```

```

19:   INVALID_CREDENTIALS: "INVALID_CREDENTIALS",
20:   ACCOUNT_INACTIVE: "ACCOUNT_INACTIVE",
21:   EMAIL_EXISTS: "EMAIL_EXISTS",
22:   USERNAME_EXISTS: "USERNAME_EXISTS",
23:
24:   // AI Service Errors
25:   AI_SERVICE_UNAVAILABLE: "AI_SERVICE_UNAVAILABLE",
26:   AI_SERVICE_TIMEOUT: "AI_SERVICE_TIMEOUT",
27:   AI_PREDICTION_FAILED: "AI_PREDICTION_FAILED",
28:   MODEL_NOT_LOADED: "MODEL_NOT_LOADED",
29:   AI_OUTPUT_TOO_LARGE: "AI_OUTPUT_TOO_LARGE",
30:
31:   // File Upload Errors
32:   INVALID_FILE_TYPE: "INVALID_FILE_TYPE",
33:   INVALID_FILE_EXTENSION: "INVALID_FILE_EXTENSION",
34:   FILE_TOO_LARGE: "FILE_TOO_LARGE",
35:   FILE_REQUIRED: "FILE_REQUIRED",
36:   INVALID_IMAGE: "INVALID_IMAGE",
37:   INVALID_FILENAME: "INVALID_FILENAME",
38:   PATH_TRAVERSAL_ATTEMPT: "PATH_TRAVERSAL_ATTEMPT",
39:
40:   // Session/History Errors
41:   SESSION_NOT_FOUND: "SESSION_NOT_FOUND",
42:   PREDICTION_NOT_FOUND: "PREDICTION_NOT_FOUND",
43:   INVALID_SESSION_ID: "INVALID_SESSION_ID",
44:   INVALID_PREDICTION_ID: "INVALID_PREDICTION_ID",
45:
46:   // Validation Errors
47:   VALIDATION_ERROR: "VALIDATION_ERROR",
48:   INVALID_INPUT: "INVALID_INPUT",
49:   PREDICTION_REQUIRED: "PREDICTION_REQUIRED",
50:
51:   // Rate Limiting - COMPLETE SET
52:   RATE_LIMIT_EXCEEDED: "RATE_LIMIT_EXCEEDED",
53:   AUTH_RATE_LIMIT_EXCEEDED: "AUTH_RATE_LIMIT_EXCEEDED",
54:   HEALTH_RATE_LIMIT_EXCEEDED: "HEALTH_RATE_LIMIT_EXCEEDED",
55:   GUEST_RATE_LIMIT_EXCEEDED: "GUEST_RATE_LIMIT_EXCEEDED",
56:   GUEST_IP_LIMIT_EXCEEDED: "GUEST_IP_LIMIT_EXCEEDED",
57:   GUEST_DAILY_LIMIT_EXCEEDED: "GUEST_DAILY_LIMIT_EXCEEDED",
58:   GUEST_SESSION_LIMIT_EXCEEDED: "GUEST_SESSION_LIMIT_EXCEEDED",
59:   REFRESH_RATE_LIMIT_EXCEEDED: "REFRESH_RATE_LIMIT_EXCEEDED",
60:   HISTORY_RATE_LIMIT_EXCEEDED: "HISTORY_RATE_LIMIT_EXCEEDED",
61:
62:   // Security
63:   SUSPICIOUS_ACTIVITY_DETECTED: "SUSPICIOUS_ACTIVITY_DETECTED",
64:   FEATURE_REQUIRES_AUTHENTICATION: "FEATURE_REQUIRES_AUTHENTICATION",
65:
66:   // Generic
67:   INTERNAL_ERROR: "INTERNAL_ERROR",
68:   NOT_FOUND: "NOT_FOUND",
69:   FORBIDDEN: "FORBIDDEN",
70:   UNAUTHORIZED: "UNAUTHORIZED",
71: };
72:
73: // =====
74: // User-Friendly Error Messages
75: // =====
76: export const ERROR_MESSAGES = {
77:   [ERROR_CODES.TOKEN_EXPIRED]: "Your session expired. Please log in again.",
78:   [ERROR_CODES.REFRESH_TOKEN_EXPIRED]: "Session expired. Please log in again.",
79:   [ERROR_CODES.INVALID_CREDENTIALS]: "Invalid email or password.",
80:   [ERROR_CODES.EMAIL_EXISTS]: "This email is already registered.",
81:   [ERROR_CODES.USERNAME_EXISTS]: "Username already taken.",
82:   [ERROR_CODES.AI_SERVICE_UNAVAILABLE]:
83:     "AI service temporarily unavailable. Please try again.",
84:   [ERROR_CODES.AI_PREDICTION_FAILED]:
85:     "Failed to analyze image. Please try again.",
86:   [ERROR_CODES.INVALID_FILE_TYPE]: "Invalid file type. Use JPG or PNG images.",
87:   [ERROR_CODES.FILE_TOO_LARGE]: "File too large. Maximum size is 5MB.",
88:   [ERROR_CODES.FILE_REQUIRED]: "Please select an image to upload.",
89:   [ERROR_CODES.RATE_LIMIT_EXCEEDED]: "Too many requests. Please wait a moment.",
90:   [ERROR_CODES.GUEST_RATE_LIMIT_EXCEEDED]:
91:     "Free prediction limit reached. Create an account to continue.",

```

```

92:   [ERROR_CODES.AUTH_RATE_LIMIT_EXCEEDED]:
93:     "Too many login attempts. Please try again later.",
94:   [ERROR_CODES.PATH_TRAVERSAL_ATTEMPT]: "Invalid file name detected.",
95:   [ERROR_CODES.INVALID_IMAGE]: "Invalid image file.",
96:   [ERROR_CODES.SUSPICIOUS_ACTIVITY_DETECTED]:
97:     "Too many rapid requests. Please slow down or create an account.",
98:   [ERROR_CODES.FEATURE_REQUIRES_AUTHENTICATION]:
99:     "This feature requires an account. Sign up for free!",
100: };
101:
102: // =====
103: // File Upload Configuration (matches backend)
104: // =====
105: export const MAX_FILE_SIZE = 5 * 1024 * 1024; // 5MB
106: export const ALLOWED_FILE_TYPES = ["image/jpeg", "image/png", "image/jpg"];
107: export const ALLOWED_EXTENSIONS = ["jpg", "jpeg", "png"];
108:
109: // =====
110: // Model Information (EXACT match to backend)
111: // =====
112: export const MODEL_NAME = "efficientnet_b2";
113: export const MODEL_VERSION = "v1.0.0";
114: export const MODEL_INPUT_SIZE = 224;
115:
116: export const MODEL_CLASSES = [
117:   "Healthy",
118:   "Pest_Fungal",
119:   "Pest_Bacterial",
120:   "Pest_Insect",
121:   "Nutrient_Nitrogen",
122:   "Nutrient_Potassium",
123:   "Water_Stress",
124:   "Not_Plant", // CRITICAL: Added Not_Plant class
125: ];
126:
127: // =====
128: // Confidence Levels (matches backend exactly)
129: // =====
130: export const getConfidenceLevel = (confidence) => {
131:   if (confidence >= 0.85)
132:     return { label: "Very High", color: "text-green-700" };
133:   if (confidence >= 0.7) return { label: "High", color: "text-green-600" };
134:   if (confidence >= 0.55)
135:     return { label: "Moderate", color: "text-yellow-600" };
136:   return { label: "Low", color: "text-red-600" };
137: };
138:
139: // =====
140: // COMPLETE Disease Information (with Not_Plant support)
141: // =====
142: export const DISEASE_INFO = {
143:   // CRITICAL: Not_Plant detection
144:   Not_Plant: {
145:     title: "Not a Plant",
146:     color: "text-gray-700",
147:     bgColor: "bg-gray-50",
148:     borderColor: "border-gray-300",
149:     description:
150:       "This image does not appear to be a plant. Please upload a clear photo of a plant's leaves or affected",
151:     icon: "?",
152:     recommendations: [
153:       "Upload a clear image of plant leaves",
154:       "Ensure the plant is the main subject",
155:       "Avoid images of animals, people, or objects",
156:       "Use good lighting and focus on the plant",
157:     ],
158:   },
159:
160:   Healthy: {
161:     title: "Healthy Plant",
162:     color: "text-green-600",
163:     bgColor: "bg-green-50",
164:     borderColor: "border-green-200",

```

```

165:     description: "Your plant is in excellent health!",
166:     icon: "?",
167:     recommendations: [
168:         "Continue current care routine",
169:         "Monitor regularly for changes",
170:         "Maintain proper watering and light",
171:         "Ensure good air circulation",
172:     ],
173: },
174:
175: Pest_Fungal: {
176:     title: "Fungal Infection",
177:     color: "text-orange-600",
178:     bgColor: "bg-orange-50",
179:     borderColor: "border-orange-200",
180:     description: "Fungal infection detected. Immediate treatment recommended.",
181:     icon: "?",
182:     recommendations: [
183:         "Apply fungicide immediately (copper-based or organic)",
184:         "Remove affected leaves and dispose properly",
185:         "Improve air circulation around plant",
186:         "Reduce watering frequency and avoid overhead watering",
187:         "Isolate plant to prevent spread",
188:     ],
189: },
190:
191: Pest_Bacterial: {
192:     title: "Bacterial Infection",
193:     color: "text-red-600",
194:     bgColor: "bg-red-50",
195:     borderColor: "border-red-200",
196:     description: "Bacterial infection detected. Quick action required.",
197:     icon: "?",
198:     recommendations: [
199:         "Isolate affected plant immediately",
200:         "Apply copper-based bactericide",
201:         "Remove and destroy infected parts",
202:         "Sterilize tools after use",
203:         "Avoid overhead watering",
204:     ],
205: },
206:
207: Pest_Insect: {
208:     title: "Insect Infestation",
209:     color: "text-yellow-700",
210:     bgColor: "bg-yellow-50",
211:     borderColor: "border-yellow-200",
212:     description: "Insect pests detected on your plant.",
213:     icon: "?",
214:     recommendations: [
215:         "Identify specific insect pest",
216:         "Apply appropriate insecticide or use neem oil",
217:         "Check for eggs under leaves",
218:         "Remove visible pests manually",
219:         "Introduce beneficial insects if possible",
220:     ],
221: },
222:
223: Nutrient_Nitrogen: {
224:     title: "Nitrogen Deficiency",
225:     color: "text-blue-600",
226:     bgColor: "bg-blue-50",
227:     borderColor: "border-blue-200",
228:     description: "Plant shows signs of nitrogen deficiency.",
229:     icon: "?",
230:     recommendations: [
231:         "Apply nitrogen-rich fertilizer",
232:         "Use compost or well-rotted manure",
233:         "Monitor leaf color changes",
234:         "Ensure proper soil pH (6.0-7.0)",
235:         "Consider foliar feeding for quick results",
236:     ],
237: },

```



```

238:
239:   Nutrient_Potassium: {
240:     title: "Potassium Deficiency",
241:     color: "text-purple-600",
242:     bgColor: "bg-purple-50",
243:     borderColor: "border-purple-200",
244:     description: "Potassium deficiency detected.",
245:     icon: "?",
246:     recommendations: [
247:       "Apply potassium fertilizer",
248:       "Use banana peels or wood ash as natural source",
249:       "Monitor leaf edges for browning",
250:       "Avoid over-fertilization with nitrogen",
251:       "Check soil drainage",
252:     ],
253:   },
254:
255:   Water_Stress: {
256:     title: "Water Stress",
257:     color: "text-cyan-600",
258:     bgColor: "bg-cyan-50",
259:     borderColor: "border-cyan-200",
260:     description: "Plant is experiencing water stress.",
261:     icon: "?",
262:     recommendations: [
263:       "Adjust watering schedule based on soil moisture",
264:       "Check soil moisture regularly (finger test)",
265:       "Ensure proper drainage to prevent waterlogging",
266:       "Consider mulching to retain moisture",
267:       "Water deeply but less frequently",
268:     ],
269:   },
270: };
271:
272: // =====
273: // Guest Mode Configuration (matches backend)
274: // =====
275: export const GUEST_LIMITS = {
276:   IP_LIMIT: 5, // Per hour
277:   FINGERPRINT_LIMIT: 10, // Per day
278:   SESSION_LIMIT: 4, // Per session
279:   PREDICTION_LIMIT: 10, // Daily limit per IP
280: };
281:
282: // =====
283: // Feature Flags
284: // =====
285: export const FEATURES = {
286:   GUEST_MODE: true,
287:   OPTIONAL_LOGIN: true,
288:   PDF_DOWNLOAD: true,
289:   HISTORY_TRACKING: true,
290:   ANALYTICS: false,
291: };

```

File: src\utils\helpers.js

```

=====
1: // src/utils/helpers.js - COMPLETE HELPER UTILITIES
2:
3: /**
4:  * Format date to human-readable string
5:  */
6: export const formatDate = (dateString) => {
7:   const date = new Date(dateString);
8:   const now = new Date();
9:   const diffTime = Math.abs(now - date);
10:  const diffDays = Math.ceil(diffTime / (1000 * 60 * 60 * 24));
11:
12:  if (diffDays === 0) {
13:    return "Today";
14:  } else if (diffDays === 1) {

```

```

15:     return "Yesterday";
16:   } else if (diffDays < 7) {
17:     return `${diffDays} days ago`;
18:   } else {
19:     return date.toLocaleDateString("en-US", {
20:       month: "short",
21:       day: "numeric",
22:       year: "numeric",
23:     });
24:   }
25: };
26:
27: /**
28:  * Format date with time
29:  */
30: export const formatDate = (dateString) => {
31:   const date = new Date(dateString);
32:   return date.toLocaleString("en-US", {
33:     month: "short",
34:     day: "numeric",
35:     year: "numeric",
36:     hour: "2-digit",
37:     minute: "2-digit",
38:   });
39: };
40:
41: /**
42:  * Truncate text to specified length
43:  */
44: export const truncateText = (text, maxLength = 50) => {
45:   if (text.length <= maxLength) return text;
46:   return text.substring(0, maxLength) + "...";
47: };
48:
49: /**
50:  * Get pagination metadata
51:  */
52: export const getPaginationMetadata = (total, limit, offset) => {
53:   return {
54:     total,
55:     limit,
56:     offset,
57:     hasMore: offset + limit < total,
58:     currentPage: Math.floor(offset / limit) + 1,
59:     totalPages: Math.ceil(total / limit),
60:   };
61: };
62:
63: /**
64:  * Format file size to human-readable
65:  */
66: export const formatFileSize = (bytes) => {
67:   if (bytes === 0) return "0 Bytes";
68:
69:   const k = 1024;
70:   const sizes = ["Bytes", "KB", "MB", "GB"];
71:   const i = Math.floor(Math.log(bytes) / Math.log(k));
72:
73:   return Math.round((bytes / Math.pow(k, i)) * 100) / 100 + " " + sizes[i];
74: };
75:
76: /**
77:  * Debounce function
78:  */
79: export const debounce = (func, wait) => {
80:   let timeout;
81:   return function executedFunction(...args) {
82:     const later = () => {
83:       clearTimeout(timeout);
84:       func(...args);
85:     };
86:     clearTimeout(timeout);
87:     timeout = setTimeout(later, wait);

```

```

88:   };
89: };
90:
91: /**
92:  * Deep clone object
93:  */
94: export const deepClone = (obj) => {
95:   return JSON.parse(JSON.stringify(obj));
96: };
97:
98: /**
99:  * Check if object is empty
100:  */
101: export const isEmpty = (obj) => {
102:   return Object.keys(obj).length === 0;
103: };
104:
105: /**
106:  * Capitalize first letter
107:  */
108: export const capitalize = (str) => {
109:   return str.charAt(0).toUpperCase() + str.slice(1);
110: };
111:
112: /**
113:  * Generate random ID
114:  */
115: export const generateId = () => {
116:   return Date.now().toString(36) + Math.random().toString(36).substr(2);
117: };
118:
119: /**
120:  * Sleep/delay utility
121:  */
122: export const sleep = (ms) => {
123:   return new Promise((resolve) => setTimeout(resolve, ms));
124: };
125:
126: /**
127:  * Format confidence percentage
128:  */
129: export const formatConfidence = (confidence) => {
130:   return `${(confidence * 100).toFixed(1)}%`;
131: };
132:
133: /**
134:  * Parse error message from API response
135:  */
136: export const parseErrorMessage = (error) => {
137:   if (error.response?.data?.error) {
138:     return error.response.data.error;
139:   }
140:   if (error.response?.data?.message) {
141:     return error.response.data.message;
142:   }
143:   if (error.message) {
144:     return error.message;
145:   }
146:   return "An unexpected error occurred";
147: };
148:
149: /**
150:  * Check if user is authenticated
151:  */
152: export const isAuthenticated = () => {
153:   return !!localStorage.getItem("accessToken");
154: };
155:
156: /**
157:  * Format duration in milliseconds to readable string
158:  */
159: export const formatDuration = (ms) => {
160:   if (ms < 1000) return `${ms}ms`;

```

```

161:   if (ms < 60000) return `${(ms / 1000).toFixed(1)}s`;
162:   return `${Math.floor(ms / 60000)}m ${Math.floor((ms % 60000) / 1000)}s`;
163: };
164:
165: /**
166:  * Group array by key
167:  */
168: export const groupBy = (array, key) => {
169:   return array.reduce((result, item) => {
170:     const group = item[key];
171:     if (!result[group]) {
172:       result[group] = [];
173:     }
174:     result[group].push(item);
175:     return result;
176:   }, {});
177: };
178:
179: /**
180:  * Sort array by key
181:  */
182: export const sortBy = (array, key, order = "asc") => {
183:   return [...array].sort((a, b) => {
184:     if (order === "asc") {
185:       return a[key] > b[key] ? 1 : -1;
186:     }
187:     return a[key] < b[key] ? 1 : -1;
188:   });
189: };
190:
191: /**
192:  * Filter array by search term
193:  */
194: export const filterBySearch = (array, searchTerm, keys) => {
195:   const term = searchTerm.toLowerCase();
196:   return array.filter((item) =>
197:     keys.some((key) => item[key]?.toLowerCase().includes(term)),
198:   );
199: };
200:
201: /**
202:  * Calculate percentage
203:  */
204: export const calculatePercentage = (value, total) => {
205:   if (total === 0) return 0;
206:   return Math.round((value / total) * 100);
207: };
208:
209: /**
210:  * Validate image file
211:  */
212: export const isValidImage = (file) => {
213:   const validTypes = ["image/jpeg", "image/jpg", "image/png"];
214:   const maxSize = 5 * 1024 * 1024; // 5MB
215:
216:   return validTypes.includes(file.type) && file.size <= maxSize;
217: };
218:
219: /**
220:  * Copy text to clipboard
221:  */
222: export const copyToClipboard = async (text) => {
223:   try {
224:     await navigator.clipboard.writeText(text);
225:     return true;
226:   } catch (error) {
227:     console.error("Failed to copy:", error);
228:     return false;
229:   }
230: };
231:
232: /**
233:  * Download data as JSON file

```

```

234: */
235: export const downloadJSON = (data, filename = "data.json") => {
236:   const blob = new Blob([JSON.stringify(data, null, 2)], {
237:     type: "application/json",
238:   });
239:   const url = window.URL.createObjectURL(blob);
240:   const link = document.createElement("a");
241:   link.href = url;
242:   link.download = filename;
243:   document.body.appendChild(link);
244:   link.click();
245:   document.body.removeChild(link);
246:   window.URL.revokeObjectURL(url);
247: };

```

File: src\utils\pdfGenerator.js

```

=====
1: import jsPDF from 'jspdf'
2:
3: export const generatePDFReport = async (prediction, imageSrc) => {
4:   const pdf = new jsPDF('p', 'mm', 'a4')
5:   const pageWidth = pdf.internal.pageSize.getWidth()
6:   const pageHeight = pdf.internal.pageSize.getHeight()
7:
8:   // Add title
9:   pdf.setFontSize(24)
10:  pdf.setTextColor(34, 197, 94) // Primary green
11:  pdf.text('Plant Health Report', pageWidth / 2, 20, { align: 'center' })
12:
13:  // Add date
14:  pdf.setFontSize(10)
15:  pdf.setTextColor(100, 100, 100)
16:  pdf.text(`Generated: ${new Date().toLocaleDateString()}`, pageWidth / 2, 28, { align: 'center' })
17:
18:  // Add image (if available)
19:  if (imageSrc) {
20:    try {
21:      const imgData = await loadImage(imageSrc)
22:      const imgWidth = 80
23:      const imgHeight = 60
24:      const imgX = (pageWidth - imgWidth) / 2
25:      pdf.drawImage(imgData, 'JPEG', imgX, 35, imgWidth, imgHeight)
26:    } catch (error) {
27:      console.error('Error adding image to PDF:', error)
28:    }
29:  }
30:
31:  let yPos = imageSrc ? 105 : 40
32:
33:  // Diagnosis Result
34:  pdf.setFontSize(16)
35:  pdf.setTextColor(0, 0, 0)
36:  pdf.text('Diagnosis Result', 15, yPos)
37:  yPos += 10
38:
39:  pdf.setFontSize(12)
40:  pdf.setTextColor(50, 50, 50)
41:  pdf.text(`Condition: ${prediction.class.replace(/_/g, ' ')}`, 15, yPos)
42:  yPos += 7
43:  pdf.text(`Confidence: ${((prediction.confidence * 100).toFixed(1))}%`, 15, yPos)
44:  yPos += 7
45:  if (prediction.category) {
46:    pdf.text(`Category: ${prediction.category}`, 15, yPos)
47:    yPos += 7
48:  }
49:  if (prediction.subtype) {
50:    pdf.text(`Subtype: ${prediction.subtype}`, 15, yPos)
51:    yPos += 7
52:  }
53:  yPos += 3
54:

```

```

55: // Recommendations
56: if (prediction.explanation) {
57:   pdf.setFontSize(14)
58:   pdf.setTextColor(0, 0, 0)
59:   pdf.text('Recommendations', 15, yPos)
60:   yPos += 8
61:
62:   pdf.setFontSize(10)
63:   pdf.setTextColor(50, 50, 50)
64:   const splitText = pdf.splitTextToSize(prediction.explanation, pageWidth - 30)
65:   pdf.text(splitText, 15, yPos)
66:   yPos += splitText.length * 5
67: }
68:
69: // Model info
70: yPos += 10
71: pdf.setFontSize(9)
72: pdf.setTextColor(120, 120, 120)
73: pdf.text(`Model: ${prediction.model_name || 'efficientnet_b2'}`, 15, yPos)
74: yPos += 5
75: pdf.text(`Processing Time: ${prediction.processing_time_ms || 'N/A'}ms`, 15, yPos)
76:
77: // Footer
78: pdf.setFontSize(8)
79: pdf.setTextColor(150, 150, 150)
80: pdf.text('Plant Health Detector - AI-Powered Diagnosis', pageWidth / 2, pageHeight - 10, { align: 'center' })
81: pdf.text('This is an automated diagnosis. Consult experts for critical decisions.', pageWidth / 2, pageHeight - 5, { align: 'center' })
82:
83: // Download
84: pdf.save(`plant-diagnosis-${Date.now()}.pdf`)
85: }
86:
87: const loadImage = (src) => {
88:   return new Promise((resolve, reject) => {
89:     const img = new Image()
90:     img.crossOrigin = 'Anonymous'
91:     img.onload = () => {
92:       const canvas = document.createElement('canvas')
93:       canvas.width = img.width
94:       canvas.height = img.height
95:       const ctx = canvas.getContext('2d')
96:       ctx.drawImage(img, 0, 0)
97:       resolve(canvas.toDataURL('image/jpeg', 0.8))
98:     }
99:     img.onerror = reject
100:    img.src = src
101:  })
102: }

```

File: src\utils\validators.js

```

=====
1: // src/utils/validators.js - EXACT MATCH TO BACKEND VALIDATION
2:
3: /**
4:  * Validate email format
5:  * MATCHES backend: /^[^\\s@]+@[^\\s@]+\\.([^\\s@]+)$/
6:  */
7: export const validateEmail = (email) => {
8:   const re = /^[^\\s@]+@[^\\s@]+\\.([^\\s@]+)$/;
9:   return re.test(email);
10: };
11:
12: /**
13:  * Validate password strength
14:  * MATCHES backend: 8+ chars, uppercase, lowercase, number, special char
15:  */
16: export const validatePassword = (password) => {
17:   return (
18:     password.length >= 8 &&
19:     /[a-z]/.test(password) &&
20:     /[A-Z]/.test(password) &&

```

```

21:     /\d/.test(password) &&
22:     /[!%*?&]/.test(password)
23:   );
24: };
25:
26: /**
27:  * Validate username format
28:  * MATCHES backend: 3-30 alphanumeric + underscore
29:  */
30: export const validateUsername = (username) => {
31:   return /^[a-zA-Z0-9_]{3,30}$/.test(username);
32: };
33:
34: /**
35:  * Get password strength indicator
36:  * Returns visual feedback for password strength
37:  */
38: export const getPasswordStrength = (password) => {
39:   let strength = 0;
40:
41:   if (password.length >= 8) strength++;
42:   if (password.length >= 12) strength++;
43:   if (/[a-z]/.test(password)) strength++;
44:   if (/[A-Z]/.test(password)) strength++;
45:   if (/\\d/.test(password)) strength++;
46:   if (/[!%*?&]/.test(password)) strength++;
47:
48:   if (strength <= 2) {
49:     return {
50:       label: "Weak",
51:       color: "bg-red-500",
52:       width: "33%",
53:       textColor: "text-red-700",
54:     };
55:   }
56:
57:   if (strength <= 4) {
58:     return {
59:       label: "Medium",
60:       color: "bg-yellow-500",
61:       width: "66%",
62:       textColor: "text-yellow-700",
63:     };
64:   }
65:
66:   return {
67:     label: "Strong",
68:     color: "bg-green-500",
69:     width: "100%",
70:     textColor: "text-green-700",
71:   };
72: };
73:
74: /**
75:  * Validate file type
76:  */
77: export const validateFileType = (file) => {
78:   const allowedTypes = ["image/jpeg", "image/jpg", "image/png"];
79:   return allowedTypes.includes(file.type);
80: };
81:
82: /**
83:  * Validate file size (max 5MB)
84:  */
85: export const validateFileSize = (file) => {
86:   const maxSize = 5 * 1024 * 1024; // 5MB
87:   return file.size <= maxSize;
88: };
89:
90: /**
91:  * Get detailed password validation errors
92:  */
93: export const getPasswordErrors = (password) => {

```

```

94:   const errors = [];
95:
96:   if (password.length < 8) {
97:     errors.push("At least 8 characters required");
98:   }
99:   if (!/[a-z]/.test(password)) {
100:     errors.push("At least one lowercase letter required");
101:   }
102:   if (!/[A-Z]/.test(password)) {
103:     errors.push("At least one uppercase letter required");
104:   }
105:   if (!/\d/.test(password)) {
106:     errors.push("At least one number required");
107:   }
108:   if (!/[@$!%*?&]/.test(password)) {
109:     errors.push("At least one special character (@$!%*?&) required");
110:   }
111:
112:   return errors;
113: };

```

File: tailwind.config.js

```

1: /** @type {import('tailwindcss').Config} */
2: export default {
3:   content: [
4:     "./index.html",
5:     "./src/**/*.{js,ts,jsx,tsx}",
6:   ],
7:   theme: {
8:     extend: {
9:       colors: {
10:         primary: {
11:           50: '#f0fdf4',
12:           100: '#dcfce7',
13:           200: '#bbf7d0',
14:           300: '#86efac',
15:           400: '#4ade80',
16:           500: '#22c55e',
17:           600: '#16a34a',
18:           700: '#15803d',
19:           800: '#166534',
20:           900: '#14532d',
21:           950: '#052e16',
22:         },
23:         forest: {
24:           50: '#f4fef7',
25:           100: '#e8fcef',
26:           200: '#c5f7d7',
27:           300: '#a1f2be',
28:           400: '#5be88d',
29:           500: '#14dd5c',
30:           600: '#12c753',
31:           700: '#0fa645',
32:           800: '#0c8537',
33:           900: '#0a6d2d',
34:         },
35:         sage: {
36:           50: '#f7f9f5',
37:           100: '#eff3eb',
38:           200: '#d7e0cd',
39:           300: '#bfceaf',
40:           400: '#8fa973',
41:           500: '#5f8437',
42:           600: '#567632',
43:           700: '#48622a',
44:           800: '#3a4f22',
45:           900: '#2f401b',
46:         },
47:       },
48:       fontFamily: {

```



```

49:     sans: ['Inter', 'system-ui', 'sans-serif'],
50:     display: ['Poppins', 'Inter', 'sans-serif'],
51:   },
52:   animation: {
53:     'fade-in': 'fadeIn 0.5s ease-in-out',
54:     'slide-up': 'slideUp 0.5s ease-out',
55:     'slide-down': 'slideDown 0.5s ease-out',
56:     'scale-in': 'scaleIn 0.3s ease-out',
57:   },
58:   keyframes: {
59:     fadeIn: {
60:       '0%': { opacity: '0' },
61:       '100%': { opacity: '1' },
62:     },
63:     slideUp: {
64:       '0%': { transform: 'translateY(20px)', opacity: '0' },
65:       '100%': { transform: 'translateY(0)', opacity: '1' },
66:     },
67:     slideDown: {
68:       '0%': { transform: 'translateY(-20px)', opacity: '0' },
69:       '100%': { transform: 'translateY(0)', opacity: '1' },
70:     },
71:     scaleIn: {
72:       '0%': { transform: 'scale(0.9)', opacity: '0' },
73:       '100%': { transform: 'scale(1)', opacity: '1' },
74:     },
75:   },
76: },
77: },
78: plugins: [],
79: }

```

■ File: vite.config.js

```

=====
1: import { defineConfig } from 'vite'
2: import react from '@vitejs/plugin-react'
3: import path from 'path'
4:
5: export default defineConfig({
6:   plugins: [react()],
7:   resolve: {
8:     alias: {
9:       '@': path.resolve(__dirname, './src'),
10:    },
11:  },
12:  server: {
13:    port: 5173,
14:    proxy: {
15:      '/api': {
16:        target: 'http://localhost:5000',
17:        changeOrigin: true,
18:      },
19:    },
20:  },
21: })
=====

```