Table of Contents

[1. TECH STACK DECISION: 1](#_Toc211803659)

[2. Step 1: Create new project with stable setup 1](#_Toc211803660)

[3. Step 2: Install stable Tailwind v3 1](#_Toc211803661)

[4. D:\Saif-Solutions\projects\authbase-stable\tailwind.config.js 1](#_Toc211803662)

[5. D:\Saif-Solutions\projects\authbase-stable\src\index.css 2](#_Toc211803663)

[6. D:\Saif-Solutions\projects\authbase-stable\src\App.tsx 7](#_Toc211803664)

[7. Step 1: Install essential dependencies 8](#_Toc211803665)

[8. Step 2: Install shadcn/ui components 8](#_Toc211803666)

[9. Step 3: Add core components 8](#_Toc211803667)

[10. D:\Saif-Solutions\projects\authbase-stable\tsconfig.json 8](#_Toc211803668)

## TECH STACK DECISION:

text

Frontend: React + TypeScript + Tailwind CSS

Components: shadcn/ui (professional, customizable)

State: React Context + useReducer

API: Axios with interceptors

## Step 1: Create new project with stable setup

bash

cd ..

npm create vite@latest authbase-stable -- --template react-ts

cd authbase-stable

npm install

## Step 2: Install stable Tailwind v3

bash

npm install -D tailwindcss@^3.4.0 postcss autoprefixer

npx tailwindcss init -p

## D:\Saif-Solutions\projects\authbase-stable\tailwind.config.js

/\*\* @type {import('tailwindcss').Config} \*/

export default {

  darkMode: ["class"],

  content: ["./index.html", "./src/\*\*/\*.{js,ts,jsx,tsx}"],

  theme: {

    extend: {

      borderRadius: {

        lg: "var(--radius)",

        md: "calc(var(--radius) - 2px)",

        sm: "calc(var(--radius) - 4px)",

      },

      colors: {

        background: "hsl(var(--background))",

        foreground: "hsl(var(--foreground))",

        card: {

          DEFAULT: "hsl(var(--card))",

          foreground: "hsl(var(--card-foreground))",

        },

        popover: {

          DEFAULT: "hsl(var(--popover))",

          foreground: "hsl(var(--popover-foreground))",

        },

        primary: {

          DEFAULT: "hsl(var(--primary))",

          foreground: "hsl(var(--primary-foreground))",

        },

        secondary: {

          DEFAULT: "hsl(var(--secondary))",

          foreground: "hsl(var(--secondary-foreground))",

        },

        muted: {

          DEFAULT: "hsl(var(--muted))",

          foreground: "hsl(var(--muted-foreground))",

        },

        accent: {

          DEFAULT: "hsl(var(--accent))",

          foreground: "hsl(var(--accent-foreground))",

        },

        destructive: {

          DEFAULT: "hsl(var(--destructive))",

          foreground: "hsl(var(--destructive-foreground))",

        },

        border: "hsl(var(--border))",

        input: "hsl(var(--input))",

        ring: "hsl(var(--ring))",

        chart: {

          1: "hsl(var(--chart-1))",

          2: "hsl(var(--chart-2))",

          3: "hsl(var(--chart-3))",

          4: "hsl(var(--chart-4))",

          5: "hsl(var(--chart-5))",

        },

      },

      // Add explicit color extensions for better dark mode support

      backgroundColor: {

        "dark-card": "#1f2937",

        "dark-sidebar": "#111827",

        "dark-input": "#374151",

      },

      textColor: {

        "dark-primary": "#f9fafb",

        "dark-secondary": "#d1d5db",

        "dark-muted": "#9ca3af",

      },

      borderColor: {

        "dark-border": "#374151",

      },

    },

  },

  plugins: [require("tailwindcss-animate")],

};

## D:\Saif-Solutions\projects\authbase-stable\src\index.css

@tailwind base;

@tailwind components;

@tailwind utilities;

/\* CSS Variables for light theme \*/

:root {

--background: 0 0% 100%;

--foreground: 222.2 84% 4.9%;

--card: 0 0% 100%;

--card-foreground: 222.2 84% 4.9%;

--popover: 0 0% 100%;

--popover-foreground: 222.2 84% 4.9%;

--primary: 222.2 47.4% 11.2%;

--primary-foreground: 210 40% 98%;

--secondary: 210 40% 96%;

--secondary-foreground: 222.2 47.4% 11.2%;

--muted: 210 40% 96%;

--muted-foreground: 215.4 16.3% 46.9%;

--accent: 210 40% 96%;

--accent-foreground: 222.2 47.4% 11.2%;

--destructive: 0 84.2% 60.2%;

--destructive-foreground: 210 40% 98%;

--border: 214.3 31.8% 91.4%;

--input: 214.3 31.8% 91.4%;

--ring: 222.2 84% 4.9%;

--radius: 0.5rem;

--chart-1: 12 76% 61%;

--chart-2: 173 58% 39%;

--chart-3: 197 37% 24%;

--chart-4: 43 74% 66%;

--chart-5: 27 87% 67%;

}

/\* CSS Variables for dark theme \*/

.dark {

--background: 222.2 84% 4.9%;

--foreground: 210 40% 98%;

--card: 222.2 84% 4.9%;

--card-foreground: 210 40% 98%;

--popover: 222.2 84% 4.9%;

--popover-foreground: 210 40% 98%;

--primary: 210 40% 98%;

--primary-foreground: 222.2 47.4% 11.2%;

--secondary: 217.2 32.6% 17.5%;

--secondary-foreground: 210 40% 98%;

--muted: 217.2 32.6% 17.5%;

--muted-foreground: 215 20.2% 65.1%;

--accent: 217.2 32.6% 17.5%;

--accent-foreground: 210 40% 98%;

--destructive: 0 62.8% 30.6%;

--destructive-foreground: 210 40% 98%;

--border: 217.2 32.6% 17.5%;

--input: 217.2 32.6% 17.5%;

--ring: 212.7 26.8% 83.9%;

--chart-1: 220 70% 50%;

--chart-2: 160 60% 45%;

--chart-3: 30 80% 55%;

--chart-4: 280 65% 60%;

--chart-5: 340 75% 55%;

}

/\* React Router Link styles \*/

a.active {

background-color: rgb(37 99 235);

color: white;

}

a.active:hover {

background-color: rgb(29 78 216);

}

/\* Custom scrollbar for better UX \*/

::-webkit-scrollbar {

width: 6px;

}

::-webkit-scrollbar-track {

background: #f1f1f1;

}

::-webkit-scrollbar-thumb {

background: #c1c1c1;

border-radius: 3px;

}

::-webkit-scrollbar-thumb:hover {

background: #a8a8a8;

}

/\* Dark mode scrollbar \*/

.dark ::-webkit-scrollbar-track {

background: #374151;

}

.dark ::-webkit-scrollbar-thumb {

background: #6b7280;

}

.dark ::-webkit-scrollbar-thumb:hover {

background: #9ca3af;

}

/\* Dark mode base styles \*/

.dark {

color-scheme: dark;

}

/\* Smooth transitions for theme switching \*/

\* {

transition: background-color 0.2s ease, border-color 0.2s ease,

color 0.2s ease, box-shadow 0.2s ease;

}

/\* Ensure smooth theme transitions for specific elements \*/

html {

transition: background-color 0.2s ease, color 0.2s ease;

}

body {

transition: background-color 0.2s ease, color 0.2s ease;

}

/\* Custom dark mode utility classes \*/

@layer utilities {

.dark .dark\:bg-gray-900 {

background-color: #111827;

}

.dark .dark\:bg-gray-800 {

background-color: #1f2937;

}

.dark .dark\:bg-gray-700 {

background-color: #374151;

}

.dark .dark\:text-white {

color: #f9fafb;

}

.dark .dark\:text-gray-100 {

color: #f3f4f6;

}

.dark .dark\:text-gray-200 {

color: #e5e7eb;

}

.dark .dark\:text-gray-300 {

color: #d1d5db;

}

.dark .dark\:text-gray-400 {

color: #9ca3af;

}

.dark .dark\:border-gray-700 {

border-color: #374151;

}

.dark .dark\:border-gray-600 {

border-color: #4b5563;

}

.dark .dark\:shadow-lg {

box-shadow: 0 10px 15px -3px rgba(0, 0, 0, 0.3),

0 4px 6px -2px rgba(0, 0, 0, 0.2);

}

}

/\* Custom component styles \*/

@layer components {

/\* Enhanced card shadows \*/

.card-shadow {

box-shadow: 0 1px 3px 0 rgba(0, 0, 0, 0.1), 0 1px 2px 0 rgba(0, 0, 0, 0.06);

}

.dark .card-shadow {

box-shadow: 0 1px 3px 0 rgba(0, 0, 0, 0.3), 0 1px 2px 0 rgba(0, 0, 0, 0.2);

}

/\* Glass morphism effect \*/

.glass {

background: rgba(255, 255, 255, 0.1);

backdrop-filter: blur(10px);

border: 1px solid rgba(255, 255, 255, 0.2);

}

.dark .glass {

background: rgba(0, 0, 0, 0.2);

border: 1px solid rgba(255, 255, 255, 0.1);

}

/\* Custom search input that works with both CSS variables and explicit colors \*/

.search-input {

border: 1px solid hsl(var(--border));

background: hsl(var(--background));

color: hsl(var(--foreground));

border-radius: 0.375rem;

padding: 0.5rem 0.75rem 0.5rem 2.5rem;

width: 100%;

height: 2.5rem;

font-size: 0.875rem;

}

.search-input:focus {

outline: none;

ring: 2px solid hsl(var(--ring));

border-color: hsl(var(--ring));

box-shadow: 0 0 0 2px hsl(var(--ring) / 0.2);

}

.search-input::placeholder {

color: hsl(var(--muted-foreground));

}

/\* Fallback for search inputs using explicit colors \*/

.search-input-fallback {

border: 1px solid #d1d5db;

background: white;

color: #111827;

border-radius: 0.375rem;

padding: 0.5rem 0.75rem 0.5rem 2.5rem;

width: 100%;

height: 2.5rem;

font-size: 0.875rem;

}

.dark .search-input-fallback {

border: 1px solid #4b5563;

background: #374151;

color: white;

}

.search-input-fallback:focus {

outline: none;

border-color: #3b82f6;

box-shadow: 0 0 0 2px rgba(59, 130, 246, 0.2);

}

.search-input-fallback::placeholder {

color: #6b7280;

}

.dark .search-input-fallback::placeholder {

color: #9ca3af;

}

}

/\* Force dark mode styles for inputs when needed \*/

.dark input.search-input-force {

border-color: #4b5563 !important;

background-color: #374151 !important;

color: white !important;

}

.dark input.search-input-force::placeholder {

color: #9ca3af !important;

}

## D:\Saif-Solutions\projects\authbase-stable\src\App.tsx

import { BrowserRouter as Router, Routes, Route } from "react-router-dom";

import { Layout } from "./components/layout/Layout";

import { Dashboard } from "./pages/Dashboard";

import { Login } from "./pages/Login";

import { Users } from "./pages/Users";

import { Sessions } from "./pages/Sessions";

import { Analytics } from "./pages/Analytics";

import { Settings } from "./pages/Settings";

import { Toaster } from "@/components/ui/Toaster";

function App() {

return (

<>

<Router>

<Routes>

<Route path="/login" element={<Login />} />

<Route

path="/\*"

element={

<Layout>

<Routes>

<Route path="/dashboard" element={<Dashboard />} />

<Route path="/users" element={<Users />} />

<Route path="/sessions" element={<Sessions />} />

<Route path="/analytics" element={<Analytics />} />

<Route path="/settings" element={<Settings />} />

<Route path="/" element={<Dashboard />} />

</Routes>

</Layout>

}

/>

</Routes>

</Router>

<Toaster />

</>

);

}

export default App;

## Step 1: Install essential dependencies

bash

npm install react-router-dom lucide-react axios

npm install clsx tailwind-merge

## Step 2: Install shadcn/ui components

bash

npx shadcn@latest init

Choose: Default, Slate, Yes, @/\*, No

## Step 3: Add core components

bash

npx shadcn@latest add button card table input badge dropdown-menu

## D:\Saif-Solutions\projects\authbase-stable\tsconfig.json

{

  "compilerOptions": {

    "target": "ES2020",

    "useDefineForClassFields": true,

    "lib": ["ES2020", "DOM", "DOM.Iterable"],

    "module": "ESNext",

    "skipLibCheck": true,

    "moduleResolution": "bundler",

    "allowImportingTsExtensions": true,

    "resolveJsonModule": true,

    "isolatedModules": true,

    "noEmit": true,

    "jsx": "react-jsx",

    "strict": true,

    "noUnusedLocals": true,

    "noUnusedParameters": true,

    "noFallthroughCasesInSwitch": true,

    "baseUrl": ".",

    "paths": {

      "@/\*": ["./src/\*"]

    }

  },

  "include": ["src"]

}

## D:\Saif-Solutions\projects\authbase-stable\tsconfig.node.json

{

"compilerOptions": {

"composite": true,

"skipLibCheck": true,

"module": "ESNext",

"moduleResolution": "bundler",

"allowSyntheticDefaultImports": true

},

"include": ["vite.config.ts"]

}

## NEXT STEP: ADD CORE COMPONENTS

bash

npx shadcn@latest add button card table input badge dropdown-menu avatar

**This installs the professional components you need for:**

* Buttons throughout the app
* Cards for dashboard metrics
* Tables for user management
* Inputs for forms and search
* Badges for status indicators
* Dropdown menus for actions
* Avatars for user profiles

## D:\Saif-Solutions\projects\authbase-stable\src\components\layout\Sidebar.tsx

import { cn } from "@/lib/utils";

import {

LayoutDashboard,

Users,

Settings,

Shield,

BarChart3,

} from "lucide-react";

import { NavLink, useLocation } from "react-router-dom";

const navigation = [

{ name: "Dashboard", href: "/dashboard", icon: LayoutDashboard },

{ name: "Users", href: "/users", icon: Users },

{ name: "Sessions", href: "/sessions", icon: Shield },

{ name: "Analytics", href: "/analytics", icon: BarChart3 },

{ name: "Settings", href: "/settings", icon: Settings },

];

export function Sidebar() {

const location = useLocation();

return (

<div className="flex h-full w-64 flex-col bg-gray-900 border-r border-gray-800">

{/\* Logo \*/}

<div className="flex h-16 items-center justify-center border-b border-gray-800 px-4">

<div className="flex items-center space-x-2">

<Shield className="h-8 w-8 text-blue-500" />

<span className="text-xl font-bold text-white">AuthBase Pro</span>

</div>

</div>

{/\* Navigation \*/}

<nav className="flex-1 space-y-1 px-3 py-6">

{navigation.map((item) => {

const isActive =

location.pathname === item.href ||

(item.href === "/dashboard" && location.pathname === "/");

return (

<NavLink

key={item.name}

to={item.href}

className={cn(

"group flex items-center rounded-lg px-3 py-2.5 text-sm font-medium transition-all duration-200",

isActive

? "bg-blue-600 text-white shadow-sm"

: "text-gray-300 hover:bg-gray-800 hover:text-white"

)}

>

<item.icon

className={cn(

"mr-3 h-5 w-5 transition-colors",

isActive

? "text-white"

: "text-gray-400 group-hover:text-white"

)}

/>

{item.name}

</NavLink>

);

})}

</nav>

{/\* User section \*/}

<div className="border-t border-gray-800 p-4">

<div className="flex items-center space-x-3">

<div className="h-8 w-8 rounded-full bg-gradient-to-br from-blue-500 to-blue-600 flex items-center justify-center shadow-sm">

<span className="text-sm font-medium text-white">A</span>

</div>

<div className="flex-1 min-w-0">

<p className="text-sm font-medium text-white truncate">

Admin User

</p>

<p className="text-xs text-gray-400 truncate">admin@authbase.com</p>

</div>

</div>

</div>

</div>

);

}

## D:\Saif-Solutions\projects\authbase-stable\src\components\layout\Header.tsx

import { Bell, Search, ChevronDown } from "lucide-react";

import { Button } from "@/components/ui/button";

import { Input } from "@/components/ui/input";

import { ThemeToggle } from "./ThemeToggle";

import {

DropdownMenu,

DropdownMenuContent,

DropdownMenuItem,

DropdownMenuTrigger,

} from "@/components/ui/dropdown-menu";

export function Header() {

return (

<header className="flex h-16 items-center justify-between border-b border-gray-200 bg-white px-6 shadow-sm">

{/\* Page title \*/}

<div>

<h1 className="text-xl font-semibold text-gray-900">Dashboard</h1>

<p className="text-sm text-gray-500">Welcome back, Admin</p>

</div>

{/\* Actions \*/}

<div className="flex items-center space-x-4">

{/\* Search \*/}

<div className="relative">

<Search className="absolute left-3 top-1/2 h-4 w-4 -translate-y-1/2 text-gray-500" />

<Input

type="search"

placeholder="Search..."

className="w-64 pl-10 border-gray-300"

/>

</div>

<ThemeToggle />

{/\* Notifications \*/}

<DropdownMenu>

<DropdownMenuTrigger asChild>

<Button variant="ghost" size="icon" className="relative">

<Bell className="h-5 w-5" />

<span className="absolute -top-1 -right-1 flex h-5 w-5 items-center justify-center rounded-full bg-red-500 text-xs font-medium text-white">

3

</span>

</Button>

</DropdownMenuTrigger>

<DropdownMenuContent align="end" className="w-80">

<div className="p-2 border-b border-gray-100">

<p className="font-semibold text-gray-900">Notifications</p>

</div>

<DropdownMenuItem className="flex flex-col items-start p-3 cursor-pointer">

<p className="font-medium text-gray-900">New user registered</p>

<p className="text-sm text-gray-500">

john@example.com just signed up

</p>

<p className="text-xs text-gray-400 mt-1">2 minutes ago</p>

</DropdownMenuItem>

</DropdownMenuContent>

</DropdownMenu>

{/\* User menu \*/}

<DropdownMenu>

<DropdownMenuTrigger asChild>

<Button

variant="ghost"

className="flex items-center space-x-2 hover:bg-gray-100 px-3 py-2"

>

<div className="h-8 w-8 rounded-full bg-gradient-to-br from-blue-500 to-blue-600 flex items-center justify-center shadow-sm">

<span className="text-sm font-semibold text-white">A</span>

</div>

<div className="hidden sm:block text-left">

<p className="text-sm font-semibold text-gray-900">

Admin User

</p>

<p className="text-xs text-gray-500">Super Admin</p>

</div>

<ChevronDown className="h-4 w-4 text-gray-500" />

</Button>

</DropdownMenuTrigger>

<DropdownMenuContent align="end" className="w-56">

<DropdownMenuItem className="cursor-pointer flex items-center gap-2">

<div className="h-6 w-6 rounded-full bg-gradient-to-br from-blue-600 to-blue-700 flex items-center justify-center">

<span className="text-xs font-semibold text-white">A</span>

</div>

<div>

<p className="font-medium">Admin User</p>

<p className="text-sm text-gray-500">admin@authbase.com</p>

</div>

</DropdownMenuItem>

<div className="border-t border-gray-100 my-1"></div>

<DropdownMenuItem className="cursor-pointer">

Profile Settings

</DropdownMenuItem>

<DropdownMenuItem className="cursor-pointer">

Billing & Plan

</DropdownMenuItem>

<DropdownMenuItem className="cursor-pointer text-red-600 focus:text-red-600">

Sign Out

</DropdownMenuItem>

</DropdownMenuContent>

</DropdownMenu>

</div>

</header>

);

}

## D:\Saif-Solutions\projects\authbase-stable\src\components\layout\Layout.tsx

import { Sidebar } from "./Sidebar";

import { Header } from "./Header";

interface LayoutProps {

children: React.ReactNode;

}

export function Layout({ children }: LayoutProps) {

return (

<div className="flex h-screen bg-gray-100 dark:bg-gray-900">

{/\* Sidebar \*/}

<Sidebar />

{/\* Main content \*/}

<div className="flex flex-1 flex-col">

<Header />

<main className="flex-1 overflow-auto p-6 bg-gray-50 dark:bg-gray-900">

{children}

</main>

</div>

</div>

);

}

## D:\Saif-Solutions\projects\authbase-stable\src\components\layout\ThemeToggle.tsx

import { Moon, Sun } from "lucide-react";

import { Button } from "@/components/ui/button";

import { useState, useEffect } from "react";

export function ThemeToggle() {

const [theme, setTheme] = useState<"light" | "dark">("light");

useEffect(() => {

const savedTheme = localStorage.getItem("theme") as "light" | "dark" | null;

if (savedTheme) {

setTheme(savedTheme);

document.documentElement.classList.toggle("dark", savedTheme === "dark");

}

}, []);

const toggleTheme = () => {

const newTheme = theme === "light" ? "dark" : "light";

setTheme(newTheme);

localStorage.setItem("theme", newTheme);

document.documentElement.classList.toggle("dark", newTheme === "dark");

};

return (

<Button

variant="ghost"

size="icon"

onClick={toggleTheme}

className="relative h-9 w-9 rounded-lg border border-gray-200 bg-white/50 backdrop-blur-sm hover:bg-gray-100 dark:border-gray-700 dark:bg-gray-800/50 dark:hover:bg-gray-700"

>

<Sun className="h-4 w-4 rotate-0 scale-100 transition-all dark:-rotate-90 dark:scale-0" />

<Moon className="absolute h-4 w-4 rotate-90 scale-0 transition-all dark:rotate-0 dark:scale-100" />

<span className="sr-only">Toggle theme</span>

</Button>

);

}

## D:\Saif-Solutions\projects\authbase-stable\src\pages\Analytics.tsx

import {

TrendingUp,

Users,

Shield,

Activity,

Eye,

Clock,

BarChart3,

Download,

} from "lucide-react";

import {

Card,

CardContent,

CardHeader,

CardTitle,

CardDescription,

} from "@/components/ui/card";

import { Button } from "@/components/ui/button";

import { Badge } from "@/components/ui/badge";

export function Analytics() {

const metrics = [

{

title: "Total Users",

value: "1,248",

change: "+12%",

description: "From last month",

icon: Users,

color: "text-blue-600 dark:text-blue-400",

bgColor: "bg-blue-50 dark:bg-blue-900/50",

},

{

title: "Active Sessions",

value: "342",

change: "+8%",

description: "From last week",

icon: Shield,

color: "text-green-600 dark:text-green-400",

bgColor: "bg-green-50 dark:bg-green-900/50",

},

{

title: "API Requests",

value: "24.8K",

change: "+23%",

description: "From yesterday",

icon: Activity,

color: "text-purple-600 dark:text-purple-400",

bgColor: "bg-purple-50 dark:bg-purple-900/50",

},

{

title: "Success Rate",

value: "99.8%",

change: "+0.2%",

description: "System stability",

icon: TrendingUp,

color: "text-orange-600 dark:text-orange-400",

bgColor: "bg-orange-50 dark:bg-orange-900/50",

},

];

const chartData = [

{ day: "Mon", users: 120, sessions: 85, requests: 4200 },

{ day: "Tue", users: 145, sessions: 92, requests: 4800 },

{ day: "Wed", users: 132, sessions: 78, requests: 4500 },

{ day: "Thu", users: 168, sessions: 105, requests: 5200 },

{ day: "Fri", users: 156, sessions: 98, requests: 4900 },

{ day: "Sat", users: 142, sessions: 88, requests: 4600 },

{ day: "Sun", users: 128, sessions: 82, requests: 4400 },

];

const topEvents = [

{ event: "User Login", count: 1248, trend: "up" },

{ event: "Password Reset", count: 342, trend: "down" },

{ event: "Session Created", count: 892, trend: "up" },

{ event: "API Call", count: 24800, trend: "up" },

{ event: "Failed Login", count: 45, trend: "down" },

];

const maxRequests = Math.max(...chartData.map((d) => d.requests));

return (

<div className="space-y-6">

{/\* Page Header \*/}

<div className="flex flex-col sm:flex-row justify-between items-start sm:items-center gap-4">

<div className="space-y-1">

<h1 className="text-2xl font-bold text-gray-900 dark:text-white">

Analytics Dashboard

</h1>

<p className="text-gray-600 dark:text-gray-400">

Comprehensive insights and performance metrics

</p>

</div>

<div className="flex items-center gap-3">

<Button

variant="outline"

className="flex items-center gap-2 dark:border-gray-600 dark:text-gray-300 dark:hover:bg-gray-700"

>

<Download className="h-4 w-4" />

Export Report

</Button>

<Badge

variant="secondary"

className="bg-green-100 text-green-800 border-green-200 dark:bg-green-900 dark:text-green-200 dark:border-green-700"

>

<Eye className="h-3 w-3 mr-1" />

Live Data

</Badge>

</div>

</div>

{/\* Key Metrics \*/}

<div className="grid gap-4 md:grid-cols-2 lg:grid-cols-4">

{metrics.map((metric) => (

<Card

key={metric.title}

className="border-gray-200 shadow-sm dark:border-gray-700 dark:bg-gray-800"

>

<CardHeader className="flex flex-row items-center justify-between space-y-0 pb-3">

<CardTitle className="text-sm font-medium text-gray-600 dark:text-gray-400">

{metric.title}

</CardTitle>

<div className={`p-2 rounded-lg ${metric.bgColor}`}>

<metric.icon className={`h-4 w-4 ${metric.color}`} />

</div>

</CardHeader>

<CardContent>

<div className="text-2xl font-bold text-gray-900 dark:text-white">

{metric.value}

</div>

<p className="text-xs text-gray-500 dark:text-gray-400 mt-1">

<span className="text-green-600 dark:text-green-400 font-medium">

{metric.change}

</span>{" "}

{metric.description}

</p>

</CardContent>

</Card>

))}

</div>

<div className="grid gap-6 lg:grid-cols-2">

{/\* Requests Chart \*/}

<Card className="border-gray-200 shadow-sm dark:border-gray-700 dark:bg-gray-800">

<CardHeader>

<CardTitle className="flex items-center gap-2 dark:text-white">

<BarChart3 className="h-5 w-5 text-gray-600 dark:text-gray-400" />

API Requests (Last 7 Days)

</CardTitle>

<CardDescription className="dark:text-gray-400">

Daily API request volume and trends

</CardDescription>

</CardHeader>

<CardContent>

<div className="space-y-4">

<div className="flex items-end justify-between h-40">

{chartData.map((day) => (

<div

key={day.day}

className="flex flex-col items-center space-y-2 flex-1"

>

<div

className="w-8 bg-gradient-to-t from-blue-500 to-blue-600 rounded-t-lg transition-all hover:from-blue-600 hover:to-blue-700"

style={{

height: `${(day.requests / maxRequests) \* 100}%`,

}}

/>

<div className="text-xs text-gray-500 dark:text-gray-400">

{day.day}

</div>

<div className="text-xs font-medium text-gray-900 dark:text-white">

{day.requests.toLocaleString()}

</div>

</div>

))}

</div>

</div>

</CardContent>

</Card>

{/\* Top Events \*/}

<Card className="border-gray-200 shadow-sm dark:border-gray-700 dark:bg-gray-800">

<CardHeader>

<CardTitle className="flex items-center gap-2 dark:text-white">

<Activity className="h-5 w-5 text-gray-600 dark:text-gray-400" />

Top Events

</CardTitle>

<CardDescription className="dark:text-gray-400">

Most frequent authentication events

</CardDescription>

</CardHeader>

<CardContent>

<div className="space-y-4">

{topEvents.map((event) => (

<div

key={event.event}

className="flex items-center justify-between p-3 rounded-lg border border-gray-100 dark:border-gray-700"

>

<div className="flex items-center gap-3">

<div

className={`p-2 rounded-lg ${

event.trend === "up"

? "bg-green-50 text-green-600 dark:bg-green-900/50 dark:text-green-400"

: "bg-red-50 text-red-600 dark:bg-red-900/50 dark:text-red-400"

}`}

>

<TrendingUp className="h-4 w-4" />

</div>

<div>

<div className="font-medium text-gray-900 dark:text-white">

{event.event}

</div>

<div className="text-sm text-gray-500 dark:text-gray-400">

{event.count.toLocaleString()} events

</div>

</div>

</div>

<Badge

variant="secondary"

className={

event.trend === "up"

? "bg-green-100 text-green-800 border-green-200 dark:bg-green-900 dark:text-green-200 dark:border-green-700"

: "bg-red-100 text-red-800 border-red-200 dark:bg-red-900 dark:text-red-200 dark:border-red-700"

}

>

{event.trend === "up" ? "↑" : "↓"}

</Badge>

</div>

))}

</div>

</CardContent>

</Card>

</div>

{/\* Detailed Stats \*/}

<div className="grid gap-6 md:grid-cols-3">

<Card className="border-gray-200 shadow-sm dark:border-gray-700 dark:bg-gray-800">

<CardHeader>

<CardTitle className="flex items-center gap-2 text-sm dark:text-white">

<Users className="h-4 w-4 text-blue-600 dark:text-blue-400" />

User Growth

</CardTitle>

</CardHeader>

<CardContent>

<div className="space-y-2">

<div className="flex justify-between">

<span className="text-sm text-gray-600 dark:text-gray-400">

New Users

</span>

<span className="font-medium text-green-600 dark:text-green-400">

+124

</span>

</div>

<div className="flex justify-between">

<span className="text-sm text-gray-600 dark:text-gray-400">

Active Rate

</span>

<span className="font-medium text-gray-900 dark:text-white">

74%

</span>

</div>

<div className="flex justify-between">

<span className="text-sm text-gray-600 dark:text-gray-400">

Retention

</span>

<span className="font-medium text-gray-900 dark:text-white">

89%

</span>

</div>

</div>

</CardContent>

</Card>

<Card className="border-gray-200 shadow-sm dark:border-gray-700 dark:bg-gray-800">

<CardHeader>

<CardTitle className="flex items-center gap-2 text-sm dark:text-white">

<Shield className="h-4 w-4 text-green-600 dark:text-green-400" />

Security Metrics

</CardTitle>

</CardHeader>

<CardContent>

<div className="space-y-2">

<div className="flex justify-between">

<span className="text-sm text-gray-600 dark:text-gray-400">

Failed Logins

</span>

<span className="font-medium text-red-600 dark:text-red-400">

45

</span>

</div>

<div className="flex justify-between">

<span className="text-sm text-gray-600 dark:text-gray-400">

Suspicious Activity

</span>

<span className="font-medium text-orange-600 dark:text-orange-400">

12

</span>

</div>

<div className="flex justify-between">

<span className="text-sm text-gray-600 dark:text-gray-400">

Security Score

</span>

<span className="font-medium text-green-600 dark:text-green-400">

98%

</span>

</div>

</div>

</CardContent>

</Card>

<Card className="border-gray-200 shadow-sm dark:border-gray-700 dark:bg-gray-800">

<CardHeader>

<CardTitle className="flex items-center gap-2 text-sm dark:text-white">

<Clock className="h-4 w-4 text-purple-600 dark:text-purple-400" />

Performance

</CardTitle>

</CardHeader>

<CardContent>

<div className="space-y-2">

<div className="flex justify-between">

<span className="text-sm text-gray-600 dark:text-gray-400">

Avg Response Time

</span>

<span className="font-medium text-gray-900 dark:text-white">

42ms

</span>

</div>

<div className="flex justify-between">

<span className="text-sm text-gray-600 dark:text-gray-400">

Uptime

</span>

<span className="font-medium text-green-600 dark:text-green-400">

99.9%

</span>

</div>

<div className="flex justify-between">

<span className="text-sm text-gray-600 dark:text-gray-400">

Peak Load

</span>

<span className="font-medium text-gray-900 dark:text-white">

2.4K RPM

</span>

</div>

</div>

</CardContent>

</Card>

</div>

</div>

);

}

## D:\Saif-Solutions\projects\authbase-stable\src\pages\Dashboard.tsx

import {

Users,

Shield,

Activity,

TrendingUp,

CheckCircle,

AlertTriangle,

} from "lucide-react";

import { Card, CardContent, CardHeader, CardTitle } from "@/components/ui/card";

import { Badge } from "@/components/ui/badge";

export function Dashboard() {

const metrics = [

{

title: "Total Users",

value: "1,248",

change: "+12%",

description: "From last month",

icon: Users,

color: "text-blue-600 dark:text-blue-400",

bgColor: "bg-blue-50 dark:bg-blue-900/50",

},

{

title: "Active Sessions",

value: "342",

change: "+8%",

description: "From last week",

icon: Shield,

color: "text-green-600 dark:text-green-400",

bgColor: "bg-green-50 dark:bg-green-900/50",

},

{

title: "API Requests",

value: "24.8K",

change: "+23%",

description: "From yesterday",

icon: Activity,

color: "text-purple-600 dark:text-purple-400",

bgColor: "bg-purple-50 dark:bg-purple-900/50",

},

{

title: "Success Rate",

value: "99.8%",

change: "+0.2%",

description: "System stability",

icon: TrendingUp,

color: "text-orange-600 dark:text-orange-400",

bgColor: "bg-orange-50 dark:bg-orange-900/50",

},

];

const recentActivity = [

{

action: "User registered",

user: "john@example.com",

time: "2 minutes ago",

status: "success",

},

{

action: "Password reset",

user: "sarah@company.com",

time: "15 minutes ago",

status: "success",

},

{

action: "Failed login attempt",

user: "unknown@ip.com",

time: "1 hour ago",

status: "warning",

},

{

action: "API key generated",

user: "admin@authbase.com",

time: "2 hours ago",

status: "success",

},

];

return (

<div className="space-y-6">

{/\* Welcome Header \*/}

<div className="flex flex-col sm:flex-row justify-between items-start sm:items-center gap-4">

<div>

<h1 className="text-2xl font-bold text-gray-900 dark:text-white">

Dashboard Overview

</h1>

<p className="text-gray-600 dark:text-gray-400">

Welcome to your AuthBase Pro administration panel

</p>

</div>

<Badge

variant="secondary"

className="bg-green-100 text-green-800 border-green-200 dark:bg-green-900 dark:text-green-200 dark:border-green-700"

>

Live Data

</Badge>

</div>

{/\* Metrics Grid \*/}

<div className="grid gap-4 md:grid-cols-2 lg:grid-cols-4">

{metrics.map((metric) => (

<Card

key={metric.title}

className="border-gray-200 shadow-sm dark:border-gray-700 dark:bg-gray-800"

>

<CardHeader className="flex flex-row items-center justify-between space-y-0 pb-3">

<CardTitle className="text-sm font-medium text-gray-600 dark:text-gray-400">

{metric.title}

</CardTitle>

<div className={`p-2 rounded-lg ${metric.bgColor}`}>

<metric.icon className={`h-4 w-4 ${metric.color}`} />

</div>

</CardHeader>

<CardContent>

<div className="text-2xl font-bold text-gray-900 dark:text-white">

{metric.value}

</div>

<p className="text-xs text-gray-500 dark:text-gray-400 mt-1">

<span className="text-green-600 dark:text-green-400 font-medium">

{metric.change}

</span>{" "}

{metric.description}

</p>

</CardContent>

</Card>

))}

</div>

{/\* Recent Activity \*/}

<Card className="border-gray-200 shadow-sm dark:border-gray-700 dark:bg-gray-800">

<CardHeader>

<CardTitle className="flex items-center gap-2 dark:text-white">

<Activity className="h-5 w-5 text-gray-600 dark:text-gray-400" />

Recent Activity

</CardTitle>

</CardHeader>

<CardContent>

<div className="space-y-4">

{recentActivity.map((activity, index) => (

<div

key={index}

className="flex items-start space-x-4 p-3 rounded-lg hover:bg-gray-50 dark:hover:bg-gray-700/50"

>

<div

className={`p-2 rounded-full ${

activity.status === "success"

? "bg-green-50 text-green-600 dark:bg-green-900/50 dark:text-green-400"

: "bg-orange-50 text-orange-600 dark:bg-orange-900/50 dark:text-orange-400"

}`}

>

{activity.status === "success" ? (

<CheckCircle className="h-4 w-4" />

) : (

<AlertTriangle className="h-4 w-4" />

)}

</div>

<div className="flex-1 space-y-1">

<p className="text-sm font-medium text-gray-900 dark:text-white">

{activity.action}

</p>

<p className="text-sm text-gray-500 dark:text-gray-400">

{activity.user}

</p>

</div>

<div className="text-right">

<p className="text-sm text-gray-500 dark:text-gray-400">

{activity.time}

</p>

<Badge

variant="secondary"

className={`text-xs ${

activity.status === "success"

? "bg-green-100 text-green-800 dark:bg-green-900 dark:text-green-200"

: "bg-orange-100 text-orange-800 dark:bg-orange-900 dark:text-orange-200"

}`}

>

{activity.status}

</Badge>

</div>

</div>

))}

</div>

</CardContent>

</Card>

</div>

);

}

## D:\Saif-Solutions\projects\authbase-stable\src\pages\Login.tsx

import { useState } from "react";

import { Shield, Mail, Lock } from "lucide-react";

import { Button } from "@/components/ui/button";

import {

Card,

CardContent,

CardDescription,

CardHeader,

CardTitle,

} from "@/components/ui/card";

export function Login() {

const [email, setEmail] = useState("");

const [password, setPassword] = useState("");

const handleSubmit = (e: React.FormEvent) => {

e.preventDefault();

// For demo, redirect to dashboard

window.location.href = "/dashboard";

};

return (

<div className="min-h-screen bg-gradient-to-br from-blue-50 to-indigo-100 dark:from-gray-900 dark:to-gray-800 flex items-center justify-center p-4">

<Card className="w-full max-w-md border-gray-200 shadow-lg dark:border-gray-700 dark:bg-gray-800">

<CardHeader className="text-center">

<div className="flex justify-center mb-4">

<Shield className="h-12 w-12 text-blue-600 dark:text-blue-400" />

</div>

<CardTitle className="text-2xl font-bold dark:text-white">

AuthBase Pro

</CardTitle>

<CardDescription className="dark:text-gray-400">

Sign in to your admin dashboard

</CardDescription>

</CardHeader>

<CardContent>

<form onSubmit={handleSubmit} className="space-y-4">

<div className="space-y-2">

<div className="relative">

<Mail className="absolute left-3 top-3 h-4 w-4 text-gray-500 dark:text-gray-400" />

<input

type="email"

placeholder="Email address"

className="w-full h-10 pl-10 pr-3 rounded-md border border-gray-300 bg-white text-gray-900 dark:border-gray-600 dark:bg-gray-700 dark:text-white focus:outline-none focus:ring-2 focus:ring-blue-500 focus:border-blue-500"

value={email}

onChange={(e) => setEmail(e.target.value)}

required

/>

</div>

</div>

<div className="space-y-2">

<div className="relative">

<Lock className="absolute left-3 top-3 h-4 w-4 text-gray-500 dark:text-gray-400" />

<input

type="password"

placeholder="Password"

className="w-full h-10 pl-10 pr-3 rounded-md border border-gray-300 bg-white text-gray-900 dark:border-gray-600 dark:bg-gray-700 dark:text-white focus:outline-none focus:ring-2 focus:ring-blue-500 focus:border-blue-500"

value={password}

onChange={(e) => setPassword(e.target.value)}

required

/>

</div>

</div>

<Button type="submit" className="w-full">

Sign In

</Button>

</form>

<div className="mt-6 text-center text-sm text-gray-600 dark:text-gray-400">

<p>Demo credentials:</p>

<p className="font-mono bg-gray-100 dark:bg-gray-700 p-2 rounded mt-1">

admin@authbase.com / password

</p>

</div>

</CardContent>

</Card>

</div>

);

}

## D:\Saif-Solutions\projects\authbase-stable\src\pages\Sessions.tsx

import { useState } from "react";

import {

MoreHorizontal,

Edit,

Trash2,

Search,

Filter,

Laptop,

Smartphone,

Tablet,

Globe,

Clock,

Calendar,

} from "lucide-react";

import { Button } from "@/components/ui/button";

// import { Input } from "@/components/ui/input";

import {

Card,

CardContent,

CardHeader,

CardTitle,

CardDescription,

} from "@/components/ui/card";

import {

Table,

TableBody,

TableCell,

TableHead,

TableHeader,

TableRow,

} from "@/components/ui/table";

import { Badge } from "@/components/ui/badge";

import {

DropdownMenu,

DropdownMenuContent,

DropdownMenuItem,

DropdownMenuTrigger,

} from "@/components/ui/dropdown-menu";

const mockSessions = [

{

id: "1",

userId: "user1",

userName: "John Smith",

userAgent: "Chrome on Windows",

deviceType: "desktop",

ipAddress: "192.168.1.100",

location: "New York, US",

lastActivity: "2024-01-15T10:30:00Z",

createdAt: "2024-01-15T08:15:00Z",

status: "active",

},

{

id: "2",

userId: "user2",

userName: "Sarah Johnson",

userAgent: "Safari on iPhone",

deviceType: "mobile",

ipAddress: "192.168.1.101",

location: "San Francisco, US",

lastActivity: "2024-01-15T09:45:00Z",

createdAt: "2024-01-14T16:30:00Z",

status: "active",

},

];

export function Sessions() {

const [searchTerm, setSearchTerm] = useState("");

const [sessions] = useState(mockSessions);

const filteredSessions = sessions.filter(

(session) =>

session.userName.toLowerCase().includes(searchTerm.toLowerCase()) ||

session.ipAddress.includes(searchTerm) ||

session.location.toLowerCase().includes(searchTerm.toLowerCase())

);

const formatDate = (dateString: string) => {

return new Date(dateString).toLocaleDateString("en-US", {

month: "short",

day: "numeric",

year: "numeric",

hour: "2-digit",

minute: "2-digit",

});

};

const getTimeAgo = (dateString: string) => {

const date = new Date(dateString);

const now = new Date();

const diffMs = now.getTime() - date.getTime();

const diffMins = Math.floor(diffMs / 60000);

const diffHours = Math.floor(diffMins / 60);

const diffDays = Math.floor(diffHours / 24);

if (diffMins < 60) return `${diffMins}m ago`;

if (diffHours < 24) return `${diffHours}h ago`;

return `${diffDays}d ago`;

};

const getDeviceIcon = (deviceType: string) => {

switch (deviceType) {

case "desktop":

return <Laptop className="h-4 w-4" />;

case "mobile":

return <Smartphone className="h-4 w-4" />;

case "tablet":

return <Tablet className="h-4 w-4" />;

default:

return <Globe className="h-4 w-4" />;

}

};

const activeSessions = sessions.filter((s) => s.status === "active").length;

const totalSessions = sessions.length;

return (

<div className="space-y-6">

{/\* Page Header \*/}

<div className="flex flex-col sm:flex-row justify-between items-start sm:items-center gap-4">

<div className="space-y-1">

<h1 className="text-2xl font-bold text-gray-900 dark:text-white">

Session Management

</h1>

<p className="text-gray-600 dark:text-gray-400">

Monitor and manage active user sessions

</p>

</div>

<div className="flex items-center gap-3">

<Button

variant="outline"

className="flex items-center gap-2 dark:border-gray-600 dark:text-gray-300"

>

<Filter className="h-4 w-4" />

Export Logs

</Button>

<Button className="flex items-center gap-2 bg-red-600 hover:bg-red-700 dark:bg-red-700 dark:hover:bg-red-800">

<Trash2 className="h-4 w-4" />

Revoke All

</Button>

</div>

</div>

{/\* Statistics Cards \*/}

<div className="grid gap-4 md:grid-cols-3">

<Card className="border-gray-200 shadow-sm dark:border-gray-700 dark:bg-gray-800">

<CardContent className="p-4 flex items-center justify-between">

<div>

<div className="text-2xl font-bold text-gray-900 dark:text-white">

{totalSessions}

</div>

<div className="text-sm text-gray-600 dark:text-gray-400">

Total Sessions

</div>

</div>

<div className="p-2 bg-blue-50 rounded-lg dark:bg-blue-900/50">

<Clock className="h-5 w-5 text-blue-600 dark:text-blue-400" />

</div>

</CardContent>

</Card>

<Card className="border-gray-200 shadow-sm dark:border-gray-700 dark:bg-gray-800">

<CardContent className="p-4 flex items-center justify-between">

<div>

<div className="text-2xl font-bold text-green-600 dark:text-green-400">

{activeSessions}

</div>

<div className="text-sm text-gray-600 dark:text-gray-400">

Active Now

</div>

</div>

<div className="p-2 bg-green-50 rounded-lg dark:bg-green-900/50">

<Laptop className="h-5 w-5 text-green-600 dark:text-green-400" />

</div>

</CardContent>

</Card>

<Card className="border-gray-200 shadow-sm dark:border-gray-700 dark:bg-gray-800">

<CardContent className="p-4 flex items-center justify-between">

<div>

<div className="text-2xl font-bold text-orange-600 dark:text-orange-400">

{totalSessions - activeSessions}

</div>

<div className="text-sm text-gray-600 dark:text-gray-400">

Expired

</div>

</div>

<div className="p-2 bg-orange-50 rounded-lg dark:bg-orange-900/50">

<Calendar className="h-5 w-5 text-orange-600 dark:text-orange-400" />

</div>

</CardContent>

</Card>

</div>

{/\* Search and Filters \*/}

<Card className="border-gray-200 shadow-sm dark:border-gray-700 dark:bg-gray-800">

<CardContent className="p-4">

<div className="flex flex-col lg:flex-row gap-4 items-start lg:items-center justify-between">

<div className="flex flex-col sm:flex-row gap-3 w-full lg:w-auto">

<div className="relative w-full sm:w-80">

<Search className="absolute left-3 top-1/2 h-4 w-4 -translate-y-1/2 text-gray-500 dark:text-gray-400 z-10" />

<input

type="text"

placeholder="Search by user, IP, or location..."

value={searchTerm}

onChange={(e) => setSearchTerm(e.target.value)}

className="search-input-fallback focus:border-blue-500 focus:ring-2 focus:ring-blue-500/20"

/>

</div>

<Button

variant="outline"

className="flex items-center gap-2 border-gray-300 dark:border-gray-600 dark:text-gray-300 dark:hover:bg-gray-700"

>

<Filter className="h-4 w-4" />

Filter

</Button>

</div>

<div className="text-sm text-gray-600 dark:text-gray-400 whitespace-nowrap">

Showing {filteredSessions.length} of {totalSessions} sessions

</div>

</div>

</CardContent>

</Card>

{/\* Sessions Table \*/}

<Card className="border-gray-200 shadow-sm overflow-hidden dark:border-gray-700 dark:bg-gray-800">

<CardHeader className="bg-gray-50/50 border-b border-gray-200 dark:bg-gray-800/50 dark:border-gray-700">

<CardTitle className="flex items-center gap-2 dark:text-white">

Active Sessions

<Badge

variant="secondary"

className="bg-blue-100 text-blue-800 border-blue-200 dark:bg-blue-900 dark:text-blue-200 dark:border-blue-700"

>

{filteredSessions.length}

</Badge>

</CardTitle>

<CardDescription className="dark:text-gray-400">

Monitor and manage user authentication sessions

</CardDescription>

</CardHeader>

<CardContent className="p-0">

<div className="overflow-x-auto">

<Table>

<TableHeader>

<TableRow className="bg-gray-50/80 hover:bg-gray-50 dark:bg-gray-800/80 dark:hover:bg-gray-800">

<TableHead className="font-semibold text-gray-900 dark:text-white py-4">

User & Device

</TableHead>

<TableHead className="font-semibold text-gray-900 dark:text-white py-4">

Location

</TableHead>

<TableHead className="font-semibold text-gray-900 dark:text-white py-4">

Last Activity

</TableHead>

<TableHead className="font-semibold text-gray-900 dark:text-white py-4">

Session Started

</TableHead>

<TableHead className="font-semibold text-gray-900 dark:text-white py-4">

Status

</TableHead>

<TableHead className="font-semibold text-gray-900 dark:text-white py-4 text-right">

Actions

</TableHead>

</TableRow>

</TableHeader>

<TableBody>

{filteredSessions.map((session) => (

<TableRow

key={session.id}

className="hover:bg-gray-50/50 border-b border-gray-100 dark:hover:bg-gray-700/50 dark:border-gray-700"

>

<TableCell className="py-4">

<div className="flex items-center gap-3">

<div className="p-2 bg-gray-100 rounded-lg dark:bg-gray-700">

{getDeviceIcon(session.deviceType)}

</div>

<div>

<div className="font-semibold text-gray-900 dark:text-white">

{session.userName}

</div>

<div className="text-sm text-gray-500 dark:text-gray-400">

{session.userAgent}

</div>

<div className="text-xs text-gray-400 dark:text-gray-500 mt-1">

{session.ipAddress}

</div>

</div>

</div>

</TableCell>

<TableCell className="py-4">

<div className="text-sm text-gray-900 dark:text-white">

{session.location}

</div>

</TableCell>

<TableCell className="py-4">

<div className="flex items-center gap-2 text-sm text-gray-600 dark:text-gray-400">

<Clock className="h-3 w-3" />

{getTimeAgo(session.lastActivity)}

</div>

<div className="text-xs text-gray-400 dark:text-gray-500">

{formatDate(session.lastActivity)}

</div>

</TableCell>

<TableCell className="py-4">

<div className="text-sm text-gray-600 dark:text-gray-400">

{formatDate(session.createdAt)}

</div>

</TableCell>

<TableCell className="py-4">

<Badge

variant={

session.status === "active" ? "default" : "secondary"

}

className={

session.status === "active"

? "bg-green-100 text-green-800 border-green-200 hover:bg-green-100 font-medium dark:bg-green-900 dark:text-green-200 dark:border-green-700"

: "bg-gray-100 text-gray-800 border-gray-200 hover:bg-gray-100 font-medium dark:bg-gray-700 dark:text-gray-200 dark:border-gray-600"

}

>

{session.status}

</Badge>

</TableCell>

<TableCell className="py-4 text-right">

<DropdownMenu>

<DropdownMenuTrigger asChild>

<Button

variant="ghost"

size="icon"

className="h-8 w-8 hover:bg-gray-100 dark:hover:bg-gray-700"

>

<MoreHorizontal className="h-4 w-4" />

</Button>

</DropdownMenuTrigger>

<DropdownMenuContent

align="end"

className="w-48 dark:bg-gray-800 dark:border-gray-700"

>

<DropdownMenuItem className="flex items-center gap-2 cursor-pointer text-sm dark:hover:bg-gray-700 dark:text-gray-300">

<Edit className="h-4 w-4" />

Edit User

</DropdownMenuItem>

<DropdownMenuItem className="flex items-center gap-2 cursor-pointer text-sm text-red-600 focus:text-red-600 dark:hover:bg-gray-700">

<Trash2 className="h-4 w-4" />

Delete User

</DropdownMenuItem>

</DropdownMenuContent>

</DropdownMenu>

</TableCell>

</TableRow>

))}

</TableBody>

</Table>

</div>

</CardContent>

</Card>

</div>

);

}

## D:\Saif-Solutions\projects\authbase-stable\src\pages\Settings.tsx

import { useState } from "react";

import { Save, Bell, Shield, Database, Globe, Key } from "lucide-react";

import { Button } from "@/components/ui/button";

import {

Card,

CardContent,

CardDescription,

CardHeader,

CardTitle,

} from "@/components/ui/card";

import { Label } from "@/components/ui/label";

import { Switch } from "@/components/ui/switch";

import { Tabs, TabsContent, TabsList, TabsTrigger } from "@/components/ui/tabs";

export function Settings() {

const [settings, setSettings] = useState({

// General Settings

siteName: "AuthBase Pro",

supportEmail: "support@authbase.com",

timezone: "UTC",

// Security Settings

twoFactorAuth: true,

sessionTimeout: 30,

passwordPolicy: "strong",

ipWhitelist: "",

// Notification Settings

emailNotifications: true,

slackNotifications: false,

securityAlerts: true,

weeklyReports: true,

// API Settings

apiRateLimit: 1000,

webhookUrl: "",

enableWebhooks: false,

});

const handleSave = () => {

// In a real app, this would save to your backend

console.log("Saving settings:", settings);

};

const handleInputChange = (key: string, value: string | number | boolean) => {

setSettings((prev) => ({

...prev,

[key]: value,

}));

};

return (

<div className="space-y-6">

{/\* Page Header \*/}

<div className="flex flex-col sm:flex-row justify-between items-start sm:items-center gap-4">

<div className="space-y-1">

<h1 className="text-2xl font-bold text-gray-900 dark:text-white">

Settings

</h1>

<p className="text-gray-600 dark:text-gray-400">

Configure your AuthBase Pro instance

</p>

</div>

<Button onClick={handleSave} className="flex items-center gap-2">

<Save className="h-4 w-4" />

Save Changes

</Button>

</div>

<Tabs defaultValue="general" className="space-y-6">

<TabsList className="grid w-full grid-cols-4 bg-gray-100 dark:bg-gray-800">

<TabsTrigger

value="general"

className="flex items-center gap-2 data-[state=active]:bg-white data-[state=active]:dark:bg-gray-700"

>

<Globe className="h-4 w-4" />

General

</TabsTrigger>

<TabsTrigger

value="security"

className="flex items-center gap-2 data-[state=active]:bg-white data-[state=active]:dark:bg-gray-700"

>

<Shield className="h-4 w-4" />

Security

</TabsTrigger>

<TabsTrigger

value="notifications"

className="flex items-center gap-2 data-[state=active]:bg-white data-[state=active]:dark:bg-gray-700"

>

<Bell className="h-4 w-4" />

Notifications

</TabsTrigger>

<TabsTrigger

value="api"

className="flex items-center gap-2 data-[state=active]:bg-white data-[state=active]:dark:bg-gray-700"

>

<Database className="h-4 w-4" />

API

</TabsTrigger>

</TabsList>

{/\* General Settings \*/}

<TabsContent value="general" className="space-y-4">

<Card className="border-gray-200 shadow-sm dark:border-gray-700 dark:bg-gray-800">

<CardHeader>

<CardTitle className="flex items-center gap-2 dark:text-white">

<Globe className="h-5 w-5 text-gray-600 dark:text-gray-400" />

General Settings

</CardTitle>

<CardDescription className="dark:text-gray-400">

Configure basic application settings and preferences

</CardDescription>

</CardHeader>

<CardContent className="space-y-4">

<div className="grid gap-4 md:grid-cols-2">

<div className="space-y-2">

<Label htmlFor="siteName" className="dark:text-gray-300">

Site Name

</Label>

<input

id="siteName"

value={settings.siteName}

onChange={(e) =>

handleInputChange("siteName", e.target.value)

}

className="w-full h-10 px-3 rounded-md border border-gray-300 bg-white text-gray-900 dark:border-gray-600 dark:bg-gray-700 dark:text-white focus:outline-none focus:ring-2 focus:ring-blue-500 focus:border-blue-500"

/>

</div>

<div className="space-y-2">

<Label htmlFor="supportEmail" className="dark:text-gray-300">

Support Email

</Label>

<input

id="supportEmail"

type="email"

value={settings.supportEmail}

onChange={(e) =>

handleInputChange("supportEmail", e.target.value)

}

className="w-full h-10 px-3 rounded-md border border-gray-300 bg-white text-gray-900 dark:border-gray-600 dark:bg-gray-700 dark:text-white focus:outline-none focus:ring-2 focus:ring-blue-500 focus:border-blue-500"

/>

</div>

</div>

<div className="space-y-2">

<Label htmlFor="timezone" className="dark:text-gray-300">

Timezone

</Label>

<input

id="timezone"

value={settings.timezone}

onChange={(e) =>

handleInputChange("timezone", e.target.value)

}

className="w-full h-10 px-3 rounded-md border border-gray-300 bg-white text-gray-900 dark:border-gray-600 dark:bg-gray-700 dark:text-white focus:outline-none focus:ring-2 focus:ring-blue-500 focus:border-blue-500"

/>

</div>

</CardContent>

</Card>

</TabsContent>

{/\* Security Settings \*/}

<TabsContent value="security" className="space-y-4">

<Card className="border-gray-200 shadow-sm dark:border-gray-700 dark:bg-gray-800">

<CardHeader>

<CardTitle className="flex items-center gap-2 dark:text-white">

<Shield className="h-5 w-5 text-gray-600 dark:text-gray-400" />

Security Settings

</CardTitle>

<CardDescription className="dark:text-gray-400">

Configure authentication and security policies

</CardDescription>

</CardHeader>

<CardContent className="space-y-6">

<div className="flex items-center justify-between">

<div className="space-y-0.5">

<Label htmlFor="twoFactorAuth" className="dark:text-gray-300">

Two-Factor Authentication

</Label>

<div className="text-sm text-gray-500 dark:text-gray-400">

Require 2FA for all users

</div>

</div>

<Switch

id="twoFactorAuth"

checked={settings.twoFactorAuth}

onCheckedChange={(checked) =>

handleInputChange("twoFactorAuth", checked)

}

/>

</div>

<div className="space-y-2">

<Label htmlFor="sessionTimeout" className="dark:text-gray-300">

Session Timeout (minutes)

</Label>

<input

id="sessionTimeout"

type="number"

value={settings.sessionTimeout}

onChange={(e) =>

handleInputChange(

"sessionTimeout",

parseInt(e.target.value)

)

}

className="w-full h-10 px-3 rounded-md border border-gray-300 bg-white text-gray-900 dark:border-gray-600 dark:bg-gray-700 dark:text-white focus:outline-none focus:ring-2 focus:ring-blue-500 focus:border-blue-500"

/>

</div>

<div className="space-y-2">

<Label htmlFor="passwordPolicy" className="dark:text-gray-300">

Password Policy

</Label>

<div className="flex gap-2">

{["basic", "strong", "very-strong"].map((policy) => (

<Button

key={policy}

variant={

settings.passwordPolicy === policy

? "default"

: "outline"

}

onClick={() =>

handleInputChange("passwordPolicy", policy)

}

className="capitalize dark:border-gray-600 dark:text-gray-300 dark:hover:bg-gray-700"

>

{policy.replace("-", " ")}

</Button>

))}

</div>

</div>

<div className="space-y-2">

<Label htmlFor="ipWhitelist" className="dark:text-gray-300">

IP Whitelist

</Label>

<input

id="ipWhitelist"

placeholder="192.168.1.1, 10.0.0.1"

value={settings.ipWhitelist}

onChange={(e) =>

handleInputChange("ipWhitelist", e.target.value)

}

className="w-full h-10 px-3 rounded-md border border-gray-300 bg-white text-gray-900 dark:border-gray-600 dark:bg-gray-700 dark:text-white focus:outline-none focus:ring-2 focus:ring-blue-500 focus:border-blue-500"

/>

<div className="text-sm text-gray-500 dark:text-gray-400">

Comma-separated list of allowed IP addresses

</div>

</div>

</CardContent>

</Card>

</TabsContent>

{/\* Notification Settings \*/}

<TabsContent value="notifications" className="space-y-4">

<Card className="border-gray-200 shadow-sm dark:border-gray-700 dark:bg-gray-800">

<CardHeader>

<CardTitle className="flex items-center gap-2 dark:text-white">

<Bell className="h-5 w-5 text-gray-600 dark:text-gray-400" />

Notification Settings

</CardTitle>

<CardDescription className="dark:text-gray-400">

Configure how you receive alerts and reports

</CardDescription>

</CardHeader>

<CardContent className="space-y-6">

{[

{

id: "emailNotifications",

label: "Email Notifications",

description: "Receive important alerts via email",

},

{

id: "slackNotifications",

label: "Slack Notifications",

description: "Send alerts to your Slack channel",

},

{

id: "securityAlerts",

label: "Security Alerts",

description: "Get notified about security events",

},

{

id: "weeklyReports",

label: "Weekly Reports",

description: "Receive weekly analytics reports",

},

].map((item) => (

<div

key={item.id}

className="flex items-center justify-between"

>

<div className="space-y-0.5">

<Label htmlFor={item.id} className="dark:text-gray-300">

{item.label}

</Label>

<div className="text-sm text-gray-500 dark:text-gray-400">

{item.description}

</div>

</div>

<Switch

id={item.id}

checked={

settings[item.id as keyof typeof settings] as boolean

}

onCheckedChange={(checked) =>

handleInputChange(item.id, checked)

}

/>

</div>

))}

</CardContent>

</Card>

</TabsContent>

{/\* API Settings \*/}

<TabsContent value="api" className="space-y-4">

<Card className="border-gray-200 shadow-sm dark:border-gray-700 dark:bg-gray-800">

<CardHeader>

<CardTitle className="flex items-center gap-2 dark:text-white">

<Database className="h-5 w-5 text-gray-600 dark:text-gray-400" />

API Settings

</CardTitle>

<CardDescription className="dark:text-gray-400">

Configure API rate limits and webhook integrations

</CardDescription>

</CardHeader>

<CardContent className="space-y-6">

<div className="space-y-2">

<Label htmlFor="apiRateLimit" className="dark:text-gray-300">

API Rate Limit (requests/minute)

</Label>

<input

id="apiRateLimit"

type="number"

value={settings.apiRateLimit}

onChange={(e) =>

handleInputChange("apiRateLimit", parseInt(e.target.value))

}

className="w-full h-10 px-3 rounded-md border border-gray-300 bg-white text-gray-900 dark:border-gray-600 dark:bg-gray-700 dark:text-white focus:outline-none focus:ring-2 focus:ring-blue-500 focus:border-blue-500"

/>

</div>

<div className="flex items-center justify-between">

<div className="space-y-0.5">

<Label

htmlFor="enableWebhooks"

className="dark:text-gray-300"

>

Enable Webhooks

</Label>

<div className="text-sm text-gray-500 dark:text-gray-400">

Send events to external services

</div>

</div>

<Switch

id="enableWebhooks"

checked={settings.enableWebhooks}

onCheckedChange={(checked) =>

handleInputChange("enableWebhooks", checked)

}

/>

</div>

{settings.enableWebhooks && (

<div className="space-y-2">

<Label htmlFor="webhookUrl" className="dark:text-gray-300">

Webhook URL

</Label>

<input

id="webhookUrl"

placeholder="https://api.example.com/webhook"

value={settings.webhookUrl}

onChange={(e) =>

handleInputChange("webhookUrl", e.target.value)

}

className="w-full h-10 px-3 rounded-md border border-gray-300 bg-white text-gray-900 dark:border-gray-600 dark:bg-gray-700 dark:text-white focus:outline-none focus:ring-2 focus:ring-blue-500 focus:border-blue-500"

/>

</div>

)}

<div className="p-4 bg-gray-50 rounded-lg dark:bg-gray-700/50">

<div className="flex items-center gap-2 mb-2">

<Key className="h-4 w-4 text-gray-600 dark:text-gray-400" />

<Label className="dark:text-gray-300">API Keys</Label>

</div>

<div className="text-sm text-gray-600 dark:text-gray-400 mb-3">

Manage your API keys for integration

</div>

<Button

variant="outline"

className="dark:border-gray-600 dark:text-gray-300 dark:hover:bg-gray-700"

>

Generate New API Key

</Button>

</div>

</CardContent>

</Card>

</TabsContent>

</Tabs>

</div>

);

}

## D:\Saif-Solutions\projects\authbase-stable\src\pages\Users.tsx

import { useState } from "react";

import {

MoreHorizontal,

Edit,

Trash2,

Plus,

Search,

Filter,

Download,

Mail,

Phone,

Calendar,

CheckCircle,

XCircle,

Users as UsersIcon,

} from "lucide-react";

import { Button } from "@/components/ui/button";

// import { Input } from "@/components/ui/input";

import {

Card,

CardContent,

CardHeader,

CardTitle,

CardDescription,

} from "@/components/ui/card";

import {

Table,

TableBody,

TableCell,

TableHead,

TableHeader,

TableRow,

} from "@/components/ui/table";

import { Badge } from "@/components/ui/badge";

import {

DropdownMenu,

DropdownMenuContent,

DropdownMenuItem,

DropdownMenuTrigger,

} from "@/components/ui/dropdown-menu";

const mockUsers = [

{

id: "1",

name: "John Smith",

email: "john@example.com",

phone: "+1 (555) 123-4567",

role: "Admin",

status: "Active",

lastLogin: "2024-01-15T10:30:00Z",

createdAt: "2024-01-10T14:20:00Z",

verified: true,

},

{

id: "2",

name: "Sarah Johnson",

email: "sarah@company.com",

phone: "+1 (555) 987-6543",

role: "User",

status: "Active",

lastLogin: "2024-01-14T16:45:00Z",

createdAt: "2024-01-12T09:15:00Z",

verified: true,

},

{

id: "3",

name: "Mike Chen",

email: "mike@startup.io",

phone: "+1 (555) 456-7890",

role: "User",

status: "Inactive",

lastLogin: "2024-01-05T11:20:00Z",

createdAt: "2024-01-03T13:40:00Z",

verified: false,

},

{

id: "4",

name: "Emily Davis",

email: "emily@techcorp.com",

phone: "+1 (555) 234-5678",

role: "Moderator",

status: "Active",

lastLogin: "2024-01-15T08:15:00Z",

createdAt: "2024-01-08T10:30:00Z",

verified: true,

},

];

export function Users() {

const [searchTerm, setSearchTerm] = useState("");

const [users] = useState(mockUsers);

const filteredUsers = users.filter(

(user) =>

user.name.toLowerCase().includes(searchTerm.toLowerCase()) ||

user.email.toLowerCase().includes(searchTerm.toLowerCase())

);

const formatDate = (dateString: string) => {

return new Date(dateString).toLocaleDateString("en-US", {

month: "short",

day: "numeric",

year: "numeric",

});

};

// Statistics

const totalUsers = users.length;

const activeUsers = users.filter((u) => u.status === "Active").length;

const verifiedUsers = users.filter((u) => u.verified).length;

const adminUsers = users.filter((u) => u.role === "Admin").length;

return (

<div className="space-y-6">

{/\* Page Header \*/}

<div className="flex flex-col sm:flex-row justify-between items-start sm:items-center gap-4">

<div className="space-y-1">

<h1 className="text-2xl font-bold text-gray-900 dark:text-white">

User Management

</h1>

<p className="text-gray-600 dark:text-gray-400">

Manage and monitor all registered users in your system

</p>

</div>

<div className="flex items-center gap-3">

<Button

variant="outline"

className="flex items-center gap-2 dark:border-gray-600 dark:text-gray-300 dark:hover:bg-gray-700"

>

<Download className="h-4 w-4" />

Export

</Button>

<Button className="flex items-center gap-2 bg-blue-600 hover:bg-blue-700 dark:bg-blue-700 dark:hover:bg-blue-800">

<Plus className="h-4 w-4" />

Add User

</Button>

</div>

</div>

{/\* Statistics Cards \*/}

<div className="grid gap-4 md:grid-cols-2 lg:grid-cols-4">

<Card className="border-gray-200 shadow-sm dark:border-gray-700 dark:bg-gray-800">

<CardContent className="p-4 flex items-center justify-between">

<div>

<div className="text-2xl font-bold text-gray-900 dark:text-white">

{totalUsers}

</div>

<div className="text-sm text-gray-600 dark:text-gray-400">

Total Users

</div>

</div>

<div className="p-2 bg-blue-50 rounded-lg dark:bg-blue-900/50">

<UsersIcon className="h-5 w-5 text-blue-600 dark:text-blue-400" />

</div>

</CardContent>

</Card>

<Card className="border-gray-200 shadow-sm dark:border-gray-700 dark:bg-gray-800">

<CardContent className="p-4 flex items-center justify-between">

<div>

<div className="text-2xl font-bold text-green-600 dark:text-green-400">

{activeUsers}

</div>

<div className="text-sm text-gray-600 dark:text-gray-400">

Active Users

</div>

</div>

<div className="p-2 bg-green-50 rounded-lg dark:bg-green-900/50">

<CheckCircle className="h-5 w-5 text-green-600 dark:text-green-400" />

</div>

</CardContent>

</Card>

<Card className="border-gray-200 shadow-sm dark:border-gray-700 dark:bg-gray-800">

<CardContent className="p-4 flex items-center justify-between">

<div>

<div className="text-2xl font-bold text-purple-600 dark:text-purple-400">

{verifiedUsers}

</div>

<div className="text-sm text-gray-600 dark:text-gray-400">

Verified

</div>

</div>

<div className="p-2 bg-purple-50 rounded-lg dark:bg-purple-900/50">

<Mail className="h-5 w-5 text-purple-600 dark:text-purple-400" />

</div>

</CardContent>

</Card>

<Card className="border-gray-200 shadow-sm dark:border-gray-700 dark:bg-gray-800">

<CardContent className="p-4 flex items-center justify-between">

<div>

<div className="text-2xl font-bold text-orange-600 dark:text-orange-400">

{adminUsers}

</div>

<div className="text-sm text-gray-600 dark:text-gray-400">

Admins

</div>

</div>

<div className="p-2 bg-orange-50 rounded-lg dark:bg-orange-900/50">

<Filter className="h-5 w-5 text-orange-600 dark:text-orange-400" />

</div>

</CardContent>

</Card>

</div>

{/\* Search and Filters \*/}

<Card className="border-gray-200 shadow-sm dark:border-gray-700 dark:bg-gray-800">

<CardContent className="p-4">

<div className="flex flex-col lg:flex-row gap-4 items-start lg:items-center justify-between">

<div className="flex flex-col sm:flex-row gap-3 w-full lg:w-auto">

<div className="relative w-full sm:w-80">

<Search className="absolute left-3 top-1/2 h-4 w-4 -translate-y-1/2 text-gray-500 dark:text-gray-400 z-10" />

<input

type="text"

placeholder="Search by user, IP, or location..."

value={searchTerm}

onChange={(e) => setSearchTerm(e.target.value)}

className="search-input-fallback focus:border-blue-500 focus:ring-2 focus:ring-blue-500/20"

/>

</div>

<Button

variant="outline"

className="flex items-center gap-2 border-gray-300 dark:border-gray-600 dark:text-gray-300 dark:hover:bg-gray-700"

>

<Filter className="h-4 w-4" />

Filter

</Button>

</div>

<div className="text-sm text-gray-600 dark:text-gray-400 whitespace-nowrap">

Showing {filteredUsers.length} of {totalUsers} users

</div>

</div>

</CardContent>

</Card>

{/\* Users Table \*/}

<Card className="border-gray-200 shadow-sm overflow-hidden dark:border-gray-700 dark:bg-gray-800">

<CardHeader className="bg-gray-50/50 border-b border-gray-200 dark:bg-gray-800/50 dark:border-gray-700">

<CardTitle className="flex items-center gap-2 dark:text-white">

Users

<Badge

variant="secondary"

className="bg-blue-100 text-blue-800 border-blue-200 dark:bg-blue-900 dark:text-blue-200 dark:border-blue-700"

>

{filteredUsers.length}

</Badge>

</CardTitle>

<CardDescription className="dark:text-gray-400">

Manage user accounts, roles, and access permissions

</CardDescription>

</CardHeader>

<CardContent className="p-0">

<div className="overflow-x-auto">

<Table>

<TableHeader>

<TableRow className="bg-gray-50/80 hover:bg-gray-50 dark:bg-gray-800/80 dark:hover:bg-gray-800">

<TableHead className="font-semibold text-gray-900 dark:text-white py-4">

User

</TableHead>

<TableHead className="font-semibold text-gray-900 dark:text-white py-4">

Contact

</TableHead>

<TableHead className="font-semibold text-gray-900 dark:text-white py-4">

Role

</TableHead>

<TableHead className="font-semibold text-gray-900 dark:text-white py-4">

Status

</TableHead>

<TableHead className="font-semibold text-gray-900 dark:text-white py-4">

Last Login

</TableHead>

<TableHead className="font-semibold text-gray-900 dark:text-white py-4 text-right">

Actions

</TableHead>

</TableRow>

</TableHeader>

<TableBody>

{filteredUsers.map((user) => (

<TableRow

key={user.id}

className="hover:bg-gray-50/50 border-b border-gray-100 dark:hover:bg-gray-700/50 dark:border-gray-700"

>

<TableCell className="py-4">

<div className="flex items-center gap-3">

<div className="h-10 w-10 rounded-full bg-gradient-to-br from-blue-500 to-blue-600 flex items-center justify-center shadow-sm">

<span className="text-sm font-semibold text-white">

{user.name

.split(" ")

.map((n) => n[0])

.join("")}

</span>

</div>

<div>

<div className="font-semibold text-gray-900 dark:text-white">

{user.name}

</div>

<div className="flex items-center gap-1 mt-1">

{user.verified ? (

<CheckCircle className="h-3 w-3 text-green-500 dark:text-green-400" />

) : (

<XCircle className="h-3 w-3 text-gray-400 dark:text-gray-500" />

)}

<div className="text-sm text-gray-500 dark:text-gray-400">

{user.email}

</div>

</div>

</div>

</div>

</TableCell>

<TableCell className="py-4">

<div className="space-y-1">

<div className="flex items-center gap-2 text-sm text-gray-600 dark:text-gray-400">

<Mail className="h-3 w-3" />

{user.email}

</div>

<div className="flex items-center gap-2 text-sm text-gray-600 dark:text-gray-400">

<Phone className="h-3 w-3" />

{user.phone}

</div>

</div>

</TableCell>

<TableCell className="py-4">

<Badge

variant="outline"

className={

user.role === "Admin"

? "border-blue-200 bg-blue-50 text-blue-700 font-medium dark:border-blue-700 dark:bg-blue-900/50 dark:text-blue-300"

: user.role === "Moderator"

? "border-purple-200 bg-purple-50 text-purple-700 font-medium dark:border-purple-700 dark:bg-purple-900/50 dark:text-purple-300"

: "border-gray-200 bg-gray-50 text-gray-700 font-medium dark:border-gray-600 dark:bg-gray-700/50 dark:text-gray-300"

}

>

{user.role}

</Badge>

</TableCell>

<TableCell className="py-4">

<Badge

variant={

user.status === "Active" ? "default" : "secondary"

}

className={

user.status === "Active"

? "font-medium dark:bg-green-900 dark:text-green-200 dark:border-green-700"

: "font-medium dark:bg-gray-700 dark:text-gray-200 dark:border-gray-600"

}

>

{user.status}

</Badge>

</TableCell>

<TableCell className="py-4">

<div className="flex items-center gap-2 text-sm text-gray-600 dark:text-gray-400">

<Calendar className="h-3 w-3" />

{formatDate(user.lastLogin)}

</div>

</TableCell>

<TableCell className="py-4 text-right">

<DropdownMenu>

<DropdownMenuTrigger asChild>

<Button

variant="ghost"

size="icon"

className="h-8 w-8 hover:bg-gray-100 dark:hover:bg-gray-700"

>

<MoreHorizontal className="h-4 w-4" />

</Button>

</DropdownMenuTrigger>

<DropdownMenuContent

align="end"

className="w-48 dark:bg-gray-800 dark:border-gray-700"

>

<DropdownMenuItem className="flex items-center gap-2 cursor-pointer text-sm dark:hover:bg-gray-700">

<Edit className="h-4 w-4" />

Edit User

</DropdownMenuItem>

<DropdownMenuItem className="flex items-center gap-2 cursor-pointer text-sm text-red-600 focus:text-red-600 dark:hover:bg-gray-700">

<Trash2 className="h-4 w-4" />

Delete User

</DropdownMenuItem>

</DropdownMenuContent>

</DropdownMenu>

</TableCell>

</TableRow>

))}

</TableBody>

</Table>

</div>

</CardContent>

</Card>

</div>

);

}

## D:\Saif-Solutions\projects\authbase-stable\vite.config.ts

import { defineConfig } from "vite";

import react from "@vitejs/plugin-react";

import path from "path";

// https://vitejs.dev/config/

export default defineConfig({

plugins: [react()],

resolve: {

alias: {

"@": path.resolve(\_\_dirname, "./src"),

},

},

});

## D:\Saif-Solutions\projects\authbase-stable\src\services\authAPI.ts

import axios from "axios";

import {

User,

LoginCredentials,

CreateUserData,

UpdateUserData,

Session,

Analytics,

} from "@/types/api";

const API\_BASE\_URL = "http://localhost:3001/api";

const api = axios.create({

baseURL: API\_BASE\_URL,

headers: {

"Content-Type": "application/json",

},

});

// Request interceptor to add auth token

api.interceptors.request.use(

(config) => {

const token = localStorage.getItem("authToken");

if (token) {

config.headers.Authorization = `Bearer ${token}`;

}

return config;

},

(error) => {

return Promise.reject(error);

}

);

// Response interceptor to handle errors

api.interceptors.response.use(

(response) => response,

(error) => {

if (error.response?.status === 401) {

localStorage.removeItem("authToken");

window.location.href = "/login";

}

return Promise.reject(error);

}

);

export const authAPI = {

// Authentication

login: (credentials: LoginCredentials) =>

api.post<{ token: string; user: User }>("/auth/login", credentials),

logout: () => api.post("/auth/logout"),

refreshToken: () => api.post<{ token: string }>("/auth/refresh"),

// Users

getUsers: () => api.get<{ users: User[] }>("/users"),

getUser: (id: string) => api.get<{ user: User }>(`/users/${id}`),

createUser: (userData: CreateUserData) =>

api.post<{ user: User }>("/users", userData),

updateUser: (id: string, userData: UpdateUserData) =>

api.put<{ user: User }>(`/users/${id}`, userData),

deleteUser: (id: string) => api.delete(`/users/${id}`),

// Sessions

getSessions: () => api.get<{ sessions: Session[] }>("/sessions"),

revokeSession: (sessionId: string) => api.delete(`/sessions/${sessionId}`),

// Analytics

getAnalytics: () => api.get<Analytics>("/analytics"),

};

export default api;

## 🚀 **STEP 2: CREATE PROPER TYPES FOR API**

**First, install axios:**

bash

npm install axios

**Then create proper TypeScript interfaces:**

## D:\Saif-Solutions\projects\authbase-stable\src\types\api.ts

export interface User {

id: string;

name: string;

email: string;

phone?: string;

role: "Admin" | "User" | "Moderator";

status: "Active" | "Inactive";

lastLogin: string;

createdAt: string;

verified: boolean;

}

export interface LoginCredentials {

email: string;

password: string;

}

export interface CreateUserData {

name: string;

email: string;

password: string;

role: "Admin" | "User" | "Moderator";

phone?: string;

}

export interface UpdateUserData {

name?: string;

email?: string;

role?: "Admin" | "User" | "Moderator";

status?: "Active" | "Inactive";

phone?: string;

}

export interface Session {

id: string;

userId: string;

userAgent: string;

ipAddress: string;

lastActivity: string;

createdAt: string;

}

export interface Analytics {

totalUsers: number;

activeSessions: number;

dailyRequests: number;

successRate: number;

}

## First, install the Tabs component:

bash

npx shadcn@latest add tabs switch label

## D:\Saif-Solutions\projects\authbase-stable\src\components\ui\SkeletonLoader.tsx

import { Card, CardContent, CardHeader } from "@/components/ui/card";

export function SkeletonLoader() {

return (

<div className="space-y-6 animate-pulse">

{/\* Metrics Skeleton \*/}

<div className="grid gap-4 md:grid-cols-2 lg:grid-cols-4">

{[1, 2, 3, 4].map((i) => (

<Card key={i} className="border-gray-200 dark:border-gray-700">

<CardHeader className="flex flex-row items-center justify-between space-y-0 pb-3">

<div className="h-4 w-20 bg-gray-200 dark:bg-gray-700 rounded"></div>

<div className="h-8 w-8 bg-gray-200 dark:bg-gray-700 rounded-lg"></div>

</CardHeader>

<CardContent>

<div className="h-8 w-16 bg-gray-200 dark:bg-gray-700 rounded mb-2"></div>

<div className="h-3 w-24 bg-gray-200 dark:bg-gray-700 rounded"></div>

</CardContent>

</Card>

))}

</div>

{/\* Table Skeleton \*/}

<Card className="border-gray-200 dark:border-gray-700">

<CardHeader>

<div className="h-6 w-32 bg-gray-200 dark:bg-gray-700 rounded"></div>

</CardHeader>

<CardContent>

<div className="space-y-3">

{[1, 2, 3, 4].map((i) => (

<div key={i} className="flex items-center space-x-4">

<div className="h-10 w-10 bg-gray-200 dark:bg-gray-700 rounded-full"></div>

<div className="space-y-2 flex-1">

<div className="h-4 w-32 bg-gray-200 dark:bg-gray-700 rounded"></div>

<div className="h-3 w-24 bg-gray-200 dark:bg-gray-700 rounded"></div>

</div>

</div>

))}

</div>

</CardContent>

</Card>

</div>

);

}

## 5. TOAST NOTIFICATIONS

**Install:**

bash

npm install sonner

## D:\Saif-Solutions\projects\authbase-stable\src\components\ui\Toaster.tsx

import { Toaster as SonnerToaster } from "sonner";

export function Toaster() {

return (

<SonnerToaster

position="top-right"

toastOptions={{

style: {

background: "hsl(var(--background))",

color: "hsl(var(--foreground))",

border: "1px solid hsl(var(--border))",

},

}}

/>

);

}

## D:\Saif-Solutions\projects\authbase-stable\src\components\ui\LoadingButton.tsx jsx

import { Button } from "@/components/ui/button";

import { Loader2 } from "lucide-react";

interface LoadingButtonProps extends React.ButtonHTMLAttributes<HTMLButtonElement> {

loading?: boolean;

children: React.ReactNode;

}

export function LoadingButton({ loading, children, ...props }: LoadingButtonProps) {

return (

<Button {...props} disabled={loading || props.disabled}>

{loading && <Loader2 className="mr-2 h-4 w-4 animate-spin" />}

{children}

</Button>

);

}

## D:\Saif-Solutions\projects\authbase-stable\src\components\ui\EmptyState.tsx

import { LucideIcon } from "lucide-react";

import { Button } from "@/components/ui/button";

interface EmptyStateProps {

icon: LucideIcon;

title: string;

description: string;

action?: {

label: string;

onClick: () => void;

};

}

export function EmptyState({

icon: Icon,

title,

description,

action,

}: EmptyStateProps) {

return (

<div className="text-center py-12">

<div className="mx-auto h-16 w-16 bg-gradient-to-br from-blue-500 to-blue-600 rounded-full flex items-center justify-center mb-4">

<Icon className="h-8 w-8 text-white" />

</div>

<h3 className="text-lg font-semibold text-gray-900 dark:text-white mb-2">

{title}

</h3>

<p className="text-gray-500 dark:text-gray-400 max-w-sm mx-auto mb-6">

{description}

</p>

{action && <Button onClick={action.onClick}>{action.label}</Button>}

</div>

);

}

## D:\Saif-Solutions\projects\authbase-stable\src\components\ui\badge.tsx

import \* as React from "react";

import { type VariantProps } from "class-variance-authority";

import { cn } from "@/lib/utils";

import { badgeVariants } from "@/lib/variants";

export interface BadgeProps

extends React.HTMLAttributes<HTMLDivElement>,

VariantProps<typeof badgeVariants> {}

function Badge({ className, variant, ...props }: BadgeProps) {

return (

<div className={cn(badgeVariants({ variant }), className)} {...props} />

);

}

export { Badge };

## D:\Saif-Solutions\projects\authbase-stable\src\lib\darkModeUtils.ts

export const darkModeClasses = {

text: {

primary: "dark:text-white",

secondary: "dark:text-gray-300",

muted: "dark:text-gray-400",

},

bg: {

card: "dark:bg-gray-800",

sidebar: "dark:bg-gray-900",

},

border: {

default: "dark:border-gray-700",

},

};

## D:\Saif-Solutions\projects\authbase-stable\src\lib\utils.ts

import { type ClassValue, clsx } from "clsx";

import { twMerge } from "tailwind-merge";

export function cn(...inputs: ClassValue[]) {

return twMerge(clsx(inputs));

}

## D:\Saif-Solutions\projects\authbase-stable\src\lib\variants.ts

import { cva } from "class-variance-authority";

export const badgeVariants = cva(

"inline-flex items-center rounded-full border px-2.5 py-0.5 text-xs font-semibold transition-colors focus:outline-none focus:ring-2 focus:ring-ring focus:ring-offset-2",

{

variants: {

variant: {

default:

"border-transparent bg-primary text-primary-foreground hover:bg-primary/80",

secondary:

"border-transparent bg-secondary text-secondary-foreground hover:bg-secondary/80",

destructive:

"border-transparent bg-destructive text-destructive-foreground hover:bg-destructive/80",

outline: "text-foreground",

},

},

defaultVariants: {

variant: "default",

},

}

);

## D:\Saif-Solutions\projects\authbase-stable\src\components\ui\input.tsx

import \* as React from "react"

import { cn } from "@/lib/utils"

const Input = React.forwardRef<HTMLInputElement, React.ComponentProps<"input">>(

  ({ className, type, ...props }, ref) => {

    return (

      <input

        type={type}

        className={cn(

          "flex h-10 w-full rounded-md border border-input bg-background px-3 py-2 text-base ring-offset-background file:border-0 file:bg-transparent file:text-sm file:font-medium file:text-foreground placeholder:text-muted-foreground focus-visible:outline-none focus-visible:ring-2 focus-visible:ring-ring focus-visible:ring-offset-2 disabled:cursor-not-allowed disabled:opacity-50 md:text-sm",

          className

        )}

        ref={ref}

        {...props}

      />

    )

  }

)

Input.displayName = "Input"

export { Input }

## END

## IMMEDIATE ACTION PLAN

## **WEEK** 1: AUTHENTICATION & REAL DATA FLOW

## Fix Authentication Flow

typescript

*// Create src/contexts/AuthContext.tsx*

import { createContext, useContext, useState, useEffect } from 'react';

import { authAPI } from '@/services/authAPI';

interface AuthContextType {

user: User | null;

login: (credentials: LoginCredentials) => Promise<void>;

logout: () => void;

isLoading: boolean;

}

const AuthContext = createContext<AuthContextType | undefined>(undefined);

export function AuthProvider({ children }: { children: React.ReactNode }) {

const [user, setUser] = useState<User | null>(null);

const [isLoading, setIsLoading] = useState(true);

useEffect(() => {

const token = localStorage.getItem('authToken');

if (token) {

*// Verify token and get user data*

authAPI.refreshToken().then(response => {

setUser(response.data.user);

}).catch(() => {

localStorage.removeItem('authToken');

});

}

setIsLoading(false);

}, []);

const login = async (credentials: LoginCredentials) => {

const response = await authAPI.login(credentials);

localStorage.setItem('authToken', response.data.token);

setUser(response.data.user);

};

const logout = () => {

localStorage.removeItem('authToken');

setUser(null);

window.location.href = '/login';

};

return (

<AuthContext.Provider value={{ user, login, logout, isLoading }}>

{children}

</AuthContext.Provider>

);

}

## **2. Add Route Protection**

typescript

*// Create src/components/auth/ProtectedRoute.tsx*

import { Navigate } from 'react-router-dom';

import { useAuth } from '@/contexts/AuthContext';

export function ProtectedRoute({ children }: { children: React.ReactNode }) {

const { user, isLoading } = useAuth();

if (isLoading) {

return <div>Loading...</div>;

}

if (!user) {

return <Navigate to="/login" replace />;

}

return <>{children}</>;

}

*// Update App.tsx*

<Router>

<AuthProvider>

<Routes>

<Route path="/login" element={<Login />} />

<Route path="/\*" element={

<ProtectedRoute>

<Layout>

{*/\* ... existing routes ... \*/*}

</Layout>

</ProtectedRoute>

} />

</Routes>

</AuthProvider>

</Router>

## **3. Connect to Real Backend**

Create a mock backend for demonstration:

bash

npm install json-server concurrently

json

*// Create db.json*

{

"users": [

{

"id": "1",

"name": "Admin User",

"email": "admin@authbase.com",

"role": "Admin",

"status": "Active",

"lastLogin": "2024-01-15T10:30:00Z",

"createdAt": "2024-01-10T14:20:00Z",

"verified": true

}

],

"sessions": [],

"analytics": {

"totalUsers": 1248,

"activeSessions": 342,

"dailyRequests": 24800,

"successRate": 99.8

}

}

## **WEEK 2:** ENHANCE **USER EXPERIENCE**

## **1. Add Loading States**

typescript

*// Update Users.tsx with real data fetching*

import { useState, useEffect } from 'react';

import { authAPI } from '@/services/authAPI';

import { SkeletonLoader } from '@/components/ui/SkeletonLoader';

export function Users() {

const [users, setUsers] = useState<User[]>([]);

const [isLoading, setIsLoading] = useState(true);

useEffect(() => {

const fetchUsers = async () => {

try {

const response = await authAPI.getUsers();

setUsers(response.data.users);

} catch (error) {

console.error('Failed to fetch users:', error);

} finally {

setIsLoading(false);

}

};

fetchUsers();

}, []);

if (isLoading) {

return <SkeletonLoader />;

}

*// ... rest of component*

}

## **2. Implement Real Search/Filter**

typescript

*// Enhanced search functionality*

const [filters, setFilters] = useState({

role: '',

status: '',

verified: ''

});

const filteredUsers = users.filter(user => {

const matchesSearch = user.name.toLowerCase().includes(searchTerm.toLowerCase()) ||

user.email.toLowerCase().includes(searchTerm.toLowerCase());

const matchesRole = !filters.role || user.role === filters.role;

const matchesStatus = !filters.status || user.status === filters.status;

const matchesVerified = filters.verified === '' ||

(filters.verified === 'true' && user.verified) ||

(filters.verified === 'false' && !user.verified);

return matchesSearch && matchesRole && matchesStatus && matchesVerified;

});

## WEEK **3: DEPLOYMENT READY**

## **1. Add Environment Configuration**

typescript

*// Create src/config/env.ts*

export const config = {

apiBaseUrl: import.meta.env.VITE\_API\_BASE\_URL || 'http://localhost:3001/api',

appName: import.meta.env.VITE\_APP\_NAME || 'AuthBase Pro',

nodeEnv: import.meta.env.NODE\_ENV || 'development'

};

## **2. Performance Optimization**

typescript

*// Implement lazy loading for pages*

import { lazy, Suspense } from 'react';

const Dashboard = lazy(() => import('@/pages/Dashboard'));

const Users = lazy(() => import('@/pages/Users'));

*// In App.tsx*

<Suspense fallback={<div>Loading...</div>}>

<Routes>

<Route path="/dashboard" element={<Dashboard />} />

<Route path="/users" element={<Users />} />

</Routes>

</Suspense>

## 🎯 SPECIFIC CODE FIXES NEEDED

## **1. Fix Input Component Issues**

Your current implementation uses raw <input> elements instead of shadcn/ui Input components. Update:

typescript

*// In Login.tsx - replace raw inputs with:*

import { Input } from "@/components/ui/input";

<Input

type="email"

placeholder="Email address"

value={email}

onChange={(e) => setEmail(e.target.value)}

className="pl-10"

/>

## **2. Add Missing shadcn/ui Components**

bash

npx shadcn@latest add skeleton

npx shadcn@latest add dialog

npx shadcn@latest add form

## **3. Enhance Error Handling**

typescript

*// Create src/utils/errorHandler.ts*

import { toast } from 'sonner';

export const handleAPIError = (error: any, fallbackMessage?: string) => {

const message = error.response?.data?.message || fallbackMessage || 'An error occurred';

toast.error(message);

return message;

};

## **🚀 DEPLOYMENT STRATEGY**

## **1. Docker Configuration**

dockerfile

# Create Dockerfile

FROM node:18-alpine

WORKDIR /app

COPY package\*.json ./

RUN npm ci --only=production

COPY . .

RUN npm run build

EXPOSE 4173

CMD ["npm", "run", "preview"]

## **2. Vite Configuration for Production**

typescript

*// Update vite.config.ts*

export default defineConfig({

plugins: [react()],

resolve: {

alias: {

'@': path.resolve(\_\_dirname, './src'),

},

},

build: {

rollupOptions: {

output: {

manualChunks: {

vendor: ['react', 'react-dom'],

ui: ['@/components/ui']

}

}

}

}

})

## **📈 FINAL ASSESSMENT**

**Current Status:** 85% Complete  
**Confidence Level:** High - Foundation is excellent

**Strengths:**

* Professional UI/UX design
* Comprehensive component structure
* Excellent TypeScript implementation
* Proper separation of concerns

**Immediate Wins:**

1. Add authentication context (2-3 hours)
2. Connect to mock API (1-2 hours)
3. Add loading states (2-3 hours)

**Estimated Time to Production Ready:** 1-2 weeks