**WorkShop – Project**

**1. DashboardLayout.tsx**

import React, { useState } from 'react';

import { Link, useLocation, useNavigate } from 'react-router-dom';

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

import type { ReactNode } from 'react';

interface DashboardLayoutProps {

  children: ReactNode;

}

const DashboardLayout: React.FC<DashboardLayoutProps> = ({ children }) => {

  const { user, logout } = useAuth();

  const location = useLocation();

  const navigate = useNavigate();

  const [isMobileMenuOpen, setIsMobileMenuOpen] = useState(false);

  const handleLogout = () => {

    logout();

    navigate('/login');

  };

  const navigation = [

    { name: 'Dashboard', href: '/dashboard', icon: '📊' },

    { name: 'Profile', href: '/dashboard/profile', icon: '👤' },

    { name: 'Settings', href: '/dashboard/settings', icon: '⚙️' },

    ...(user?.userType === 'admin' ? [

      { name: 'Users', href: '/dashboard/users', icon: '👥' },

      { name: 'Analytics', href: '/dashboard/analytics', icon: '📈' }

    ] : []),

  ];

  return (

    <div className="min-h-screen bg-gray-50">

      {/\* Mobile menu overlay \*/}

      {isMobileMenuOpen && (

        <div className="fixed inset-0 flex z-40 md:hidden">

          <div className="fixed inset-0 bg-gray-600 bg-opacity-75" onClick={() => setIsMobileMenuOpen(false)} />

          <div className="relative flex-1 flex flex-col max-w-xs w-full bg-white">

            <div className="absolute top-0 right-0 -mr-12 pt-2">

              <button

                type="button"

                className="ml-1 flex items-center justify-center h-10 w-10 rounded-full focus:outline-none focus:ring-2 focus:ring-inset focus:ring-white"

                onClick={() => setIsMobileMenuOpen(false)}

              >

                <span className="sr-only">Close sidebar</span>

                <svg className="h-6 w-6 text-white" fill="none" viewBox="0 0 24 24" stroke="currentColor">

                  <path strokeLinecap="round" strokeLinejoin="round" strokeWidth={2} d="M6 18L18 6M6 6l12 12" />

                </svg>

              </button>

            </div>

            <div className="flex-1 h-0 pt-5 pb-4 overflow-y-auto">

              <div className="flex-shrink-0 flex items-center px-4">

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

              </div>

              <nav className="mt-5 px-2 space-y-1">

                {navigation.map((item) => (

                  <Link

                    key={item.name}

                    to={item.href}

                    className={`group flex items-center px-2 py-2 text-base font-medium rounded-md ${

                      location.pathname === item.href

                        ? 'bg-gray-100 text-gray-900'

                        : 'text-gray-600 hover:bg-gray-50 hover:text-gray-900'

                    }`}

                    onClick={() => setIsMobileMenuOpen(false)}

                  >

                    <span className="mr-3 text-lg">{item.icon}</span>

                    {item.name}

                  </Link>

                ))}

              </nav>

            </div>

            <div className="flex-shrink-0 flex border-t border-gray-200 p-4">

              <div className="flex items-center">

                <div className="bg-indigo-500 rounded-full h-10 w-10 flex items-center justify-center">

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

                    {user?.name.charAt(0).toUpperCase()}

                  </span>

                </div>

                <div className="ml-3">

                  <p className="text-sm font-medium text-gray-700">{user?.name}</p>

                  <p className="text-xs text-gray-500 capitalize">{user?.userType}</p>

                </div>

              </div>

            </div>

          </div>

        </div>

      )}

      {/\* Desktop sidebar \*/}

      <div className="hidden md:flex md:w-64 md:flex-col md:fixed md:inset-y-0">

        <div className="flex-1 flex flex-col min-h-0 border-r border-gray-200 bg-white">

          <div className="flex-1 flex flex-col pt-5 pb-4 overflow-y-auto">

            <div className="flex items-center flex-shrink-0 px-4">

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

            </div>

            <nav className="mt-5 flex-1 px-2 bg-white space-y-1">

              {navigation.map((item) => (

                <Link

                  key={item.name}

                  to={item.href}

                  className={`group flex items-center px-2 py-2 text-sm font-medium rounded-md transition-colors duration-200 ${

                    location.pathname === item.href

                      ? 'bg-gray-100 text-gray-900'

                      : 'text-gray-600 hover:bg-gray-50 hover:text-gray-900'

                  }`}

                >

                  <span className="mr-3 text-lg">{item.icon}</span>

                  {item.name}

                </Link>

              ))}

            </nav>

          </div>

          <div className="flex-shrink-0 flex border-t border-gray-200 p-4">

            <div className="flex items-center w-full">

              <div className="bg-indigo-500 rounded-full h-10 w-10 flex items-center justify-center">

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

                  {user?.name.charAt(0).toUpperCase()}

                </span>

              </div>

              <div className="ml-3 flex-1">

                <p className="text-sm font-medium text-gray-700">{user?.name}</p>

                <p className="text-xs text-gray-500 capitalize">{user?.userType}</p>

              </div>

              <button

                onClick={handleLogout}

                className="ml-2 p-2 text-gray-400 hover:text-gray-600 transition-colors duration-200"

                title="Logout"

              >

                <svg className="h-5 w-5" fill="none" viewBox="0 0 24 24" stroke="currentColor">

                  <path

                    strokeLinecap="round"

                    strokeLinejoin="round"

                    strokeWidth={2}

                    d="M17 16l4-4m0 0l-4-4m4 4H7m6 4v1a3 3 0 01-3 3H6a3 3 0 01-3-3V7a3 3 0 013-3h4a3 3 0 013 3v1"

                  />

                </svg>

              </button>

            </div>

          </div>

        </div>

      </div>

      {/\* Main content \*/}

      <div className="md:pl-64 flex flex-col flex-1">

        <div className="sticky top-0 z-10 md:hidden pl-1 pt-1 sm:pl-3 sm:pt-3 bg-gray-50">

          <button

            type="button"

            className="-ml-0.5 -mt-0.5 h-12 w-12 inline-flex items-center justify-center rounded-md text-gray-500 hover:text-gray-900 focus:outline-none focus:ring-2 focus:ring-inset focus:ring-indigo-500"

            onClick={() => setIsMobileMenuOpen(true)}

          >

            <span className="sr-only">Open sidebar</span>

            <svg className="h-6 w-6" fill="none" viewBox="0 0 24 24" stroke="currentColor">

              <path strokeLinecap="round" strokeLinejoin="round" strokeWidth={2} d="M4 6h16M4 12h16M4 18h16" />

            </svg>

          </button>

        </div>

        <main className="flex-1">

          <div className="py-6">

            <div className="max-w-7xl mx-auto px-4 sm:px-6 md:px-8">

              {/\* Header \*/}

              <div className="md:flex md:items-center md:justify-between mb-6">

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

                  <h2 className="text-2xl font-bold leading-7 text-gray-900 sm:text-3xl sm:truncate">

                    Welcome back, {user?.name}!

                  </h2>

                  <p className="mt-1 text-sm text-gray-500">

                    {user?.userType === 'admin' ? 'Administrator Dashboard' : 'User Dashboard'}

                  </p>

                </div>

                <div className="mt-4 flex md:mt-0 md:ml-4">

                  <button

                    onClick={handleLogout}

                    className="md:hidden bg-white py-2 px-4 border border-gray-300 rounded-md shadow-sm text-sm font-medium text-gray-700 hover:bg-gray-50 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-indigo-500 transition-colors duration-200"

                  >

                    Logout

                  </button>

                </div>

              </div>

              {/\* Page content \*/}

              <div className="bg-white shadow rounded-lg">

                {children}

              </div>

            </div>

          </div>

        </main>

      </div>

    </div>

  );

};

export default DashboardLayout;

**2.Login.tsx**

import React, { useState } from 'react';

import { Link, useNavigate } from 'react-router-dom';

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

import type { LoginFormData, FormErrors } from '../types/auth';

import { validateLoginForm } from '../utils/validation';

const Login: React.FC = () => {

  const navigate = useNavigate();

  const { login } = useAuth();

  const [formData, setFormData] = useState<LoginFormData>({

    email: '',

    password: '',

  });

  const [errors, setErrors] = useState<FormErrors>({});

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

  const [showPassword, setShowPassword] = useState(false);

  const handleChange = (e: React.ChangeEvent<HTMLInputElement>) => {

    const { name, value } = e.target;

    setFormData(prev => ({ ...prev, [name]: value }));

    // Clear error when user starts typing

    if (errors[name]) {

      setErrors(prev => ({ ...prev, [name]: '' }));

    }

  };

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

    e.preventDefault();

    // Validate form

    const validationErrors = validateLoginForm(formData);

    if (Object.keys(validationErrors).length > 0) {

      setErrors(validationErrors);

      return;

    }

    setIsLoading(true);

    setErrors({});

    try {

      const result = await login(formData);

      if (result.success) {

        navigate('/dashboard');

      } else {

        setErrors({ submit: result.error || 'Login failed' });

      }

    } catch (error) {

      setErrors({ submit: 'An unexpected error occurred' });

    } finally {

      setIsLoading(false);

    }

  };

  const demoAccounts = [

    {

      email: 'admin@example.com',

      password: 'Admin123!',

      type: 'Admin',

      color: 'from-purple-500 to-purple-600',

      icon: '👑'

    },

    {

      email: 'user@example.com',

      password: 'User123!',

      type: 'User',

      color: 'from-blue-500 to-blue-600',

      icon: '👤'

    }

  ];

  const fillDemo = (email: string, password: string) => {

    setFormData({ email, password });

    setErrors({});

  };

  return (

    <div className="min-h-screen flex items-center justify-center bg-gradient-to-br from-slate-50 via-blue-50 to-indigo-100 px-4 sm:px-6 lg:px-8 relative overflow-hidden">

      {/\* Background decorations \*/}

      <div className="absolute inset-0">

        <div className="absolute top-1/4 left-1/4 w-72 h-72 bg-gradient-to-r from-indigo-400/20 to-purple-400/20 rounded-full blur-3xl animate-pulse-slow"></div>

        <div className="absolute bottom-1/4 right-1/4 w-96 h-96 bg-gradient-to-r from-purple-400/20 to-pink-400/20 rounded-full blur-3xl animate-pulse-slow animation-delay-1000"></div>

      </div>

      <div className="max-w-md w-full space-y-8 relative z-10">

        <div className="text-center animate-fade-in-up">

          <div className="mx-auto w-16 h-16 bg-gradient-to-r from-indigo-600 to-purple-600 rounded-2xl flex items-center justify-center shadow-lg mb-6">

            <span className="text-white font-bold text-2xl">A</span>

          </div>

          <h2 className="text-3xl font-extrabold text-gray-900 mb-2">

            Welcome back

          </h2>

          <p className="text-gray-600">

            Sign in to your account to continue

          </p>

          <p className="mt-2 text-sm text-gray-500">

            Don't have an account?{' '}

            <Link

              to="/signup"

              className="font-medium text-indigo-600 hover:text-indigo-500 transition-colors duration-200 relative group"

            >

              Create one here

              <span className="absolute inset-x-0 bottom-0 h-0.5 bg-indigo-600 transform scale-x-0 group-hover:scale-x-100 transition-transform duration-300"></span>

            </Link>

          </p>

        </div>

        <div className="bg-white/80 backdrop-blur-lg py-8 px-6 sm:px-10 shadow-2xl rounded-2xl border border-white/20 animate-fade-in-up animation-delay-200">

          <form className="space-y-6" onSubmit={handleSubmit}>

            <div>

              <label htmlFor="email" className="block text-sm font-semibold text-gray-700 mb-2">

                Email address

              </label>

              <div className="relative">

                <div className="absolute inset-y-0 left-0 pl-3 flex items-center pointer-events-none">

                  <svg className="h-5 w-5 text-gray-400" fill="none" viewBox="0 0 24 24" stroke="currentColor">

                    <path strokeLinecap="round" strokeLinejoin="round" strokeWidth={2} d="M16 12a4 4 0 10-8 0 4 4 0 008 0zm0 0v1.5a2.5 2.5 0 005 0V12a9 9 0 10-9 9m4.5-1.206a8.959 8.959 0 01-4.5 1.207" />

                  </svg>

                </div>

                <input

                  id="email"

                  name="email"

                  type="email"

                  autoComplete="email"

                  required

                  value={formData.email}

                  onChange={handleChange}

                  className={`appearance-none relative block w-full pl-10 pr-3 py-3 border rounded-xl placeholder-gray-400 text-gray-900 focus:outline-none focus:ring-2 focus:ring-indigo-500 focus:border-transparent focus:z-10 sm:text-sm transition-all duration-200 ${

                    errors.email ? 'border-red-300 bg-red-50' : 'border-gray-200 bg-white hover:border-gray-300'

                  }`}

                  placeholder="Enter your email"

                />

              </div>

              {errors.email && (

                <p className="mt-2 text-sm text-red-600 flex items-center animate-fade-in-up">

                  <svg className="w-4 h-4 mr-1" fill="currentColor" viewBox="0 0 20 20">

                    <path fillRule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7 4a1 1 0 11-2 0 1 1 0 012 0zm-1-9a1 1 0 00-1 1v4a1 1 0 102 0V6a1 1 0 00-1-1z" clipRule="evenodd" />

                  </svg>

                  {errors.email}

                </p>

              )}

            </div>

            <div>

              <label htmlFor="password" className="block text-sm font-semibold text-gray-700 mb-2">

                Password

              </label>

              <div className="relative">

                <div className="absolute inset-y-0 left-0 pl-3 flex items-center pointer-events-none">

                  <svg className="h-5 w-5 text-gray-400" fill="none" viewBox="0 0 24 24" stroke="currentColor">

                    <path strokeLinecap="round" strokeLinejoin="round" strokeWidth={2} d="M12 15v2m-6 4h12a2 2 0 002-2v-6a2 2 0 00-2-2H6a2 2 0 00-2 2v6a2 2 0 002 2zm10-10V7a4 4 0 00-8 0v4h8z" />

                  </svg>

                </div>

                <input

                  id="password"

                  name="password"

                  type={showPassword ? 'text' : 'password'}

                  autoComplete="current-password"

                  required

                  value={formData.password}

                  onChange={handleChange}

                  className={`appearance-none relative block w-full pl-10 pr-12 py-3 border rounded-xl placeholder-gray-400 text-gray-900 focus:outline-none focus:ring-2 focus:ring-indigo-500 focus:border-transparent focus:z-10 sm:text-sm transition-all duration-200 ${

                    errors.password ? 'border-red-300 bg-red-50' : 'border-gray-200 bg-white hover:border-gray-300'

                  }`}

                  placeholder="Enter your password"

                />

                <button

                  type="button"

                  className="absolute inset-y-0 right-0 pr-3 flex items-center hover:bg-gray-50 rounded-r-xl transition-colors duration-200"

                  onClick={() => setShowPassword(!showPassword)}

                >

                  <svg

                    className="h-5 w-5 text-gray-400 hover:text-gray-600"

                    fill="none"

                    viewBox="0 0 24 24"

                    stroke="currentColor"

                  >

                    {showPassword ? (

                      <path

                        strokeLinecap="round"

                        strokeLinejoin="round"

                        strokeWidth={2}

                        d="M13.875 18.825A10.05 10.05 0 0112 19c-4.478 0-8.268-2.943-9.543-7a9.97 9.97 0 011.563-3.029m5.858.908a3 3 0 114.243 4.243M9.878 9.878l4.242 4.242M9.878 9.878L3 3m6.878 6.878L21 21"

                      />

                    ) : (

                      <path

                        strokeLinecap="round"

                        strokeLinejoin="round"

                        strokeWidth={2}

                        d="M15 12a3 3 0 11-6 0 3 3 0 016 0z M2.458 12C3.732 7.943 7.523 5 12 5c4.478 0 8.268 2.943 9.542 7-1.274 4.057-5.064 7-9.542 7-4.477 0-8.268-2.943-9.542-7z"

                      />

                    )}

                  </svg>

                </button>

              </div>

              {errors.password && (

                <p className="mt-2 text-sm text-red-600 flex items-center animate-fade-in-up">

                  <svg className="w-4 h-4 mr-1" fill="currentColor" viewBox="0 0 20 20">

                    <path fillRule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7 4a1 1 0 11-2 0 1 1 0 012 0zm-1-9a1 1 0 00-1 1v4a1 1 0 102 0V6a1 1 0 00-1-1z" clipRule="evenodd" />

                  </svg>

                  {errors.password}

                </p>

              )}

            </div>

            {errors.submit && (

              <div className="bg-red-50 border border-red-200 rounded-xl p-4 animate-fade-in-up">

                <div className="flex">

                  <svg className="w-5 h-5 text-red-400 mr-2" fill="currentColor" viewBox="0 0 20 20">

                    <path fillRule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7 4a1 1 0 11-2 0 1 1 0 012 0zm-1-9a1 1 0 00-1 1v4a1 1 0 102 0V6a1 1 0 00-1-1z" clipRule="evenodd" />

                  </svg>

                  <p className="text-sm text-red-600">{errors.submit}</p>

                </div>

              </div>

            )}

            <div>

              <button

                type="submit"

                disabled={isLoading}

                className="group relative w-full flex justify-center py-3 px-4 border border-transparent text-sm font-medium rounded-xl text-white bg-gradient-to-r from-indigo-600 to-purple-600 hover:from-indigo-700 hover:to-purple-700 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-indigo-500 disabled:opacity-50 disabled:cursor-not-allowed shadow-lg hover:shadow-xl transform hover:scale-105 transition-all duration-200 btn-scale"

              >

                <span className="absolute left-0 inset-y-0 flex items-center pl-3">

                  {isLoading ? (

                    <svg className="animate-spin h-5 w-5 text-white" fill="none" viewBox="0 0 24 24">

                      <circle className="opacity-25" cx="12" cy="12" r="10" stroke="currentColor" strokeWidth="4"></circle>

                      <path className="opacity-75" fill="currentColor" d="M4 12a8 8 0 018-8V0C5.373 0 0 5.373 0 12h4zm2 5.291A7.962 7.962 0 014 12H0c0 3.042 1.135 5.824 3 7.938l3-2.647z"></path>

                    </svg>

                  ) : (

                    <svg className="h-5 w-5 text-white group-hover:text-white" fill="currentColor" viewBox="0 0 20 20">

                      <path fillRule="evenodd" d="M5 9V7a5 5 0 0110 0v2a2 2 0 012 2v5a2 2 0 01-2 2H5a2 2 0 01-2-2v-5a2 2 0 012-2zm8-2v2H7V7a3 3 0 016 0z" clipRule="evenodd" />

                    </svg>

                  )}

                </span>

                {isLoading ? 'Signing in...' : 'Sign in'}

              </button>

            </div>

          </form>

          {/\* Demo Accounts \*/}

          <div className="mt-8">

            <div className="relative">

              <div className="absolute inset-0 flex items-center">

                <div className="w-full border-t border-gray-200" />

              </div>

              <div className="relative flex justify-center text-sm">

                <span className="px-4 bg-white text-gray-500 font-medium">Try Demo Accounts</span>

              </div>

            </div>

            <div className="mt-6 grid grid-cols-1 gap-3 sm:grid-cols-2 sm:gap-4">

              {demoAccounts.map((account, index) => (

                <button

                  key={index}

                  type="button"

                  onClick={() => fillDemo(account.email, account.password)}

                  className={`group relative p-4 bg-gradient-to-r ${account.color} text-white rounded-xl shadow-md hover:shadow-lg transform hover:scale-105 transition-all duration-200 text-left overflow-hidden`}

                >

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

                    <div>

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

                        <span className="text-lg mr-2">{account.icon}</span>

                        <span className="font-semibold text-sm">{account.type} Demo</span>

                      </div>

                      <p className="text-xs opacity-90 truncate">{account.email}</p>

                    </div>

                    <svg className="w-4 h-4 opacity-70 group-hover:opacity-100 transition-opacity duration-200" fill="none" viewBox="0 0 24 24" stroke="currentColor">

                      <path strokeLinecap="round" strokeLinejoin="round" strokeWidth={2} d="M13 7l5 5m0 0l-5 5m5-5H6" />

                    </svg>

                  </div>

                  <div className="absolute inset-0 bg-white/10 transform scale-x-0 group-hover:scale-x-100 transition-transform duration-300 origin-left"></div>

                </button>

              ))}

            </div>

          </div>

        </div>

        {/\* Back to home link \*/}

        <div className="text-center animate-fade-in-up animation-delay-400">

          <Link

            to="/"

            className="inline-flex items-center text-sm text-gray-500 hover:text-indigo-600 transition-colors duration-200 group"

          >

            <svg className="w-4 h-4 mr-2 group-hover:-translate-x-1 transition-transform duration-200" fill="none" viewBox="0 0 24 24" stroke="currentColor">

              <path strokeLinecap="round" strokeLinejoin="round" strokeWidth={2} d="M10 19l-7-7m0 0l7-7m-7 7h18" />

            </svg>

            Back to home

          </Link>

        </div>

      </div>

    </div>

  );

};

export default Login;

**3.ProtectedRotue.tsx**

import React from 'react';

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

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

interface ProtectedRouteProps {

  children: React.ReactNode;

  requireAdmin?: boolean;

}

const ProtectedRoute: React.FC<ProtectedRouteProps> = ({ children, requireAdmin = false }) => {

  const { isAuthenticated, user, isLoading } = useAuth();

  const location = useLocation();

  // Show loading spinner while checking authentication

  if (isLoading) {

    return (

      <div className="min-h-screen flex items-center justify-center">

        <div className="animate-spin rounded-full h-12 w-12 border-b-2 border-indigo-600"></div>

      </div>

    );

  }

  // Redirect to login if not authenticated

  if (!isAuthenticated) {

    return <Navigate to="/login" state={{ from: location }} replace />;

  }

  // Check admin requirement

  if (requireAdmin && user?.userType !== 'admin') {

    return (

      <div className="min-h-screen flex items-center justify-center bg-gray-100">

        <div className="text-center">

          <div className="bg-white p-8 rounded-lg shadow-lg">

            <svg

              className="mx-auto h-12 w-12 text-red-500"

              fill="none"

              viewBox="0 0 24 24"

              stroke="currentColor"

            >

              <path

                strokeLinecap="round"

                strokeLinejoin="round"

                strokeWidth={2}

                d="M12 9v2m0 4h.01m-6.938 4h13.856c1.54 0 2.502-1.667 1.732-2.5L13.732 4c-.77-.833-1.964-.833-2.732 0L3.732 16.5c-.77.833.192 2.5 1.732 2.5z"

              />

            </svg>

            <h2 className="mt-4 text-xl font-bold text-gray-900">Access Denied</h2>

            <p className="mt-2 text-gray-600">You don't have permission to access this page.</p>

            <p className="text-sm text-gray-500">Administrator privileges required.</p>

          </div>

        </div>

      </div>

    );

  }

  return <>{children}</>;

};

export default ProtectedRoute;

**4.SignUp.tsx**

import React, { useState } from 'react';

import { Link, useNavigate } from 'react-router-dom';

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

import type { SignUpFormData, FormErrors } from '../types/auth';

import { validateSignUpForm } from '../utils/validation';

const SignUp: React.FC = () => {

  const navigate = useNavigate();

  const { signUp } = useAuth();

  const [formData, setFormData] = useState<SignUpFormData>({

    name: '',

    email: '',

    password: '',

    confirmPassword: '',

    userType: 'user',

  });

  const [errors, setErrors] = useState<FormErrors>({});

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

  const [showPassword, setShowPassword] = useState(false);

  const [showConfirmPassword, setShowConfirmPassword] = useState(false);

  const handleChange = (e: React.ChangeEvent<HTMLInputElement | HTMLSelectElement>) => {

    const { name, value } = e.target;

    setFormData(prev => ({ ...prev, [name]: value }));

    // Clear error when user starts typing

    if (errors[name]) {

      setErrors(prev => ({ ...prev, [name]: '' }));

    }

  };

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

    e.preventDefault();

    // Validate form

    const validationErrors = validateSignUpForm(formData);

    if (Object.keys(validationErrors).length > 0) {

      setErrors(validationErrors);

      return;

    }

    setIsLoading(true);

    setErrors({});

    try {

      const result = await signUp(formData);

      if (result.success) {

        alert('Account created successfully! Please login.');

        navigate('/login');

      } else {

        setErrors({ submit: result.error || 'Registration failed' });

      }

    } catch (error) {

      setErrors({ submit: 'An unexpected error occurred' });

    } finally {

      setIsLoading(false);

    }

  };

  return (

    <div className="min-h-screen flex items-center justify-center bg-gradient-to-br from-green-50 to-emerald-100 px-4 sm:px-6 lg:px-8">

      <div className="max-w-md w-full space-y-8">

        <div>

          <h2 className="mt-6 text-center text-3xl font-extrabold text-gray-900">

            Create your account

          </h2>

          <p className="mt-2 text-center text-sm text-gray-600">

            Or{' '}

            <Link

              to="/login"

              className="font-medium text-emerald-600 hover:text-emerald-500 transition-colors duration-200"

            >

              sign in to your existing account

            </Link>

          </p>

        </div>

        <div className="bg-white py-8 px-4 shadow-xl rounded-lg sm:px-10">

          <form className="space-y-6" onSubmit={handleSubmit}>

            <div>

              <label htmlFor="name" className="block text-sm font-medium text-gray-700">

                Full Name

              </label>

              <div className="mt-1">

                <input

                  id="name"

                  name="name"

                  type="text"

                  autoComplete="name"

                  required

                  value={formData.name}

                  onChange={handleChange}

                  className={`appearance-none block w-full px-3 py-2 border rounded-md shadow-sm placeholder-gray-400 focus:outline-none focus:ring-emerald-500 focus:border-emerald-500 sm:text-sm transition-colors duration-200 ${

                    errors.name ? 'border-red-300' : 'border-gray-300'

                  }`}

                  placeholder="Enter your full name"

                />

                {errors.name && (

                  <p className="mt-2 text-sm text-red-600">{errors.name}</p>

                )}

              </div>

            </div>

            <div>

              <label htmlFor="email" className="block text-sm font-medium text-gray-700">

                Email address

              </label>

              <div className="mt-1">

                <input

                  id="email"

                  name="email"

                  type="email"

                  autoComplete="email"

                  required

                  value={formData.email}

                  onChange={handleChange}

                  className={`appearance-none block w-full px-3 py-2 border rounded-md shadow-sm placeholder-gray-400 focus:outline-none focus:ring-emerald-500 focus:border-emerald-500 sm:text-sm transition-colors duration-200 ${

                    errors.email ? 'border-red-300' : 'border-gray-300'

                  }`}

                  placeholder="Enter your email"

                />

                {errors.email && (

                  <p className="mt-2 text-sm text-red-600">{errors.email}</p>

                )}

              </div>

            </div>

            <div>

              <label htmlFor="userType" className="block text-sm font-medium text-gray-700">

                User Type

              </label>

              <div className="mt-1">

                <select

                  id="userType"

                  name="userType"

                  value={formData.userType}

                  onChange={handleChange}

                  className={`appearance-none block w-full px-3 py-2 border rounded-md shadow-sm focus:outline-none focus:ring-emerald-500 focus:border-emerald-500 sm:text-sm transition-colors duration-200 ${

                    errors.userType ? 'border-red-300' : 'border-gray-300'

                  }`}

                >

                  <option value="user">Regular User</option>

                  <option value="admin">Administrator</option>

                </select>

                {errors.userType && (

                  <p className="mt-2 text-sm text-red-600">{errors.userType}</p>

                )}

              </div>

            </div>

            <div>

              <label htmlFor="password" className="block text-sm font-medium text-gray-700">

                Password

              </label>

              <div className="mt-1 relative">

                <input

                  id="password"

                  name="password"

                  type={showPassword ? 'text' : 'password'}

                  autoComplete="new-password"

                  required

                  value={formData.password}

                  onChange={handleChange}

                  className={`appearance-none block w-full px-3 py-2 pr-10 border rounded-md shadow-sm placeholder-gray-400 focus:outline-none focus:ring-emerald-500 focus:border-emerald-500 sm:text-sm transition-colors duration-200 ${

                    errors.password ? 'border-red-300' : 'border-gray-300'

                  }`}

                  placeholder="Enter your password"

                />

                <button

                  type="button"

                  className="absolute inset-y-0 right-0 pr-3 flex items-center"

                  onClick={() => setShowPassword(!showPassword)}

                >

                  <svg

                    className="h-5 w-5 text-gray-400 hover:text-gray-600"

                    fill="none"

                    viewBox="0 0 24 24"

                    stroke="currentColor"

                  >

                    {showPassword ? (

                      <path

                        strokeLinecap="round"

                        strokeLinejoin="round"

                        strokeWidth={2}

                        d="M13.875 18.825A10.05 10.05 0 0112 19c-4.478 0-8.268-2.943-9.543-7a9.97 9.97 0 011.563-3.029m5.858.908a3 3 0 114.243 4.243M9.878 9.878l4.242 4.242M9.878 9.878L3 3m6.878 6.878L21 21"

                      />

                    ) : (

                      <path

                        strokeLinecap="round"

                        strokeLinejoin="round"

                        strokeWidth={2}

                        d="M15 12a3 3 0 11-6 0 3 3 0 016 0z M2.458 12C3.732 7.943 7.523 5 12 5c4.478 0 8.268 2.943 9.542 7-1.274 4.057-5.064 7-9.542 7-4.477 0-8.268-2.943-9.542-7z"

                      />

                    )}

                  </svg>

                </button>

                {errors.password && (

                  <p className="mt-2 text-sm text-red-600">{errors.password}</p>

                )}

              </div>

            </div>

            <div>

              <label htmlFor="confirmPassword" className="block text-sm font-medium text-gray-700">

                Confirm Password

              </label>

              <div className="mt-1 relative">

                <input

                  id="confirmPassword"

                  name="confirmPassword"

                  type={showConfirmPassword ? 'text' : 'password'}

                  autoComplete="new-password"

                  required

                  value={formData.confirmPassword}

                  onChange={handleChange}

                  className={`appearance-none block w-full px-3 py-2 pr-10 border rounded-md shadow-sm placeholder-gray-400 focus:outline-none focus:ring-emerald-500 focus:border-emerald-500 sm:text-sm transition-colors duration-200 ${

                    errors.confirmPassword ? 'border-red-300' : 'border-gray-300'

                  }`}

                  placeholder="Confirm your password"

                />

                <button

                  type="button"

                  className="absolute inset-y-0 right-0 pr-3 flex items-center"

                  onClick={() => setShowConfirmPassword(!showConfirmPassword)}

                >

                  <svg

                    className="h-5 w-5 text-gray-400 hover:text-gray-600"

                    fill="none"

                    viewBox="0 0 24 24"

                    stroke="currentColor"

                  >

                    {showConfirmPassword ? (

                      <path

                        strokeLinecap="round"

                        strokeLinejoin="round"

                        strokeWidth={2}

                        d="M13.875 18.825A10.05 10.05 0 0112 19c-4.478 0-8.268-2.943-9.543-7a9.97 9.97 0 011.563-3.029m5.858.908a3 3 0 114.243 4.243M9.878 9.878l4.242 4.242M9.878 9.878L3 3m6.878 6.878L21 21"

                      />

                    ) : (

                      <path

                        strokeLinecap="round"

                        strokeLinejoin="round"

                        strokeWidth={2}

                        d="M15 12a3 3 0 11-6 0 3 3 0 016 0z M2.458 12C3.732 7.943 7.523 5 12 5c4.478 0 8.268 2.943 9.542 7-1.274 4.057-5.064 7-9.542 7-4.477 0-8.268-2.943-9.542-7z"

                      />

                    )}

                  </svg>

                </button>

                {errors.confirmPassword && (

                  <p className="mt-2 text-sm text-red-600">{errors.confirmPassword}</p>

                )}

              </div>

            </div>

            {errors.submit && (

              <div className="bg-red-50 border border-red-200 rounded-md p-4">

                <p className="text-sm text-red-600">{errors.submit}</p>

              </div>

            )}

            <div>

              <button

                type="submit"

                disabled={isLoading}

                className="group relative w-full flex justify-center py-2 px-4 border border-transparent text-sm font-medium rounded-md text-white bg-emerald-600 hover:bg-emerald-700 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-emerald-500 disabled:opacity-50 disabled:cursor-not-allowed transition-colors duration-200"

              >

                {isLoading ? (

                  <>

                    <svg className="animate-spin -ml-1 mr-3 h-5 w-5 text-white" fill="none" viewBox="0 0 24 24">

                      <circle className="opacity-25" cx="12" cy="12" r="10" stroke="currentColor" strokeWidth="4"></circle>

                      <path className="opacity-75" fill="currentColor" d="M4 12a8 8 0 018-8V0C5.373 0 0 5.373 0 12h4zm2 5.291A7.962 7.962 0 014 12H0c0 3.042 1.135 5.824 3 7.938l3-2.647z"></path>

                    </svg>

                    Creating account...

                  </>

                ) : (

                  'Create account'

                )}

              </button>

            </div>

          </form>

          <div className="mt-6">

            <div className="relative">

              <div className="absolute inset-0 flex items-center">

                <div className="w-full border-t border-gray-300" />

              </div>

              <div className="relative flex justify-center text-sm">

                <span className="px-2 bg-white text-gray-500">Password Requirements</span>

              </div>

            </div>

            <div className="mt-4">

              <ul className="text-xs text-gray-600 space-y-1">

                <li>• At least 8 characters long</li>

                <li>• Contains uppercase and lowercase letters</li>

                <li>• Contains at least one number</li>

              </ul>

            </div>

          </div>

        </div>

      </div>

    </div>

  );

};

export default SignUp;

**5.AuthContext.tsx**

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

import type { User, AuthState, LoginFormData, SignUpFormData } from '../types/auth';

import { authStorage } from '../utils/authStorage';

interface AuthContextType extends AuthState {

  login: (formData: LoginFormData) => Promise<{ success: boolean; error?: string }>;

  signUp: (formData: SignUpFormData) => Promise<{ success: boolean; error?: string }>;

  logout: () => void;

}

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

// Mock users for demonstration

const mockUsers: User[] = [

  {

    id: '1',

    email: 'admin@example.com',

    name: 'Admin User',

    userType: 'admin',

    createdAt: '2023-01-01T00:00:00Z'

  },

  {

    id: '2',

    email: 'user@example.com',

    name: 'Regular User',

    userType: 'user',

    createdAt: '2023-01-15T00:00:00Z'

  },

  {

    id: '3',

    email: 'john@example.com',

    name: 'John Doe',

    userType: 'user',

    createdAt: '2023-02-01T00:00:00Z'

  }

];

// Mock passwords (in real app, this would be handled securely on backend)

const mockPasswords: Record<string, string> = {

  'admin@example.com': 'Admin123!',

  'user@example.com': 'User123!',

  'john@example.com': 'John123!'

};

export const AuthProvider: React.FC<{ children: ReactNode }> = ({ children }) => {

  const [authState, setAuthState] = useState<AuthState>({

    user: null,

    isAuthenticated: false,

    isLoading: true

  });

  // Check for stored authentication on mount

  useEffect(() => {

    const storedUser = authStorage.getUser();

    const storedSession = authStorage.getSession();

    if (storedUser && storedSession) {

      setAuthState({

        user: storedUser,

        isAuthenticated: true,

        isLoading: false

      });

    } else {

      setAuthState(prev => ({ ...prev, isLoading: false }));

    }

  }, []);

  const login = async (formData: LoginFormData): Promise<{ success: boolean; error?: string }> => {

    try {

      // Simulate API delay

      await new Promise(resolve => setTimeout(resolve, 1000));

      // Find user by email

      const user = mockUsers.find(u => u.email.toLowerCase() === formData.email.toLowerCase());

      if (!user) {

        return { success: false, error: 'User not found' };

      }

      // Check password

      const expectedPassword = mockPasswords[user.email];

      if (formData.password !== expectedPassword) {

        return { success: false, error: 'Invalid password' };

      }

      // Store auth data

      authStorage.saveUser(user);

      authStorage.saveSession('mock-token-' + user.id);

      // Update state

      setAuthState({

        user,

        isAuthenticated: true,

        isLoading: false

      });

      return { success: true };

    } catch (error) {

      console.error('Login error:', error);

      return { success: false, error: 'Login failed. Please try again.' };

    }

  };

  const signUp = async (formData: SignUpFormData): Promise<{ success: boolean; error?: string }> => {

    try {

      // Simulate API delay

      await new Promise(resolve => setTimeout(resolve, 1000));

      // Check if user already exists

      const existingUser = mockUsers.find(u => u.email.toLowerCase() === formData.email.toLowerCase());

      if (existingUser) {

        return { success: false, error: 'User already exists with this email' };

      }

      // Create new user

      const newUser: User = {

        id: Date.now().toString(),

        email: formData.email,

        name: formData.name,

        userType: formData.userType,

        createdAt: new Date().toISOString()

      };

      // Add to mock users (in real app, this would be sent to backend)

      mockUsers.push(newUser);

      mockPasswords[newUser.email] = formData.password;

      // Store auth data

      authStorage.saveUser(newUser);

      authStorage.saveSession('mock-token-' + newUser.id);

      // Update state

      setAuthState({

        user: newUser,

        isAuthenticated: true,

        isLoading: false

      });

      return { success: true };

    } catch (error) {

      console.error('SignUp error:', error);

      return { success: false, error: 'Sign up failed. Please try again.' };

    }

  };

  const logout = () => {

    authStorage.clearAll();

    setAuthState({

      user: null,

      isAuthenticated: false,

      isLoading: false

    });

  };

  const contextValue: AuthContextType = {

    ...authState,

    login,

    signUp,

    logout

  };

  return (

    <AuthContext.Provider value={contextValue}>

      {children}

    </AuthContext.Provider>

  );

};

export const useAuth = (): AuthContextType => {

  const context = useContext(AuthContext);

  if (context === undefined) {

    throw new Error('useAuth must be used within an AuthProvider');

  }

  return context;

};

export default AuthContext;

**6.Analytics.tsx**

import React from 'react';

import ProtectedRoute from '../components/ProtectedRoute';

const Analytics: React.FC = () => {

  const chartData = [

    { month: 'Jan', users: 12, projects: 8, tasks: 45 },

    { month: 'Feb', users: 19, projects: 12, tasks: 67 },

    { month: 'Mar', users: 25, projects: 15, tasks: 89 },

    { month: 'Apr', users: 32, projects: 18, tasks: 102 },

    { month: 'May', users: 38, projects: 22, tasks: 134 },

    { month: 'Jun', users: 42, projects: 25, tasks: 156 },

  ];

  return (

    <ProtectedRoute requireAdmin={true}>

      <div className="p-6">

        <div className="mb-6">

          <h2 className="text-2xl font-bold text-gray-900">Analytics Dashboard</h2>

          <p className="text-gray-600">System performance and user analytics</p>

        </div>

        {/\* Key Metrics \*/}

        <div className="grid grid-cols-1 gap-5 sm:grid-cols-2 lg:grid-cols-4 mb-8">

          <div className="bg-gradient-to-br from-blue-500 to-blue-600 overflow-hidden shadow rounded-lg text-white">

            <div className="p-5">

              <div className="flex items-center">

                <div className="flex-shrink-0 text-3xl opacity-80">📈</div>

                <div className="ml-5 w-0 flex-1">

                  <dl>

                    <dt className="text-sm font-medium text-blue-100 truncate">Total Revenue</dt>

                    <dd className="text-2xl font-bold">$24,567</dd>

                    <dd className="text-xs text-blue-200">+12.5% from last month</dd>

                  </dl>

                </div>

              </div>

            </div>

          </div>

          <div className="bg-gradient-to-br from-green-500 to-green-600 overflow-hidden shadow rounded-lg text-white">

            <div className="p-5">

              <div className="flex items-center">

                <div className="flex-shrink-0 text-3xl opacity-80">👤</div>

                <div className="ml-5 w-0 flex-1">

                  <dl>

                    <dt className="text-sm font-medium text-green-100 truncate">Active Users</dt>

                    <dd className="text-2xl font-bold">1,234</dd>

                    <dd className="text-xs text-green-200">+8.2% from last week</dd>

                  </dl>

                </div>

              </div>

            </div>

          </div>

          <div className="bg-gradient-to-br from-purple-500 to-purple-600 overflow-hidden shadow rounded-lg text-white">

            <div className="p-5">

              <div className="flex items-center">

                <div className="flex-shrink-0 text-3xl opacity-80">📊</div>

                <div className="ml-5 w-0 flex-1">

                  <dl>

                    <dt className="text-sm font-medium text-purple-100 truncate">Conversion Rate</dt>

                    <dd className="text-2xl font-bold">3.24%</dd>

                    <dd className="text-xs text-purple-200">+0.5% from last month</dd>

                  </dl>

                </div>

              </div>

            </div>

          </div>

          <div className="bg-gradient-to-br from-orange-500 to-orange-600 overflow-hidden shadow rounded-lg text-white">

            <div className="p-5">

              <div className="flex items-center">

                <div className="flex-shrink-0 text-3xl opacity-80">⚡</div>

                <div className="ml-5 w-0 flex-1">

                  <dl>

                    <dt className="text-sm font-medium text-orange-100 truncate">System Uptime</dt>

                    <dd className="text-2xl font-bold">99.9%</dd>

                    <dd className="text-xs text-orange-200">Last 30 days</dd>

                  </dl>

                </div>

              </div>

            </div>

          </div>

        </div>

        <div className="grid grid-cols-1 lg:grid-cols-2 gap-6">

          {/\* User Growth Chart \*/}

          <div className="bg-white shadow rounded-lg p-6">

            <h3 className="text-lg font-medium text-gray-900 mb-4">User Growth</h3>

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

              {chartData.map((item) => (

                <div key={item.month} className="flex items-center">

                  <div className="w-12 text-sm text-gray-600 font-medium">{item.month}</div>

                  <div className="flex-1 ml-4">

                    <div className="bg-gray-200 rounded-full h-4">

                      <div

                        className="bg-gradient-to-r from-blue-500 to-blue-600 h-4 rounded-full transition-all duration-700"

                        style={{ width: `${(item.users / 50) \* 100}%` }}

                      ></div>

                    </div>

                  </div>

                  <div className="w-12 ml-4 text-sm text-gray-900 font-semibold text-right">

                    {item.users}

                  </div>

                </div>

              ))}

            </div>

          </div>

          {/\* Project Statistics \*/}

          <div className="bg-white shadow rounded-lg p-6">

            <h3 className="text-lg font-medium text-gray-900 mb-4">Project Statistics</h3>

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

              {chartData.map((item) => (

                <div key={item.month} className="flex items-center">

                  <div className="w-12 text-sm text-gray-600 font-medium">{item.month}</div>

                  <div className="flex-1 ml-4">

                    <div className="bg-gray-200 rounded-full h-4">

                      <div

                        className="bg-gradient-to-r from-green-500 to-green-600 h-4 rounded-full transition-all duration-700"

                        style={{ width: `${(item.projects / 30) \* 100}%` }}

                      ></div>

                    </div>

                  </div>

                  <div className="w-12 ml-4 text-sm text-gray-900 font-semibold text-right">

                    {item.projects}

                  </div>

                </div>

              ))}

            </div>

          </div>

        </div>

        {/\* Recent Activities \*/}

        <div className="mt-6 bg-white shadow rounded-lg">

          <div className="px-6 py-4 border-b border-gray-200">

            <h3 className="text-lg font-medium text-gray-900">System Activities</h3>

          </div>

          <div className="px-6 py-4">

            <div className="flow-root">

              <ul className="-mb-8">

                <li className="relative pb-8">

                  <div className="absolute top-5 left-5 -ml-px h-full w-0.5 bg-gray-200"></div>

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

                    <div className="relative">

                      <div className="h-10 w-10 rounded-full bg-green-500 flex items-center justify-center">

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

                      </div>

                    </div>

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

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

                        <span className="font-medium text-gray-900">System Backup</span> completed successfully

                      </div>

                      <div className="mt-1 text-xs text-gray-400">2 hours ago</div>

                    </div>

                  </div>

                </li>

                <li className="relative pb-8">

                  <div className="absolute top-5 left-5 -ml-px h-full w-0.5 bg-gray-200"></div>

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

                    <div className="relative">

                      <div className="h-10 w-10 rounded-full bg-blue-500 flex items-center justify-center">

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

                      </div>

                    </div>

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

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

                        <span className="font-medium text-gray-900">New user registration</span> - john@example.com

                      </div>

                      <div className="mt-1 text-xs text-gray-400">4 hours ago</div>

                    </div>

                  </div>

                </li>

                <li className="relative pb-8">

                  <div className="absolute top-5 left-5 -ml-px h-full w-0.5 bg-gray-200"></div>

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

                    <div className="relative">

                      <div className="h-10 w-10 rounded-full bg-yellow-500 flex items-center justify-center">

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

                      </div>

                    </div>

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

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

                        <span className="font-medium text-gray-900">High CPU usage</span> detected on server-01

                      </div>

                      <div className="mt-1 text-xs text-gray-400">6 hours ago</div>

                    </div>

                  </div>

                </li>

                <li className="relative">

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

                    <div className="relative">

                      <div className="h-10 w-10 rounded-full bg-purple-500 flex items-center justify-center">

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

                      </div>

                    </div>

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

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

                        <span className="font-medium text-gray-900">Database maintenance</span> scheduled for tonight

                      </div>

                      <div className="mt-1 text-xs text-gray-400">8 hours ago</div>

                    </div>

                  </div>

                </li>

              </ul>

            </div>

          </div>

        </div>

        {/\* Performance Metrics \*/}

        <div className="mt-6 grid grid-cols-1 lg:grid-cols-3 gap-6">

          <div className="bg-white shadow rounded-lg p-6">

            <h4 className="text-lg font-medium text-gray-900 mb-4">Server Performance</h4>

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

              <div>

                <div className="flex justify-between text-sm">

                  <span className="text-gray-600">CPU Usage</span>

                  <span className="font-semibold text-gray-900">23%</span>

                </div>

                <div className="mt-2 bg-gray-200 rounded-full h-2">

                  <div className="bg-green-500 h-2 rounded-full" style={{ width: '23%' }}></div>

                </div>

              </div>

              <div>

                <div className="flex justify-between text-sm">

                  <span className="text-gray-600">Memory Usage</span>

                  <span className="font-semibold text-gray-900">67%</span>

                </div>

                <div className="mt-2 bg-gray-200 rounded-full h-2">

                  <div className="bg-yellow-500 h-2 rounded-full" style={{ width: '67%' }}></div>

                </div>

              </div>

              <div>

                <div className="flex justify-between text-sm">

                  <span className="text-gray-600">Disk Usage</span>

                  <span className="font-semibold text-gray-900">45%</span>

                </div>

                <div className="mt-2 bg-gray-200 rounded-full h-2">

                  <div className="bg-blue-500 h-2 rounded-full" style={{ width: '45%' }}></div>

                </div>

              </div>

            </div>

          </div>

          <div className="bg-white shadow rounded-lg p-6">

            <h4 className="text-lg font-medium text-gray-900 mb-4">Traffic Sources</h4>

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

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

                <span className="text-sm text-gray-600">Direct</span>

                <div className="flex items-center">

                  <div className="w-16 bg-gray-200 rounded-full h-2 mr-3">

                    <div className="bg-blue-500 h-2 rounded-full" style={{ width: '60%' }}></div>

                  </div>

                  <span className="text-sm font-semibold text-gray-900">60%</span>

                </div>

              </div>

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

                <span className="text-sm text-gray-600">Social Media</span>

                <div className="flex items-center">

                  <div className="w-16 bg-gray-200 rounded-full h-2 mr-3">

                    <div className="bg-green-500 h-2 rounded-full" style={{ width: '25%' }}></div>

                  </div>

                  <span className="text-sm font-semibold text-gray-900">25%</span>

                </div>

              </div>

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

                <span className="text-sm text-gray-600">Search Engines</span>

                <div className="flex items-center">

                  <div className="w-16 bg-gray-200 rounded-full h-2 mr-3">

                    <div className="bg-purple-500 h-2 rounded-full" style={{ width: '15%' }}></div>

                  </div>

                  <span className="text-sm font-semibold text-gray-900">15%</span>

                </div>

              </div>

            </div>

          </div>

          <div className="bg-white shadow rounded-lg p-6">

            <h4 className="text-lg font-medium text-gray-900 mb-4">Top Pages</h4>

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

              <div className="flex justify-between">

                <span className="text-sm text-gray-600 truncate">/dashboard</span>

                <span className="text-sm font-semibold text-gray-900">2,341</span>

              </div>

              <div className="flex justify-between">

                <span className="text-sm text-gray-600 truncate">/profile</span>

                <span className="text-sm font-semibold text-gray-900">1,892</span>

              </div>

              <div className="flex justify-between">

                <span className="text-sm text-gray-600 truncate">/settings</span>

                <span className="text-sm font-semibold text-gray-900">1,234</span>

              </div>

              <div className="flex justify-between">

                <span className="text-sm text-gray-600 truncate">/users</span>

                <span className="text-sm font-semibold text-gray-900">987</span>

              </div>

            </div>

          </div>

        </div>

      </div>

    </ProtectedRoute>

  );

};

export default Analytics;

**7.Dashboard.tsx**

import React, { useState, useEffect } from 'react';

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

const Dashboard: React.FC = () => {

  const { user } = useAuth();

  const [animatedStats, setAnimatedStats] = useState({

    projects: 0,

    tasks: 0,

    completed: 0,

    teamMembers: 0

  });

  const finalStats = {

    projects: 12,

    tasks: 24,

    completed: 48,

    teamMembers: user?.userType === 'admin' ? 8 : 3

  };

  // Animate stats on component mount

  useEffect(() => {

    const duration = 2000;

    const steps = 60;

    const stepTime = duration / steps;

    let currentStep = 0;

    const interval = setInterval(() => {

      currentStep++;

      const progress = currentStep / steps;

      const easeOut = 1 - Math.pow(1 - progress, 3);

      setAnimatedStats({

        projects: Math.round(finalStats.projects \* easeOut),

        tasks: Math.round(finalStats.tasks \* easeOut),

        completed: Math.round(finalStats.completed \* easeOut),

        teamMembers: Math.round(finalStats.teamMembers \* easeOut)

      });

      if (currentStep >= steps) {

        clearInterval(interval);

        setAnimatedStats(finalStats);

      }

    }, stepTime);

    return () => clearInterval(interval);

  }, [finalStats.projects, finalStats.tasks, finalStats.completed, finalStats.teamMembers]);

  const stats = [

    {

      name: 'Total Projects',

      value: animatedStats.projects,

      change: '+4.75%',

      changeType: 'positive' as const,

      icon: '📁',

      color: 'from-blue-500 to-blue-600',

      bgColor: 'bg-blue-50',

      iconBg: 'bg-blue-500'

    },

    {

      name: 'Active Tasks',

      value: animatedStats.tasks,

      change: '+54.02%',

      changeType: 'positive' as const,

      icon: '✅',

      color: 'from-green-500 to-green-600',

      bgColor: 'bg-green-50',

      iconBg: 'bg-green-500'

    },

    {

      name: 'Completed',

      value: animatedStats.completed,

      change: '+12.39%',

      changeType: 'positive' as const,

      icon: '🎯',

      color: 'from-purple-500 to-purple-600',

      bgColor: 'bg-purple-50',

      iconBg: 'bg-purple-500'

    },

    {

      name: 'Team Members',

      value: animatedStats.teamMembers,

      change: '+10.18%',

      changeType: 'positive' as const,

      icon: '👥',

      color: 'from-orange-500 to-orange-600',

      bgColor: 'bg-orange-50',

      iconBg: 'bg-orange-500'

    },

  ];

  const recentActivities = [

    {

      id: 1,

      action: 'Completed project review',

      time: '2 hours ago',

      type: 'success',

      icon: '✓',

      user: 'John Doe'

    },

    {

      id: 2,

      action: 'Updated user profile',

      time: '4 hours ago',

      type: 'info',

      icon: '👤',

      user: 'Jane Smith'

    },

    {

      id: 3,

      action: 'Created new task',

      time: '6 hours ago',

      type: 'warning',

      icon: '📝',

      user: 'Mike Johnson'

    },

    {

      id: 4,

      action: 'Joined team meeting',

      time: '1 day ago',

      type: 'info',

      icon: '📞',

      user: 'Sarah Wilson'

    },

    {

      id: 5,

      action: 'Submitted weekly report',

      time: '2 days ago',

      type: 'success',

      icon: '📊',

      user: 'Alex Brown'

    },

  ];

  const quickActions = [

    {

      title: 'New Project',

      description: 'Create a new project',

      icon: '➕',

      color: 'from-blue-500 to-blue-600',

      hoverColor: 'hover:from-blue-600 hover:to-blue-700'

    },

    {

      title: 'Add Task',

      description: 'Create a new task',

      icon: '📋',

      color: 'from-green-500 to-green-600',

      hoverColor: 'hover:from-green-600 hover:to-green-700'

    },

    {

      title: 'View Reports',

      description: 'Check analytics',

      icon: '📊',

      color: 'from-purple-500 to-purple-600',

      hoverColor: 'hover:from-purple-600 hover:to-purple-700'

    },

    {

      title: 'Settings',

      description: 'Manage preferences',

      icon: '⚙️',

      color: 'from-gray-500 to-gray-600',

      hoverColor: 'hover:from-gray-600 hover:to-gray-700'

    }

  ];

  return (

    <div className="p-4 sm:p-6 bg-gray-50 min-h-full">

      {/\* Welcome Section \*/}

      <div className="mb-8 animate-fade-in-up">

        <div className="bg-gradient-to-r from-indigo-600 via-purple-600 to-pink-600 rounded-2xl shadow-xl p-6 sm:p-8 text-white relative overflow-hidden">

          <div className="absolute inset-0 bg-gradient-to-r from-indigo-600/90 to-purple-600/90"></div>

          <div className="absolute top-0 right-0 -mt-4 -mr-4 w-24 h-24 bg-white/10 rounded-full blur-2xl"></div>

          <div className="absolute bottom-0 left-0 -mb-4 -ml-4 w-32 h-32 bg-white/10 rounded-full blur-2xl"></div>

          <div className="relative z-10">

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

              <div>

                <h1 className="text-2xl sm:text-3xl font-bold mb-2">

                  Welcome back, {user?.name}! 👋

                </h1>

                <p className="text-indigo-100 text-lg">

                  {user?.userType === 'admin' ? 'Administrator Dashboard' : 'User Dashboard'}

                </p>

                <p className="text-indigo-200 text-sm mt-1">

                  Here's what's happening with your projects today.

                </p>

              </div>

              <div className="mt-4 sm:mt-0">

                <div className="flex items-center space-x-2 bg-white/20 rounded-lg px-4 py-2 backdrop-blur-sm">

                  <div className="w-3 h-3 bg-green-400 rounded-full animate-pulse"></div>

                  <span className="text-sm font-medium">All systems operational</span>

                </div>

              </div>

            </div>

          </div>

        </div>

      </div>

      {/\* Stats Grid \*/}

      <div className="grid grid-cols-1 gap-4 sm:gap-6 sm:grid-cols-2 lg:grid-cols-4 mb-8">

        {stats.map((item, index) => (

          <div

            key={item.name}

            className={`relative bg-white p-6 shadow-lg rounded-2xl overflow-hidden border border-gray-100 hover:shadow-xl transition-all duration-300 transform hover:scale-105 animate-fade-in-up`}

            style={{ animationDelay: `${index \* 100}ms` }}

          >

            <div className={`absolute inset-0 bg-gradient-to-br ${item.color} opacity-5`}></div>

            <div className="relative">

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

                <div className="flex items-center">

                  <div className={`flex-shrink-0 w-12 h-12 ${item.iconBg} rounded-xl flex items-center justify-center shadow-lg`}>

                    <span className="text-2xl text-white">{item.icon}</span>

                  </div>

                </div>

                <div className={`flex items-baseline text-sm font-semibold ${

                  item.changeType === 'positive' ? 'text-green-600' : 'text-red-600'

                }`}>

                  {item.changeType === 'positive' ? (

                    <svg className="self-center flex-shrink-0 h-4 w-4 text-green-500 mr-1" fill="currentColor" viewBox="0 0 20 20">

                      <path fillRule="evenodd" d="M5.293 9.707a1 1 0 010-1.414l4-4a1 1 0 011.414 0l4 4a1 1 0 01-1.414 1.414L11 7.414V15a1 1 0 11-2 0V7.414L6.707 9.707a1 1 0 01-1.414 0z" clipRule="evenodd" />

                    </svg>

                  ) : (

                    <svg className="self-center flex-shrink-0 h-4 w-4 text-red-500 mr-1" fill="currentColor" viewBox="0 0 20 20">

                      <path fillRule="evenodd" d="M14.707 10.293a1 1 0 010 1.414l-4 4a1 1 0 01-1.414 0l-4-4a1 1 0 111.414-1.414L9 12.586V5a1 1 0 012 0v7.586l2.293-2.293a1 1 0 011.414 0z" clipRule="evenodd" />

                    </svg>

                  )}

                  {item.change}

                </div>

              </div>

              <div className="mt-4">

                <div className="text-3xl font-bold text-gray-900 mb-1">{item.value}</div>

                <div className="text-sm font-medium text-gray-500">{item.name}</div>

              </div>

            </div>

          </div>

        ))}

      </div>

      <div className="grid grid-cols-1 lg:grid-cols-2 gap-6 mb-8">

        {/\* Enhanced Quick Actions \*/}

        <div className="bg-white shadow-lg rounded-2xl overflow-hidden animate-fade-in-up animation-delay-400">

          <div className="p-6 border-b border-gray-100">

            <h3 className="text-xl font-bold text-gray-900 flex items-center">

              <span className="w-8 h-8 bg-gradient-to-r from-indigo-500 to-purple-500 rounded-lg flex items-center justify-center mr-3">

                <span className="text-white text-sm">⚡</span>

              </span>

              Quick Actions

            </h3>

            <p className="text-gray-500 text-sm mt-1">Frequently used actions for faster workflow</p>

          </div>

          <div className="p-6">

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

              {quickActions.map((action, index) => (

                <button

                  key={index}

                  className={`group relative p-4 border-2 border-dashed border-gray-200 rounded-xl hover:border-transparent transition-all duration-300 text-center overflow-hidden bg-gradient-to-br ${action.color} ${action.hoverColor} hover:text-white hover:shadow-lg hover:scale-105`}

                >

                  <div className="absolute inset-0 bg-white group-hover:bg-transparent transition-colors duration-300"></div>

                  <div className="relative z-10">

                    <div className="text-3xl mb-2 group-hover:scale-110 transition-transform duration-300">

                      {action.icon}

                    </div>

                    <div className="text-sm font-semibold text-gray-900 group-hover:text-white transition-colors duration-300">

                      {action.title}

                    </div>

                    <div className="text-xs text-gray-500 group-hover:text-white/80 transition-colors duration-300 mt-1">

                      {action.description}

                    </div>

                  </div>

                </button>

              ))}

            </div>

          </div>

        </div>

        {/\* Enhanced Recent Activity \*/}

        <div className="bg-white shadow-lg rounded-2xl overflow-hidden animate-fade-in-up animation-delay-600">

          <div className="p-6 border-b border-gray-100">

            <h3 className="text-xl font-bold text-gray-900 flex items-center">

              <span className="w-8 h-8 bg-gradient-to-r from-green-500 to-emerald-500 rounded-lg flex items-center justify-center mr-3">

                <span className="text-white text-sm">📈</span>

              </span>

              Recent Activity

            </h3>

            <p className="text-gray-500 text-sm mt-1">Latest updates from your team</p>

          </div>

          <div className="p-6">

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

              {recentActivities.map((activity) => (

                <div key={activity.id} className="flex items-start space-x-4 p-3 rounded-xl hover:bg-gray-50 transition-colors duration-200 group">

                  <div className={`flex-shrink-0 w-10 h-10 rounded-xl flex items-center justify-center shadow-sm ${

                    activity.type === 'success' ? 'bg-green-100 text-green-600' :

                    activity.type === 'warning' ? 'bg-yellow-100 text-yellow-600' :

                    'bg-blue-100 text-blue-600'

                  } group-hover:scale-110 transition-transform duration-200`}>

                    <span className="text-sm font-semibold">{activity.icon}</span>

                  </div>

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

                    <p className="text-sm font-medium text-gray-900 group-hover:text-indigo-600 transition-colors duration-200">

                      {activity.action}

                    </p>

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

                      <p className="text-xs text-gray-500">{activity.user}</p>

                      <p className="text-xs text-gray-400">{activity.time}</p>

                    </div>

                  </div>

                </div>

              ))}

            </div>

          </div>

        </div>

      </div>

      {/\* Enhanced Progress Section \*/}

      <div className="bg-white shadow-lg rounded-2xl overflow-hidden animate-fade-in-up animation-delay-800">

        <div className="p-6 border-b border-gray-100">

          <h3 className="text-xl font-bold text-gray-900 flex items-center">

            <span className="w-8 h-8 bg-gradient-to-r from-purple-500 to-pink-500 rounded-lg flex items-center justify-center mr-3">

              <span className="text-white text-sm">📊</span>

            </span>

            Progress Overview

          </h3>

        </div>

        <div className="p-6">

          <div className="grid grid-cols-1 sm:grid-cols-3 gap-6">

            {[

              { label: 'Projects Completed', value: 85, color: 'bg-green-500' },

              { label: 'Tasks in Progress', value: 60, color: 'bg-blue-500' },

              { label: 'Goals Achieved', value: 92, color: 'bg-purple-500' }

            ].map((item, index) => (

              <div key={index} className="text-center">

                <div className="relative w-24 h-24 mx-auto mb-4">

                  <svg className="w-24 h-24 transform -rotate-90" viewBox="0 0 36 36">

                    <circle

                      cx="18"

                      cy="18"

                      r="16"

                      fill="none"

                      stroke="#e5e7eb"

                      strokeWidth="3"

                    />

                    <circle

                      cx="18"

                      cy="18"

                      r="16"

                      fill="none"

                      strokeWidth="3"

                      strokeDasharray={`${item.value}, 100`}

                      className={`${item.color.replace('bg-', 'stroke-')} transition-all duration-1000`}

                      strokeLinecap="round"

                    />

                  </svg>

                  <div className="absolute inset-0 flex items-center justify-center">

                    <span className="text-xl font-bold text-gray-700">{item.value}%</span>

                  </div>

                </div>

                <p className="text-sm font-medium text-gray-600">{item.label}</p>

              </div>

            ))}

          </div>

        </div>

      </div>

    </div>

  );

};

export default Dashboard;

**8.Home.tsx**

import React, { useState } from 'react';

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

const Home: React.FC = () => {

  const [isMenuOpen, setIsMenuOpen] = useState(false);

  const [activeFeature, setActiveFeature] = useState(0);

  const features = [

    {

      icon: '🔒',

      title: 'Secure Authentication',

      description: 'Industry-standard security with encrypted passwords and session management.',

      color: 'from-blue-500 to-blue-600'

    },

    {

      icon: '👥',

      title: 'Role-Based Access',

      description: 'Different user types with appropriate permissions and access levels.',

      color: 'from-green-500 to-green-600'

    },

    {

      icon: '📱',

      title: 'Mobile Responsive',

      description: 'Fully responsive design that works seamlessly on all devices.',

      color: 'from-purple-500 to-purple-600'

    },

    {

      icon: '⚡',

      title: 'Fast & Reliable',

      description: 'Built with modern technologies for optimal performance and reliability.',

      color: 'from-orange-500 to-orange-600'

    }

  ];

  return (

    <div className="min-h-screen bg-gradient-to-br from-slate-50 via-blue-50 to-indigo-100">

      {/\* Header \*/}

      <header className="bg-white/80 backdrop-blur-lg shadow-lg sticky top-0 z-50 border-b border-white/20">

        <div className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">

          <div className="flex justify-between items-center py-4">

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

              <div className="w-10 h-10 bg-gradient-to-r from-indigo-600 to-purple-600 rounded-xl flex items-center justify-center shadow-lg">

                <span className="text-white font-bold text-lg">A</span>

              </div>

              <h1 className="text-2xl font-bold bg-gradient-to-r from-indigo-600 to-purple-600 bg-clip-text text-transparent">

                AuthApp

              </h1>

            </div>

            {/\* Desktop Navigation \*/}

            <nav className="hidden md:flex items-center space-x-1">

              <Link

                to="/login"

                className="relative px-4 py-2 text-gray-700 hover:text-indigo-600 rounded-lg transition-all duration-300 hover:bg-indigo-50 group"

              >

                Sign In

                <span className="absolute inset-x-0 bottom-0 h-0.5 bg-indigo-600 transform scale-x-0 group-hover:scale-x-100 transition-transform duration-300"></span>

              </Link>

              <Link

                to="/signup"

                className="relative px-6 py-2.5 bg-gradient-to-r from-indigo-600 to-purple-600 text-white rounded-lg shadow-lg hover:shadow-xl transform hover:scale-105 transition-all duration-300 font-medium overflow-hidden group"

              >

                <span className="relative z-10">Get Started</span>

                <div className="absolute inset-0 bg-gradient-to-r from-indigo-700 to-purple-700 transform scale-x-0 group-hover:scale-x-100 transition-transform duration-300 origin-left"></div>

              </Link>

            </nav>

            {/\* Mobile menu button \*/}

            <button

              onClick={() => setIsMenuOpen(!isMenuOpen)}

              className="md:hidden p-2 rounded-lg text-gray-600 hover:text-indigo-600 hover:bg-indigo-50 transition-colors duration-300"

            >

              <svg

                className={`w-6 h-6 transform transition-transform duration-300 ${isMenuOpen ? 'rotate-90' : ''}`}

                fill="none"

                stroke="currentColor"

                viewBox="0 0 24 24"

              >

                {isMenuOpen ? (

                  <path strokeLinecap="round" strokeLinejoin="round" strokeWidth={2} d="M6 18L18 6M6 6l12 12" />

                ) : (

                  <path strokeLinecap="round" strokeLinejoin="round" strokeWidth={2} d="M4 6h16M4 12h16M4 18h16" />

                )}

              </svg>

            </button>

          </div>

          {/\* Mobile Navigation \*/}

          <div className={`md:hidden transition-all duration-300 ease-in-out ${isMenuOpen ? 'max-h-48 pb-4' : 'max-h-0'} overflow-hidden`}>

            <nav className="flex flex-col space-y-2">

              <Link

                to="/login"

                className="px-4 py-2 text-gray-700 hover:text-indigo-600 rounded-lg transition-colors duration-300 hover:bg-indigo-50"

                onClick={() => setIsMenuOpen(false)}

              >

                Sign In

              </Link>

              <Link

                to="/signup"

                className="px-4 py-2 bg-gradient-to-r from-indigo-600 to-purple-600 text-white rounded-lg shadow-lg hover:shadow-xl transform hover:scale-105 transition-all duration-300 font-medium text-center"

                onClick={() => setIsMenuOpen(false)}

              >

                Get Started

              </Link>

            </nav>

          </div>

        </div>

      </header>

      {/\* Hero Section \*/}

      <main>

        <div className="relative overflow-hidden">

          <div className="max-w-7xl mx-auto">

            <div className="relative z-10 pb-8 sm:pb-16 md:pb-20 lg:max-w-2xl lg:w-full lg:pb-28 xl:pb-32">

              <main className="mt-10 mx-auto max-w-7xl px-4 sm:mt-12 sm:px-6 md:mt-16 lg:mt-20 lg:px-8 xl:mt-28">

                <div className="text-center lg:text-left">

                  <div className="animate-fade-in-up">

                    <h1 className="text-4xl tracking-tight font-extrabold text-gray-900 sm:text-5xl md:text-6xl lg:text-5xl xl:text-6xl">

                      <span className="block">Secure Authentication</span>

                      <span className="block bg-gradient-to-r from-indigo-600 to-purple-600 bg-clip-text text-transparent">

                        Made Simple

                      </span>

                    </h1>

                    <p className="mt-3 text-base text-gray-600 sm:mt-5 sm:text-lg sm:max-w-xl sm:mx-auto md:mt-5 md:text-xl lg:mx-0 lg:max-w-lg">

                      Experience seamless user authentication with role-based access control.

                      Built with React, TypeScript, and Tailwind CSS for modern web applications.

                    </p>

                  </div>

                  <div className="mt-8 sm:mt-10 animate-fade-in-up animation-delay-200">

                    <div className="sm:flex sm:justify-center lg:justify-start gap-4">

                      <Link

                        to="/signup"

                        className="group relative w-full sm:w-auto flex items-center justify-center px-8 py-4 border border-transparent text-base font-medium rounded-xl text-white bg-gradient-to-r from-indigo-600 to-purple-600 hover:from-indigo-700 hover:to-purple-700 md:py-4 md:text-lg md:px-10 shadow-lg hover:shadow-xl transform hover:scale-105 transition-all duration-300 overflow-hidden"

                      >

                        <span className="relative z-10 flex items-center">

                          Create Account

                          <svg className="ml-2 w-5 h-5 group-hover:translate-x-1 transition-transform duration-300" fill="none" stroke="currentColor" viewBox="0 0 24 24">

                            <path strokeLinecap="round" strokeLinejoin="round" strokeWidth={2} d="M13 7l5 5m0 0l-5 5m5-5H6" />

                          </svg>

                        </span>

                        <div className="absolute inset-0 bg-gradient-to-r from-indigo-700 to-purple-700 transform scale-x-0 group-hover:scale-x-100 transition-transform duration-300 origin-left"></div>

                      </Link>

                      <Link

                        to="/login"

                        className="mt-3 sm:mt-0 w-full sm:w-auto flex items-center justify-center px-8 py-4 border-2 border-indigo-200 text-base font-medium rounded-xl text-indigo-700 bg-white hover:bg-indigo-50 hover:border-indigo-300 md:py-4 md:text-lg md:px-10 shadow-md hover:shadow-lg transform hover:scale-105 transition-all duration-300"

                      >

                        Sign In

                      </Link>

                    </div>

                  </div>

                  {/\* Trust indicators \*/}

                  <div className="mt-12 animate-fade-in-up animation-delay-400">

                    <p className="text-sm text-gray-500 mb-4">Trusted by developers worldwide</p>

                    <div className="flex items-center justify-center lg:justify-start space-x-6 opacity-60">

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

                        <div className="w-8 h-8 bg-gradient-to-r from-blue-500 to-blue-600 rounded-lg flex items-center justify-center">

                          <span className="text-white text-xs font-bold">R</span>

                        </div>

                        <span className="text-sm font-medium text-gray-600">React</span>

                      </div>

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

                        <div className="w-8 h-8 bg-gradient-to-r from-blue-600 to-blue-700 rounded-lg flex items-center justify-center">

                          <span className="text-white text-xs font-bold">TS</span>

                        </div>

                        <span className="text-sm font-medium text-gray-600">TypeScript</span>

                      </div>

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

                        <div className="w-8 h-8 bg-gradient-to-r from-cyan-500 to-blue-500 rounded-lg flex items-center justify-center">

                          <span className="text-white text-xs font-bold">TW</span>

                        </div>

                        <span className="text-sm font-medium text-gray-600">Tailwind</span>

                      </div>

                    </div>

                  </div>

                </div>

              </main>

            </div>

          </div>

          {/\* Hero Image \*/}

          <div className="lg:absolute lg:inset-y-0 lg:right-0 lg:w-1/2">

            <div className="h-56 w-full bg-gradient-to-br from-indigo-500 via-purple-600 to-pink-600 sm:h-72 md:h-96 lg:w-full lg:h-full flex items-center justify-center relative overflow-hidden">

              {/\* Animated background elements \*/}

              <div className="absolute inset-0">

                <div className="absolute top-1/4 left-1/4 w-32 h-32 bg-white/10 rounded-full blur-xl animate-pulse"></div>

                <div className="absolute top-3/4 right-1/4 w-24 h-24 bg-white/10 rounded-full blur-lg animate-pulse animation-delay-1000"></div>

                <div className="absolute top-1/2 right-1/3 w-16 h-16 bg-white/10 rounded-full blur-md animate-pulse animation-delay-2000"></div>

              </div>

              <div className="text-center text-white relative z-10 animate-float">

                <div className="text-6xl sm:text-7xl md:text-8xl mb-6 filter drop-shadow-2xl">🔐</div>

                <h3 className="text-xl sm:text-2xl md:text-3xl font-bold mb-3 text-shadow-lg">Secure & Reliable</h3>

                <p className="text-base sm:text-lg md:text-xl opacity-90 max-w-md mx-auto px-4 text-shadow">

                  Your data is protected with industry-standard security

                </p>

              </div>

            </div>

          </div>

        </div>

        {/\* Interactive Features Section \*/}

        <div className="py-16 sm:py-20 bg-white">

          <div className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">

            <div className="lg:text-center">

              <h2 className="text-base text-indigo-600 font-semibold tracking-wide uppercase animate-fade-in-up">Features</h2>

              <p className="mt-2 text-3xl leading-8 font-extrabold tracking-tight text-gray-900 sm:text-4xl animate-fade-in-up animation-delay-200">

                Everything you need for authentication

              </p>

              <p className="mt-4 max-w-2xl text-xl text-gray-500 lg:mx-auto animate-fade-in-up animation-delay-400">

                Our authentication system provides all the essential features for secure user management.

              </p>

            </div>

            <div className="mt-16">

              <div className="grid grid-cols-1 gap-8 sm:grid-cols-2 lg:grid-cols-4">

                {features.map((feature, index) => (

                  <div

                    key={index}

                    className={`group relative bg-white rounded-2xl shadow-lg hover:shadow-2xl transition-all duration-500 transform hover:scale-105 cursor-pointer animate-fade-in-up border border-gray-100 overflow-hidden ${

                      activeFeature === index ? 'ring-2 ring-indigo-500 ring-offset-2' : ''

                    }`}

                    style={{ animationDelay: `${index \* 100}ms` }}

                    onClick={() => setActiveFeature(index)}

                    onMouseEnter={() => setActiveFeature(index)}

                  >

                    <div className="p-8 text-center">

                      <div className={`w-16 h-16 mx-auto mb-4 bg-gradient-to-r ${feature.color} rounded-2xl flex items-center justify-center shadow-lg group-hover:shadow-xl transition-shadow duration-300`}>

                        <span className="text-2xl">{feature.icon}</span>

                      </div>

                      <h3 className="text-lg font-semibold text-gray-900 mb-3 group-hover:text-indigo-600 transition-colors duration-300">

                        {feature.title}

                      </h3>

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

                        {feature.description}

                      </p>

                    </div>

                    <div className={`absolute inset-x-0 bottom-0 h-1 bg-gradient-to-r ${feature.color} transform scale-x-0 group-hover:scale-x-100 transition-transform duration-300`}></div>

                  </div>

                ))}

              </div>

            </div>

            {/\* Interactive Feature Display \*/}

            <div className="mt-16 bg-gradient-to-r from-gray-50 to-gray-100 rounded-3xl p-8 sm:p-12">

              <div className="text-center">

                <div className={`inline-flex items-center justify-center w-20 h-20 bg-gradient-to-r ${features[activeFeature].color} rounded-2xl mb-6 shadow-lg`}>

                  <span className="text-3xl">{features[activeFeature].icon}</span>

                </div>

                <h3 className="text-2xl font-bold text-gray-900 mb-4">{features[activeFeature].title}</h3>

                <p className="text-lg text-gray-600 max-w-2xl mx-auto leading-relaxed">

                  {features[activeFeature].description}

                </p>

              </div>

            </div>

          </div>

        </div>

        {/\* Enhanced Demo Section \*/}

        <div className="bg-gradient-to-br from-indigo-50 via-blue-50 to-purple-50 relative overflow-hidden">

          <div className="absolute inset-0">

            <div className="absolute top-0 left-1/4 w-72 h-72 bg-gradient-to-r from-indigo-400/20 to-purple-400/20 rounded-full blur-3xl"></div>

            <div className="absolute bottom-0 right-1/4 w-96 h-96 bg-gradient-to-r from-purple-400/20 to-pink-400/20 rounded-full blur-3xl"></div>

          </div>

          <div className="relative max-w-7xl mx-auto py-16 sm:py-20 px-4 sm:px-6 lg:px-8">

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

              <h2 className="text-3xl sm:text-4xl font-extrabold tracking-tight text-gray-900 mb-4">

                <span className="block">Ready to get started?</span>

                <span className="block bg-gradient-to-r from-indigo-600 to-purple-600 bg-clip-text text-transparent">

                  Try our demo accounts

                </span>

              </h2>

              <p className="text-lg text-gray-600 max-w-2xl mx-auto">

                Experience the full functionality with our pre-configured demo accounts

              </p>

            </div>

            <div className="grid grid-cols-1 md:grid-cols-2 gap-8 max-w-4xl mx-auto">

              {/\* Admin Demo Card \*/}

              <div className="group relative bg-white rounded-2xl shadow-xl hover:shadow-2xl transition-all duration-500 transform hover:scale-105 border border-purple-100 overflow-hidden">

                <div className="absolute inset-0 bg-gradient-to-br from-purple-500/5 to-purple-600/5"></div>

                <div className="relative p-8">

                  <div className="flex items-center mb-6">

                    <div className="w-14 h-14 bg-gradient-to-r from-purple-500 to-purple-600 rounded-2xl flex items-center justify-center shadow-lg">

                      <span className="text-2xl">👑</span>

                    </div>

                    <div className="ml-4">

                      <h3 className="text-xl font-bold text-gray-900 group-hover:text-purple-600 transition-colors duration-300">

                        Admin Demo

                      </h3>

                      <p className="text-sm text-purple-600 font-medium">Full Access</p>

                    </div>

                  </div>

                  <div className="space-y-3 mb-6">

                    <div className="flex items-center justify-between p-3 bg-gray-50 rounded-lg">

                      <span className="text-sm font-medium text-gray-600">Email:</span>

                      <code className="text-sm font-mono text-gray-900 bg-white px-2 py-1 rounded border">

                        admin@example.com

                      </code>

                    </div>

                    <div className="flex items-center justify-between p-3 bg-gray-50 rounded-lg">

                      <span className="text-sm font-medium text-gray-600">Password:</span>

                      <code className="text-sm font-mono text-gray-900 bg-white px-2 py-1 rounded border">

                        Admin123!

                      </code>

                    </div>

                  </div>

                  <div className="mb-6">

                    <h4 className="text-sm font-semibold text-gray-700 mb-3">Admin Features:</h4>

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

                      {['User Management', 'Analytics Dashboard', 'System Settings', 'Full Permissions'].map((feature, idx) => (

                        <li key={idx} className="flex items-center text-sm text-gray-600">

                          <svg className="w-4 h-4 text-green-500 mr-2" fill="currentColor" viewBox="0 0 20 20">

                            <path fillRule="evenodd" d="M16.707 5.293a1 1 0 010 1.414l-8 8a1 1 0 01-1.414 0l-4-4a1 1 0 011.414-1.414L8 12.586l7.293-7.293a1 1 0 011.414 0z" clipRule="evenodd" />

                          </svg>

                          {feature}

                        </li>

                      ))}

                    </ul>

                  </div>

                  <Link

                    to="/login"

                    className="block w-full text-center py-3 px-4 bg-gradient-to-r from-purple-500 to-purple-600 text-white font-medium rounded-xl shadow-lg hover:shadow-xl transform hover:scale-105 transition-all duration-300"

                  >

                    Try Admin Demo

                  </Link>

                </div>

              </div>

              {/\* User Demo Card \*/}

              <div className="group relative bg-white rounded-2xl shadow-xl hover:shadow-2xl transition-all duration-500 transform hover:scale-105 border border-blue-100 overflow-hidden">

                <div className="absolute inset-0 bg-gradient-to-br from-blue-500/5 to-blue-600/5"></div>

                <div className="relative p-8">

                  <div className="flex items-center mb-6">

                    <div className="w-14 h-14 bg-gradient-to-r from-blue-500 to-blue-600 rounded-2xl flex items-center justify-center shadow-lg">

                      <span className="text-2xl">�</span>

                    </div>

                    <div className="ml-4">

                      <h3 className="text-xl font-bold text-gray-900 group-hover:text-blue-600 transition-colors duration-300">

                        User Demo

                      </h3>

                      <p className="text-sm text-blue-600 font-medium">Standard Access</p>

                    </div>

                  </div>

                  <div className="space-y-3 mb-6">

                    <div className="flex items-center justify-between p-3 bg-gray-50 rounded-lg">

                      <span className="text-sm font-medium text-gray-600">Email:</span>

                      <code className="text-sm font-mono text-gray-900 bg-white px-2 py-1 rounded border">

                        user@example.com

                      </code>

                    </div>

                    <div className="flex items-center justify-between p-3 bg-gray-50 rounded-lg">

                      <span className="text-sm font-medium text-gray-600">Password:</span>

                      <code className="text-sm font-mono text-gray-900 bg-white px-2 py-1 rounded border">

                        User123!

                      </code>

                    </div>

                  </div>

                  <div className="mb-6">

                    <h4 className="text-sm font-semibold text-gray-700 mb-3">User Features:</h4>

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

                      {['Personal Dashboard', 'Profile Management', 'Settings Access', 'Project Tools'].map((feature, idx) => (

                        <li key={idx} className="flex items-center text-sm text-gray-600">

                          <svg className="w-4 h-4 text-green-500 mr-2" fill="currentColor" viewBox="0 0 20 20">

                            <path fillRule="evenodd" d="M16.707 5.293a1 1 0 010 1.414l-8 8a1 1 0 01-1.414 0l-4-4a1 1 0 011.414-1.414L8 12.586l7.293-7.293a1 1 0 011.414 0z" clipRule="evenodd" />

                          </svg>

                          {feature}

                        </li>

                      ))}

                    </ul>

                  </div>

                  <Link

                    to="/login"

                    className="block w-full text-center py-3 px-4 bg-gradient-to-r from-blue-500 to-blue-600 text-white font-medium rounded-xl shadow-lg hover:shadow-xl transform hover:scale-105 transition-all duration-300"

                  >

                    Try User Demo

                  </Link>

                </div>

              </div>

            </div>

            {/\* Quick Start Guide \*/}

            <div className="mt-16 text-center">

              <div className="inline-flex items-center space-x-4 bg-white/80 backdrop-blur-lg rounded-full px-6 py-3 shadow-lg border border-white/20">

                <span className="text-sm font-medium text-gray-600">Quick Start:</span>

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

                  <span className="w-6 h-6 bg-gradient-to-r from-indigo-500 to-purple-500 rounded-full flex items-center justify-center text-white text-xs font-bold">1</span>

                  <span className="text-sm text-gray-700">Choose account</span>

                </div>

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

                  <span className="w-6 h-6 bg-gradient-to-r from-indigo-500 to-purple-500 rounded-full flex items-center justify-center text-white text-xs font-bold">2</span>

                  <span className="text-sm text-gray-700">Login</span>

                </div>

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

                  <span className="w-6 h-6 bg-gradient-to-r from-indigo-500 to-purple-500 rounded-full flex items-center justify-center text-white text-xs font-bold">3</span>

                  <span className="text-sm text-gray-700">Explore features</span>

                </div>

              </div>

            </div>

          </div>

        </div>

      </main>

      {/\* Footer \*/}

      <footer className="bg-white">

        <div className="max-w-7xl mx-auto py-12 px-4 sm:px-6 md:flex md:items-center md:justify-between lg:px-8">

          <div className="flex justify-center space-x-6 md:order-2">

            <a href="#" className="text-gray-400 hover:text-gray-500">

              <span className="sr-only">GitHub</span>

              <svg className="h-6 w-6" fill="currentColor" viewBox="0 0 24 24">

                <path d="M12 2C6.477 2 2 6.484 2 12.017c0 4.425 2.865 8.18 6.839 9.504.5.092.682-.217.682-.483 0-.237-.008-.868-.013-1.703-2.782.605-3.369-1.343-3.369-1.343-.454-1.158-1.11-1.466-1.11-1.466-.908-.62.069-.608.069-.608 1.003.07 1.531 1.032 1.531 1.032.892 1.53 2.341 1.088 2.91.832.092-.647.35-1.088.636-1.338-2.22-.253-4.555-1.113-4.555-4.951 0-1.093.39-1.988 1.029-2.688-.103-.253-.446-1.272.098-2.65 0 0 .84-.27 2.75 1.026A9.564 9.564 0 0112 6.844c.85.004 1.705.115 2.504.337 1.909-1.296 2.747-1.027 2.747-1.027.546 1.379.202 2.398.1 2.651.64.7 1.028 1.595 1.028 2.688 0 3.848-2.339 4.695-4.566 4.943.359.309.678.92.678 1.855 0 1.338-.012 2.419-.012 2.747 0 .268.18.58.688.482A10.019 10.019 0 0022 12.017C22 6.484 17.522 2 12 2z"/>

              </svg>

            </a>

          </div>

          <div className="mt-8 md:mt-0 md:order-1">

            <p className="text-center text-base text-gray-400">

              &copy; 2025 AuthApp. Built with React & TypeScript.

            </p>

          </div>

        </div>

      </footer>

    </div>

  );

};

export default Home;

**9.Profile\_new.tsx**

import React, { useState } from 'react';

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

const Profile: React.FC = () => {

  const { user } = useAuth();

  const [isEditing, setIsEditing] = useState(false);

  const [formData, setFormData] = useState({

    name: user?.name || '',

    email: user?.email || '',

    phone: '+1 (555) 123-4567',

    location: 'San Francisco, CA',

    bio: 'Passionate developer with expertise in full-stack development, UI/UX design, and project management.',

    company: 'Tech Solutions Inc.',

    role: user?.userType === 'admin' ? 'System Administrator' : 'Software Developer',

    joinDate: 'January 2023',

    projects: user?.userType === 'admin' ? '25' : '12',

    experience: user?.userType === 'admin' ? '8 years' : '4 years'

  });

  const handleInputChange = (e: React.ChangeEvent<HTMLInputElement | HTMLTextAreaElement>) => {

    setFormData({

      ...formData,

      [e.target.name]: e.target.value

    });

  };

  const handleSave = () => {

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

    setIsEditing(false);

    // Show success message

  };

  const skills = [

    { name: 'React', level: 90, color: 'bg-blue-500' },

    { name: 'TypeScript', level: 85, color: 'bg-indigo-500' },

    { name: 'Node.js', level: 80, color: 'bg-green-500' },

    { name: 'Python', level: 75, color: 'bg-yellow-500' },

    { name: 'Database Design', level: 70, color: 'bg-purple-500' },

    { name: 'Project Management', level: user?.userType === 'admin' ? 95 : 65, color: 'bg-pink-500' }

  ];

  const achievements = [

    {

      title: 'Project Excellence Award',

      description: 'Recognized for outstanding project delivery',

      date: '2024',

      icon: '🏆',

      color: 'from-yellow-400 to-yellow-500'

    },

    {

      title: 'Team Leadership',

      description: 'Successfully led cross-functional team',

      date: '2023',

      icon: '👑',

      color: 'from-purple-400 to-purple-500'

    },

    {

      title: 'Innovation Champion',

      description: 'Introduced new development practices',

      date: '2023',

      icon: '💡',

      color: 'from-blue-400 to-blue-500'

    },

    {

      title: 'Customer Success',

      description: 'Achieved 98% client satisfaction rate',

      date: '2024',

      icon: '⭐',

      color: 'from-green-400 to-green-500'

    }

  ];

  const activityData = [

    { month: 'Jan', commits: 45, tasks: 12 },

    { month: 'Feb', commits: 52, tasks: 15 },

    { month: 'Mar', commits: 38, tasks: 9 },

    { month: 'Apr', commits: 61, tasks: 18 },

    { month: 'May', commits: 49, tasks: 14 },

    { month: 'Jun', commits: 67, tasks: 21 }

  ];

  return (

    <div className="p-4 sm:p-6 bg-gray-50 min-h-full">

      {/\* Profile Header \*/}

      <div className="bg-white shadow-xl rounded-3xl overflow-hidden mb-8 animate-fade-in-up">

        <div className="bg-gradient-to-r from-indigo-600 via-purple-600 to-pink-600 h-32 sm:h-40"></div>

        <div className="relative px-6 sm:px-8 pb-8">

          <div className="sm:flex sm:items-end sm:space-x-8">

            <div className="relative -mt-16 sm:-mt-20">

              <div className="w-24 h-24 sm:w-32 sm:h-32 bg-white rounded-full shadow-xl border-4 border-white overflow-hidden">

                <div className="w-full h-full bg-gradient-to-br from-indigo-500 to-purple-600 flex items-center justify-center">

                  <span className="text-2xl sm:text-4xl font-bold text-white">

                    {user?.name?.split(' ').map(n => n[0]).join('') || 'U'}

                  </span>

                </div>

              </div>

              <div className="absolute -bottom-2 -right-2 w-8 h-8 bg-green-500 rounded-full border-4 border-white flex items-center justify-center">

                <span className="text-white text-xs">✓</span>

              </div>

            </div>

            <div className="mt-4 sm:mt-0 sm:flex-1">

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

                <div>

                  <h1 className="text-2xl sm:text-3xl font-bold text-gray-900">{formData.name}</h1>

                  <p className="text-lg text-indigo-600 font-semibold">{formData.role}</p>

                  <p className="text-gray-500 flex items-center mt-1">

                    <span className="mr-2">📍</span>

                    {formData.location}

                  </p>

                </div>

                <div className="mt-4 sm:mt-0 flex space-x-3">

                  <button

                    onClick={() => setIsEditing(!isEditing)}

                    className="bg-indigo-600 text-white px-6 py-2 rounded-xl font-semibold hover:bg-indigo-700 transform hover:scale-105 transition-all duration-200 shadow-lg"

                  >

                    {isEditing ? 'Cancel' : 'Edit Profile'}

                  </button>

                  {isEditing && (

                    <button

                      onClick={handleSave}

                      className="bg-green-600 text-white px-6 py-2 rounded-xl font-semibold hover:bg-green-700 transform hover:scale-105 transition-all duration-200 shadow-lg"

                    >

                      Save Changes

                    </button>

                  )}

                </div>

              </div>

              <div className="mt-4 grid grid-cols-3 gap-4 sm:gap-6">

                <div className="text-center">

                  <div className="text-2xl font-bold text-gray-900">{formData.projects}</div>

                  <div className="text-sm text-gray-500">Projects</div>

                </div>

                <div className="text-center">

                  <div className="text-2xl font-bold text-gray-900">{formData.experience}</div>

                  <div className="text-sm text-gray-500">Experience</div>

                </div>

                <div className="text-center">

                  <div className="text-2xl font-bold text-gray-900">4.9</div>

                  <div className="text-sm text-gray-500">Rating</div>

                </div>

              </div>

            </div>

          </div>

        </div>

      </div>

      <div className="grid grid-cols-1 lg:grid-cols-3 gap-8">

        {/\* Left Column \*/}

        <div className="lg:col-span-2 space-y-8">

          {/\* About Section \*/}

          <div className="bg-white shadow-xl rounded-2xl overflow-hidden animate-fade-in-up animation-delay-200">

            <div className="p-6 border-b border-gray-100">

              <h3 className="text-xl font-bold text-gray-900 flex items-center">

                <span className="w-8 h-8 bg-gradient-to-r from-blue-500 to-indigo-500 rounded-lg flex items-center justify-center mr-3">

                  <span className="text-white text-sm">👤</span>

                </span>

                About Me

              </h3>

            </div>

            <div className="p-6">

              {isEditing ? (

                <textarea

                  name="bio"

                  value={formData.bio}

                  onChange={handleInputChange}

                  rows={4}

                  className="w-full px-4 py-3 border border-gray-300 rounded-xl focus:ring-2 focus:ring-indigo-500 focus:border-transparent transition-all duration-200"

                />

              ) : (

                <p className="text-gray-600 leading-relaxed">{formData.bio}</p>

              )}

            </div>

          </div>

          {/\* Skills Section \*/}

          <div className="bg-white shadow-xl rounded-2xl overflow-hidden animate-fade-in-up animation-delay-400">

            <div className="p-6 border-b border-gray-100">

              <h3 className="text-xl font-bold text-gray-900 flex items-center">

                <span className="w-8 h-8 bg-gradient-to-r from-green-500 to-emerald-500 rounded-lg flex items-center justify-center mr-3">

                  <span className="text-white text-sm">🚀</span>

                </span>

                Skills & Expertise

              </h3>

            </div>

            <div className="p-6">

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

                {skills.map((skill, index) => (

                  <div key={skill.name} className="group">

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

                      <span className="text-sm font-medium text-gray-700">{skill.name}</span>

                      <span className="text-sm text-gray-500">{skill.level}%</span>

                    </div>

                    <div className="w-full bg-gray-200 rounded-full h-3 overflow-hidden">

                      <div

                        className={`h-full ${skill.color} rounded-full transition-all duration-1000 ease-out transform`}

                        style={{

                          width: `${skill.level}%`,

                          animationDelay: `${index \* 200}ms`

                        }}

                      />

                    </div>

                  </div>

                ))}

              </div>

            </div>

          </div>

          {/\* Activity Chart \*/}

          <div className="bg-white shadow-xl rounded-2xl overflow-hidden animate-fade-in-up animation-delay-600">

            <div className="p-6 border-b border-gray-100">

              <h3 className="text-xl font-bold text-gray-900 flex items-center">

                <span className="w-8 h-8 bg-gradient-to-r from-purple-500 to-pink-500 rounded-lg flex items-center justify-center mr-3">

                  <span className="text-white text-sm">📊</span>

                </span>

                Activity Overview

              </h3>

            </div>

            <div className="p-6">

              <div className="grid grid-cols-3 sm:grid-cols-6 gap-4">

                {activityData.map((data, index) => (

                  <div key={data.month} className="text-center group">

                    <div className="relative mb-3">

                      <div className="h-24 flex items-end justify-center">

                        <div

                          className="w-8 bg-gradient-to-t from-indigo-500 to-purple-500 rounded-t-lg transition-all duration-1000 hover:from-purple-500 hover:to-pink-500 group-hover:scale-110"

                          style={{

                            height: `${(data.commits / 70) \* 100}%`,

                            animationDelay: `${index \* 100}ms`

                          }}

                        />

                      </div>

                    </div>

                    <div className="text-xs text-gray-500 font-medium">{data.month}</div>

                    <div className="text-sm font-bold text-gray-700">{data.commits}</div>

                  </div>

                ))}

              </div>

              <div className="mt-4 text-center text-sm text-gray-500">

                Monthly contributions and task completions

              </div>

            </div>

          </div>

        </div>

        {/\* Right Column \*/}

        <div className="space-y-8">

          {/\* Contact Information \*/}

          <div className="bg-white shadow-xl rounded-2xl overflow-hidden animate-fade-in-up animation-delay-800">

            <div className="p-6 border-b border-gray-100">

              <h3 className="text-xl font-bold text-gray-900 flex items-center">

                <span className="w-8 h-8 bg-gradient-to-r from-orange-500 to-red-500 rounded-lg flex items-center justify-center mr-3">

                  <span className="text-white text-sm">📞</span>

                </span>

                Contact Info

              </h3>

            </div>

            <div className="p-6 space-y-4">

              <div className="flex items-center p-3 rounded-xl bg-gray-50 hover:bg-gray-100 transition-colors duration-200">

                <span className="text-gray-400 mr-3">📧</span>

                <div>

                  <div className="text-sm text-gray-500">Email</div>

                  {isEditing ? (

                    <input

                      name="email"

                      value={formData.email}

                      onChange={handleInputChange}

                      className="text-gray-900 bg-transparent border-none outline-none"

                    />

                  ) : (

                    <div className="text-gray-900">{formData.email}</div>

                  )}

                </div>

              </div>

              <div className="flex items-center p-3 rounded-xl bg-gray-50 hover:bg-gray-100 transition-colors duration-200">

                <span className="text-gray-400 mr-3">📱</span>

                <div>

                  <div className="text-sm text-gray-500">Phone</div>

                  {isEditing ? (

                    <input

                      name="phone"

                      value={formData.phone}

                      onChange={handleInputChange}

                      className="text-gray-900 bg-transparent border-none outline-none"

                    />

                  ) : (

                    <div className="text-gray-900">{formData.phone}</div>

                  )}

                </div>

              </div>

              <div className="flex items-center p-3 rounded-xl bg-gray-50 hover:bg-gray-100 transition-colors duration-200">

                <span className="text-gray-400 mr-3">🏢</span>

                <div>

                  <div className="text-sm text-gray-500">Company</div>

                  {isEditing ? (

                    <input

                      name="company"

                      value={formData.company}

                      onChange={handleInputChange}

                      className="text-gray-900 bg-transparent border-none outline-none"

                    />

                  ) : (

                    <div className="text-gray-900">{formData.company}</div>

                  )}

                </div>

              </div>

              <div className="flex items-center p-3 rounded-xl bg-gray-50 hover:bg-gray-100 transition-colors duration-200">

                <span className="text-gray-400 mr-3">📅</span>

                <div>

                  <div className="text-sm text-gray-500">Joined</div>

                  <div className="text-gray-900">{formData.joinDate}</div>

                </div>

              </div>

            </div>

          </div>

          {/\* Achievements \*/}

          <div className="bg-white shadow-xl rounded-2xl overflow-hidden animate-fade-in-up animation-delay-1000">

            <div className="p-6 border-b border-gray-100">

              <h3 className="text-xl font-bold text-gray-900 flex items-center">

                <span className="w-8 h-8 bg-gradient-to-r from-yellow-500 to-orange-500 rounded-lg flex items-center justify-center mr-3">

                  <span className="text-white text-sm">🏆</span>

                </span>

                Achievements

              </h3>

            </div>

            <div className="p-6 space-y-4">

              {achievements.map((achievement, index) => (

                <div key={index} className="group">

                  <div className="flex items-start space-x-4 p-4 rounded-xl hover:bg-gray-50 transition-all duration-200">

                    <div className={`flex-shrink-0 w-12 h-12 bg-gradient-to-r ${achievement.color} rounded-xl flex items-center justify-center shadow-lg group-hover:scale-110 transition-transform duration-200`}>

                      <span className="text-white text-lg">{achievement.icon}</span>

                    </div>

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

                      <h4 className="text-sm font-semibold text-gray-900 group-hover:text-indigo-600 transition-colors duration-200">

                        {achievement.title}

                      </h4>

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

                        {achievement.description}

                      </p>

                      <p className="text-xs text-gray-400 mt-1 font-medium">

                        {achievement.date}

                      </p>

                    </div>

                  </div>

                </div>

              ))}

            </div>

          </div>

        </div>

      </div>

    </div>

  );

};

export default Profile;

**10.profile.tsx**

import React, { useState } from 'react';

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

const Profile: React.FC = () => {

  const { user } = useAuth();

  const [isEditing, setIsEditing] = useState(false);

  const [formData, setFormData] = useState({

    name: user?.name || '',

    email: user?.email || '',

  });

  const handleChange = (e: React.ChangeEvent<HTMLInputElement>) => {

    const { name, value } = e.target;

    setFormData(prev => ({ ...prev, [name]: value }));

  };

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

    e.preventDefault();

    // Here you would typically update the user profile

    console.log('Updating profile:', formData);

    setIsEditing(false);

    alert('Profile updated successfully!');

  };

  const handleCancel = () => {

    setFormData({

      name: user?.name || '',

      email: user?.email || '',

    });

    setIsEditing(false);

  };

  return (

    <div className="p-6">

      <div className="max-w-3xl mx-auto">

        <div className="bg-white shadow rounded-lg">

          <div className="px-6 py-4 border-b border-gray-200">

            <h3 className="text-lg font-medium text-gray-900">Profile Information</h3>

            <p className="mt-1 text-sm text-gray-500">

              Update your account information and preferences.

            </p>

          </div>

          <div className="px-6 py-6">

            <div className="flex items-center space-x-6 mb-8">

              <div className="bg-indigo-500 rounded-full h-20 w-20 flex items-center justify-center">

                <span className="text-white font-medium text-2xl">

                  {user?.name.charAt(0).toUpperCase()}

                </span>

              </div>

              <div>

                <h4 className="text-xl font-bold text-gray-900">{user?.name}</h4>

                <p className="text-gray-600">{user?.email}</p>

                <div className="mt-1">

                  <span className={`inline-flex items-center px-2.5 py-0.5 rounded-full text-xs font-medium ${

                    user?.userType === 'admin'

                      ? 'bg-purple-100 text-purple-800'

                      : 'bg-green-100 text-green-800'

                  }`}>

                    {user?.userType === 'admin' ? '👑 Administrator' : '👤 User'}

                  </span>

                </div>

              </div>

            </div>

            <form onSubmit={handleSubmit}>

              <div className="grid grid-cols-1 gap-6 sm:grid-cols-2">

                <div>

                  <label htmlFor="name" className="block text-sm font-medium text-gray-700">

                    Full Name

                  </label>

                  <div className="mt-1">

                    <input

                      type="text"

                      name="name"

                      id="name"

                      value={formData.name}

                      onChange={handleChange}

                      disabled={!isEditing}

                      className={`shadow-sm block w-full sm:text-sm border-gray-300 rounded-md ${

                        isEditing

                          ? 'focus:ring-indigo-500 focus:border-indigo-500'

                          : 'bg-gray-50 text-gray-500 cursor-not-allowed'

                      }`}

                    />

                  </div>

                </div>

                <div>

                  <label htmlFor="email" className="block text-sm font-medium text-gray-700">

                    Email Address

                  </label>

                  <div className="mt-1">

                    <input

                      type="email"

                      name="email"

                      id="email"

                      value={formData.email}

                      onChange={handleChange}

                      disabled={!isEditing}

                      className={`shadow-sm block w-full sm:text-sm border-gray-300 rounded-md ${

                        isEditing

                          ? 'focus:ring-indigo-500 focus:border-indigo-500'

                          : 'bg-gray-50 text-gray-500 cursor-not-allowed'

                      }`}

                    />

                  </div>

                </div>

                <div>

                  <label className="block text-sm font-medium text-gray-700">

                    User Type

                  </label>

                  <div className="mt-1">

                    <input

                      type="text"

                      value={user?.userType === 'admin' ? 'Administrator' : 'Regular User'}

                      disabled

                      className="shadow-sm block w-full sm:text-sm border-gray-300 rounded-md bg-gray-50 text-gray-500 cursor-not-allowed"

                    />

                  </div>

                </div>

                <div>

                  <label className="block text-sm font-medium text-gray-700">

                    Member Since

                  </label>

                  <div className="mt-1">

                    <input

                      type="text"

                      value={new Date(user?.createdAt || '').toLocaleDateString()}

                      disabled

                      className="shadow-sm block w-full sm:text-sm border-gray-300 rounded-md bg-gray-50 text-gray-500 cursor-not-allowed"

                    />

                  </div>

                </div>

              </div>

              <div className="mt-6 flex justify-end space-x-3">

                {!isEditing ? (

                  <button

                    type="button"

                    onClick={() => setIsEditing(true)}

                    className="bg-indigo-600 py-2 px-4 border border-transparent rounded-md shadow-sm text-sm font-medium text-white hover:bg-indigo-700 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-indigo-500 transition-colors duration-200"

                  >

                    Edit Profile

                  </button>

                ) : (

                  <>

                    <button

                      type="button"

                      onClick={handleCancel}

                      className="bg-white py-2 px-4 border border-gray-300 rounded-md shadow-sm text-sm font-medium text-gray-700 hover:bg-gray-50 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-indigo-500 transition-colors duration-200"

                    >

                      Cancel

                    </button>

                    <button

                      type="submit"

                      className="bg-indigo-600 py-2 px-4 border border-transparent rounded-md shadow-sm text-sm font-medium text-white hover:bg-indigo-700 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-indigo-500 transition-colors duration-200"

                    >

                      Save Changes

                    </button>

                  </>

                )}

              </div>

            </form>

          </div>

        </div>

        {/\* Activity Summary \*/}

        <div className="mt-6 bg-white shadow rounded-lg">

          <div className="px-6 py-4 border-b border-gray-200">

            <h3 className="text-lg font-medium text-gray-900">Activity Summary</h3>

          </div>

          <div className="px-6 py-6">

            <div className="grid grid-cols-1 gap-4 sm:grid-cols-2 lg:grid-cols-3">

              <div className="bg-blue-50 p-4 rounded-lg">

                <div className="flex items-center">

                  <div className="flex-shrink-0 text-2xl">📈</div>

                  <div className="ml-3">

                    <p className="text-sm font-medium text-gray-900">Projects Completed</p>

                    <p className="text-2xl font-bold text-blue-600">12</p>

                  </div>

                </div>

              </div>

              <div className="bg-green-50 p-4 rounded-lg">

                <div className="flex items-center">

                  <div className="flex-shrink-0 text-2xl">✅</div>

                  <div className="ml-3">

                    <p className="text-sm font-medium text-gray-900">Tasks Done</p>

                    <p className="text-2xl font-bold text-green-600">48</p>

                  </div>

                </div>

              </div>

              <div className="bg-purple-50 p-4 rounded-lg">

                <div className="flex items-center">

                  <div className="flex-shrink-0 text-2xl">⭐</div>

                  <div className="ml-3">

                    <p className="text-sm font-medium text-gray-900">Achievement Points</p>

                    <p className="text-2xl font-bold text-purple-600">320</p>

                  </div>

                </div>

              </div>

            </div>

          </div>

        </div>

      </div>

    </div>

  );

};

export default Profile;

**11.Setting.tsx**

import React from 'react';

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

const Settings: React.FC = () => {

  const { user } = useAuth();

  return (

    <div className="p-6">

      <div className="max-w-4xl mx-auto">

        <div className="space-y-6">

          {/\* General Settings \*/}

          <div className="bg-white shadow rounded-lg">

            <div className="px-6 py-4 border-b border-gray-200">

              <h3 className="text-lg font-medium text-gray-900">General Settings</h3>

              <p className="mt-1 text-sm text-gray-500">

                Manage your account preferences and settings.

              </p>

            </div>

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

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

                <div>

                  <h4 className="text-sm font-medium text-gray-900">Email Notifications</h4>

                  <p className="text-sm text-gray-500">Receive notifications via email</p>

                </div>

                <button

                  type="button"

                  className="bg-indigo-600 relative inline-flex flex-shrink-0 h-6 w-11 border-2 border-transparent rounded-full cursor-pointer transition-colors ease-in-out duration-200 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-indigo-500"

                  role="switch"

                  aria-checked="true"

                >

                  <span className="translate-x-5 pointer-events-none inline-block h-5 w-5 rounded-full bg-white shadow transform ring-0 transition ease-in-out duration-200"></span>

                </button>

              </div>

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

                <div>

                  <h4 className="text-sm font-medium text-gray-900">Push Notifications</h4>

                  <p className="text-sm text-gray-500">Receive push notifications in your browser</p>

                </div>

                <button

                  type="button"

                  className="bg-gray-200 relative inline-flex flex-shrink-0 h-6 w-11 border-2 border-transparent rounded-full cursor-pointer transition-colors ease-in-out duration-200 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-indigo-500"

                  role="switch"

                  aria-checked="false"

                >

                  <span className="translate-x-0 pointer-events-none inline-block h-5 w-5 rounded-full bg-white shadow transform ring-0 transition ease-in-out duration-200"></span>

                </button>

              </div>

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

                <div>

                  <h4 className="text-sm font-medium text-gray-900">Weekly Reports</h4>

                  <p className="text-sm text-gray-500">Get weekly summary reports</p>

                </div>

                <button

                  type="button"

                  className="bg-indigo-600 relative inline-flex flex-shrink-0 h-6 w-11 border-2 border-transparent rounded-full cursor-pointer transition-colors ease-in-out duration-200 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-indigo-500"

                  role="switch"

                  aria-checked="true"

                >

                  <span className="translate-x-5 pointer-events-none inline-block h-5 w-5 rounded-full bg-white shadow transform ring-0 transition ease-in-out duration-200"></span>

                </button>

              </div>

            </div>

          </div>

          {/\* Privacy Settings \*/}

          <div className="bg-white shadow rounded-lg">

            <div className="px-6 py-4 border-b border-gray-200">

              <h3 className="text-lg font-medium text-gray-900">Privacy & Security</h3>

              <p className="mt-1 text-sm text-gray-500">

                Control your privacy and security preferences.

              </p>

            </div>

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

              <div>

                <h4 className="text-sm font-medium text-gray-900 mb-3">Profile Visibility</h4>

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

                  <label className="flex items-center">

                    <input

                      type="radio"

                      name="visibility"

                      value="public"

                      className="focus:ring-indigo-500 h-4 w-4 text-indigo-600 border-gray-300"

                    />

                    <span className="ml-3 text-sm text-gray-700">

                      <span className="font-medium">Public</span> - Anyone can see your profile

                    </span>

                  </label>

                  <label className="flex items-center">

                    <input

                      type="radio"

                      name="visibility"

                      value="team"

                      defaultChecked

                      className="focus:ring-indigo-500 h-4 w-4 text-indigo-600 border-gray-300"

                    />

                    <span className="ml-3 text-sm text-gray-700">

                      <span className="font-medium">Team Only</span> - Only team members can see your profile

                    </span>

                  </label>

                  <label className="flex items-center">

                    <input

                      type="radio"

                      name="visibility"

                      value="private"

                      className="focus:ring-indigo-500 h-4 w-4 text-indigo-600 border-gray-300"

                    />

                    <span className="ml-3 text-sm text-gray-700">

                      <span className="font-medium">Private</span> - Only you can see your profile

                    </span>

                  </label>

                </div>

              </div>

              <div className="pt-6 border-t border-gray-200">

                <button className="bg-red-600 py-2 px-4 border border-transparent rounded-md shadow-sm text-sm font-medium text-white hover:bg-red-700 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-red-500 transition-colors duration-200">

                  Change Password

                </button>

              </div>

            </div>

          </div>

          {/\* Theme Settings \*/}

          <div className="bg-white shadow rounded-lg">

            <div className="px-6 py-4 border-b border-gray-200">

              <h3 className="text-lg font-medium text-gray-900">Appearance</h3>

              <p className="mt-1 text-sm text-gray-500">

                Customize how the application looks and feels.

              </p>

            </div>

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

              <div>

                <h4 className="text-sm font-medium text-gray-900 mb-3">Theme</h4>

                <div className="grid grid-cols-3 gap-3">

                  <label className="cursor-pointer">

                    <input type="radio" name="theme" value="light" defaultChecked className="sr-only" />

                    <div className="border-2 border-gray-200 rounded-lg p-4 hover:border-indigo-500 transition-colors duration-200">

                      <div className="w-full h-16 bg-white border border-gray-200 rounded-md mb-2"></div>

                      <span className="text-sm font-medium text-gray-900">Light</span>

                    </div>

                  </label>

                  <label className="cursor-pointer">

                    <input type="radio" name="theme" value="dark" className="sr-only" />

                    <div className="border-2 border-gray-200 rounded-lg p-4 hover:border-indigo-500 transition-colors duration-200">

                      <div className="w-full h-16 bg-gray-800 border border-gray-600 rounded-md mb-2"></div>

                      <span className="text-sm font-medium text-gray-900">Dark</span>

                    </div>

                  </label>

                  <label className="cursor-pointer">

                    <input type="radio" name="theme" value="auto" className="sr-only" />

                    <div className="border-2 border-gray-200 rounded-lg p-4 hover:border-indigo-500 transition-colors duration-200">

                      <div className="w-full h-16 bg-gradient-to-r from-white to-gray-800 border border-gray-300 rounded-md mb-2"></div>

                      <span className="text-sm font-medium text-gray-900">Auto</span>

                    </div>

                  </label>

                </div>

              </div>

            </div>

          </div>

          {/\* Admin Only Settings \*/}

          {user?.userType === 'admin' && (

            <div className="bg-white shadow rounded-lg border-l-4 border-purple-500">

              <div className="px-6 py-4 border-b border-gray-200">

                <h3 className="text-lg font-medium text-gray-900 flex items-center">

                  <span className="mr-2">👑</span>

                  Administrator Settings

                </h3>

                <p className="mt-1 text-sm text-gray-500">

                  Advanced settings available only to administrators.

                </p>

              </div>

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

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

                  <div>

                    <h4 className="text-sm font-medium text-gray-900">System Notifications</h4>

                    <p className="text-sm text-gray-500">Receive system alerts and maintenance notifications</p>

                  </div>

                  <button

                    type="button"

                    className="bg-indigo-600 relative inline-flex flex-shrink-0 h-6 w-11 border-2 border-transparent rounded-full cursor-pointer transition-colors ease-in-out duration-200 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-indigo-500"

                    role="switch"

                    aria-checked="true"

                  >

                    <span className="translate-x-5 pointer-events-none inline-block h-5 w-5 rounded-full bg-white shadow transform ring-0 transition ease-in-out duration-200"></span>

                  </button>

                </div>

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

                  <div>

                    <h4 className="text-sm font-medium text-gray-900">User Activity Monitoring</h4>

                    <p className="text-sm text-gray-500">Monitor user activity and generate reports</p>

                  </div>

                  <button

                    type="button"

                    className="bg-indigo-600 relative inline-flex flex-shrink-0 h-6 w-11 border-2 border-transparent rounded-full cursor-pointer transition-colors ease-in-out duration-200 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-indigo-500"

                    role="switch"

                    aria-checked="true"

                  >

                    <span className="translate-x-5 pointer-events-none inline-block h-5 w-5 rounded-full bg-white shadow transform ring-0 transition ease-in-out duration-200"></span>

                  </button>

                </div>

                <div className="pt-6 border-t border-gray-200">

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

                    <button className="bg-purple-600 py-2 px-4 border border-transparent rounded-md shadow-sm text-sm font-medium text-white hover:bg-purple-700 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-purple-500 transition-colors duration-200">

                      System Backup

                    </button>

                    <button className="bg-gray-600 py-2 px-4 border border-transparent rounded-md shadow-sm text-sm font-medium text-white hover:bg-gray-700 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-gray-500 transition-colors duration-200">

                      Export Data

                    </button>

                  </div>

                </div>

              </div>

            </div>

          )}

          {/\* Save Button \*/}

          <div className="flex justify-end">

            <button className="bg-indigo-600 py-2 px-4 border border-transparent rounded-md shadow-sm text-sm font-medium text-white hover:bg-indigo-700 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-indigo-500 transition-colors duration-200">

              Save Settings

            </button>

          </div>

        </div>

      </div>

    </div>

  );

};

export default Settings;

**12. User.tsx**

import React, { useState } from 'react';

import ProtectedRoute from '../components/ProtectedRoute';

interface User {

  id: string;

  name: string;

  email: string;

  userType: 'admin' | 'user';

  status: 'active' | 'inactive';

  lastActive: string;

  createdAt: string;

}

const Users: React.FC = () => {

  const [users, setUsers] = useState<User[]>([

    {

      id: '1',

      name: 'Admin User',

      email: 'admin@example.com',

      userType: 'admin',

      status: 'active',

      lastActive: '2 minutes ago',

      createdAt: '2024-01-15',

    },

    {

      id: '2',

      name: 'Regular User',

      email: 'user@example.com',

      userType: 'user',

      status: 'active',

      lastActive: '1 hour ago',

      createdAt: '2024-01-20',

    },

    {

      id: '3',

      name: 'John Smith',

      email: 'john@example.com',

      userType: 'user',

      status: 'active',

      lastActive: '3 hours ago',

      createdAt: '2024-02-01',

    },

    {

      id: '4',

      name: 'Jane Doe',

      email: 'jane@example.com',

      userType: 'user',

      status: 'inactive',

      lastActive: '2 days ago',

      createdAt: '2024-02-10',

    },

  ]);

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

  const [filterType, setFilterType] = useState<'all' | 'admin' | 'user'>('all');

  const [filterStatus, setFilterStatus] = useState<'all' | 'active' | 'inactive'>('all');

  const filteredUsers = users.filter(user => {

    const matchesSearch = user.name.toLowerCase().includes(searchTerm.toLowerCase()) ||

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

    const matchesType = filterType === 'all' || user.userType === filterType;

    const matchesStatus = filterStatus === 'all' || user.status === filterStatus;

    return matchesSearch && matchesType && matchesStatus;

  });

  const handleStatusToggle = (userId: string) => {

    setUsers(prevUsers =>

      prevUsers.map(user =>

        user.id === userId

          ? { ...user, status: user.status === 'active' ? 'inactive' : 'active' }

          : user

      )

    );

  };

  const handleDeleteUser = (userId: string) => {

    if (window.confirm('Are you sure you want to delete this user?')) {

      setUsers(prevUsers => prevUsers.filter(user => user.id !== userId));

    }

  };

  return (

    <ProtectedRoute requireAdmin={true}>

      <div className="p-6">

        <div className="mb-6">

          <h2 className="text-2xl font-bold text-gray-900">User Management</h2>

          <p className="text-gray-600">Manage users and their permissions</p>

        </div>

        {/\* Filters \*/}

        <div className="bg-white p-4 rounded-lg shadow mb-6">

          <div className="grid grid-cols-1 md:grid-cols-4 gap-4">

            <div>

              <label htmlFor="search" className="block text-sm font-medium text-gray-700 mb-1">

                Search Users

              </label>

              <input

                type="text"

                id="search"

                placeholder="Search by name or email..."

                value={searchTerm}

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

                className="w-full px-3 py-2 border border-gray-300 rounded-md focus:outline-none focus:ring-indigo-500 focus:border-indigo-500"

              />

            </div>

            <div>

              <label htmlFor="type-filter" className="block text-sm font-medium text-gray-700 mb-1">

                User Type

              </label>

              <select

                id="type-filter"

                value={filterType}

                onChange={(e) => setFilterType(e.target.value as 'all' | 'admin' | 'user')}

                className="w-full px-3 py-2 border border-gray-300 rounded-md focus:outline-none focus:ring-indigo-500 focus:border-indigo-500"

              >

                <option value="all">All Types</option>

                <option value="admin">Admin</option>

                <option value="user">User</option>

              </select>

            </div>

            <div>

              <label htmlFor="status-filter" className="block text-sm font-medium text-gray-700 mb-1">

                Status

              </label>

              <select

                id="status-filter"

                value={filterStatus}

                onChange={(e) => setFilterStatus(e.target.value as 'all' | 'active' | 'inactive')}

                className="w-full px-3 py-2 border border-gray-300 rounded-md focus:outline-none focus:ring-indigo-500 focus:border-indigo-500"

              >

                <option value="all">All Status</option>

                <option value="active">Active</option>

                <option value="inactive">Inactive</option>

              </select>

            </div>

            <div className="flex items-end">

              <button className="w-full bg-indigo-600 text-white px-4 py-2 rounded-md hover:bg-indigo-700 focus:outline-none focus:ring-2 focus:ring-indigo-500 transition-colors duration-200">

                Add User

              </button>

            </div>

          </div>

        </div>

        {/\* Users Table \*/}

        <div className="bg-white shadow rounded-lg overflow-hidden">

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

            <table className="min-w-full divide-y divide-gray-200">

              <thead className="bg-gray-50">

                <tr>

                  <th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">

                    User

                  </th>

                  <th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">

                    Type

                  </th>

                  <th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">

                    Status

                  </th>

                  <th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">

                    Last Active

                  </th>

                  <th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">

                    Joined

                  </th>

                  <th className="px-6 py-3 text-right text-xs font-medium text-gray-500 uppercase tracking-wider">

                    Actions

                  </th>

                </tr>

              </thead>

              <tbody className="bg-white divide-y divide-gray-200">

                {filteredUsers.map((user) => (

                  <tr key={user.id} className="hover:bg-gray-50">

                    <td className="px-6 py-4 whitespace-nowrap">

                      <div className="flex items-center">

                        <div className="bg-indigo-500 rounded-full h-10 w-10 flex items-center justify-center">

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

                            {user.name.charAt(0).toUpperCase()}

                          </span>

                        </div>

                        <div className="ml-4">

                          <div className="text-sm font-medium text-gray-900">{user.name}</div>

                          <div className="text-sm text-gray-500">{user.email}</div>

                        </div>

                      </div>

                    </td>

                    <td className="px-6 py-4 whitespace-nowrap">

                      <span className={`inline-flex items-center px-2.5 py-0.5 rounded-full text-xs font-medium ${

                        user.userType === 'admin'

                          ? 'bg-purple-100 text-purple-800'

                          : 'bg-green-100 text-green-800'

                      }`}>

                        {user.userType === 'admin' ? '👑 Admin' : '👤 User'}

                      </span>

                    </td>

                    <td className="px-6 py-4 whitespace-nowrap">

                      <span className={`inline-flex items-center px-2.5 py-0.5 rounded-full text-xs font-medium ${

                        user.status === 'active'

                          ? 'bg-green-100 text-green-800'

                          : 'bg-red-100 text-red-800'

                      }`}>

                        {user.status === 'active' ? '🟢 Active' : '🔴 Inactive'}

                      </span>

                    </td>

                    <td className="px-6 py-4 whitespace-nowrap text-sm text-gray-500">

                      {user.lastActive}

                    </td>

                    <td className="px-6 py-4 whitespace-nowrap text-sm text-gray-500">

                      {new Date(user.createdAt).toLocaleDateString()}

                    </td>

                    <td className="px-6 py-4 whitespace-nowrap text-right text-sm font-medium space-x-2">

                      <button

                        onClick={() => handleStatusToggle(user.id)}

                        className={`px-3 py-1 rounded text-xs font-medium transition-colors duration-200 ${

                          user.status === 'active'

                            ? 'bg-yellow-100 text-yellow-800 hover:bg-yellow-200'

                            : 'bg-green-100 text-green-800 hover:bg-green-200'

                        }`}

                      >

                        {user.status === 'active' ? 'Deactivate' : 'Activate'}

                      </button>

                      <button className="px-3 py-1 bg-blue-100 text-blue-800 rounded text-xs font-medium hover:bg-blue-200 transition-colors duration-200">

                        Edit

                      </button>

                      <button

                        onClick={() => handleDeleteUser(user.id)}

                        className="px-3 py-1 bg-red-100 text-red-800 rounded text-xs font-medium hover:bg-red-200 transition-colors duration-200"

                      >

                        Delete

                      </button>

                    </td>

                  </tr>

                ))}

              </tbody>

            </table>

          </div>

        </div>

        {/\* Stats \*/}

        <div className="mt-6 grid grid-cols-1 gap-5 sm:grid-cols-2 lg:grid-cols-4">

          <div className="bg-white overflow-hidden shadow rounded-lg">

            <div className="p-5">

              <div className="flex items-center">

                <div className="flex-shrink-0 text-2xl">👥</div>

                <div className="ml-5 w-0 flex-1">

                  <dl>

                    <dt className="text-sm font-medium text-gray-500 truncate">Total Users</dt>

                    <dd className="text-lg font-medium text-gray-900">{users.length}</dd>

                  </dl>

                </div>

              </div>

            </div>

          </div>

          <div className="bg-white overflow-hidden shadow rounded-lg">

            <div className="p-5">

              <div className="flex items-center">

                <div className="flex-shrink-0 text-2xl">👑</div>

                <div className="ml-5 w-0 flex-1">

                  <dl>

                    <dt className="text-sm font-medium text-gray-500 truncate">Admins</dt>

                    <dd className="text-lg font-medium text-gray-900">

                      {users.filter(u => u.userType === 'admin').length}

                    </dd>

                  </dl>

                </div>

              </div>

            </div>

          </div>

          <div className="bg-white overflow-hidden shadow rounded-lg">

            <div className="p-5">

              <div className="flex items-center">

                <div className="flex-shrink-0 text-2xl">🟢</div>

                <div className="ml-5 w-0 flex-1">

                  <dl>

                    <dt className="text-sm font-medium text-gray-500 truncate">Active</dt>

                    <dd className="text-lg font-medium text-gray-900">

                      {users.filter(u => u.status === 'active').length}

                    </dd>

                  </dl>

                </div>

              </div>

            </div>

          </div>

          <div className="bg-white overflow-hidden shadow rounded-lg">

            <div className="p-5">

              <div className="flex items-center">

                <div className="flex-shrink-0 text-2xl">🔴</div>

                <div className="ml-5 w-0 flex-1">

                  <dl>

                    <dt className="text-sm font-medium text-gray-500 truncate">Inactive</dt>

                    <dd className="text-lg font-medium text-gray-900">

                      {users.filter(u => u.status === 'inactive').length}

                    </dd>

                  </dl>

                </div>

              </div>

            </div>

          </div>

        </div>

      </div>

    </ProtectedRoute>

  );

};

export default Users;

**13.Auth.ts**

export interface User {

  id: string;

  email: string;

  name: string;

  userType: 'admin' | 'user';

  createdAt: string;

}

export interface AuthState {

  user: User | null;

  isAuthenticated: boolean;

  isLoading: boolean;

}

export interface LoginFormData {

  email: string;

  password: string;

}

export interface SignUpFormData {

  name: string;

  email: string;

  password: string;

  confirmPassword: string;

  userType: 'admin' | 'user';

}

export interface FormErrors {

  [key: string]: string;

}

**14.AuthStorage.ts**

import type { User } from '../types/auth';

const USER\_STORAGE\_KEY = 'auth\_user';

const SESSION\_STORAGE\_KEY = 'auth\_session';

export const authStorage = {

  // Local Storage - persists user data

  saveUser: (user: User): void => {

    localStorage.setItem(USER\_STORAGE\_KEY, JSON.stringify(user));

  },

  getUser: (): User | null => {

    const userData = localStorage.getItem(USER\_STORAGE\_KEY);

    return userData ? JSON.parse(userData) : null;

  },

  removeUser: (): void => {

    localStorage.removeItem(USER\_STORAGE\_KEY);

  },

  // Session Storage - manages authentication state

  saveSession: (token: string): void => {

    sessionStorage.setItem(SESSION\_STORAGE\_KEY, token);

  },

  getSession: (): string | null => {

    return sessionStorage.getItem(SESSION\_STORAGE\_KEY);

  },

  removeSession: (): void => {

    sessionStorage.removeItem(SESSION\_STORAGE\_KEY);

  },

  // Clear all auth data

  clearAll: (): void => {

    authStorage.removeUser();

    authStorage.removeSession();

  }

};

**15.Validation.ts**

import type { LoginFormData, SignUpFormData, FormErrors } from '../types/auth';

export const validateEmail = (email: string): boolean => {

  const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

  return emailRegex.test(email);

};

export const validatePassword = (password: string): boolean => {

  // Password must be at least 8 characters with at least one uppercase, one lowercase, and one number

  const passwordRegex = /^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)[a-zA-Z\d@$!%\*?&]{8,}$/;

  return passwordRegex.test(password);

};

export const validateLoginForm = (data: LoginFormData): FormErrors => {

  const errors: FormErrors = {};

  if (!data.email.trim()) {

    errors.email = 'Email is required';

  } else if (!validateEmail(data.email)) {

    errors.email = 'Please enter a valid email address';

  }

  if (!data.password) {

    errors.password = 'Password is required';

  }

  return errors;

};

export const validateSignUpForm = (data: SignUpFormData): FormErrors => {

  const errors: FormErrors = {};

  if (!data.name.trim()) {

    errors.name = 'Name is required';

  } else if (data.name.trim().length < 2) {

    errors.name = 'Name must be at least 2 characters';

  }

  if (!data.email.trim()) {

    errors.email = 'Email is required';

  } else if (!validateEmail(data.email)) {

    errors.email = 'Please enter a valid email address';

  }

  if (!data.password) {

    errors.password = 'Password is required';

  } else if (!validatePassword(data.password)) {

    errors.password = 'Password must be at least 8 characters with uppercase, lowercase, and number';

  }

  if (!data.confirmPassword) {

    errors.confirmPassword = 'Please confirm your password';

  } else if (data.password !== data.confirmPassword) {

    errors.confirmPassword = 'Passwords do not match';

  }

  if (!data.userType) {

    errors.userType = 'Please select a user type';

  }

  return errors;

};

**16.App.css**

#root {

  max-width: 1280px;

  margin: 0 auto;

  padding: 2rem;

  text-align: center;

}

.logo {

  height: 6em;

  padding: 1.5em;

  will-change: filter;

  transition: filter 300ms;

}

.logo:hover {

  filter: drop-shadow(0 0 2em #646cffaa);

}

.logo.react:hover {

  filter: drop-shadow(0 0 2em #61dafbaa);

}

@keyframes logo-spin {

  from {

    transform: rotate(0deg);

  }

  to {

    transform: rotate(360deg);

  }

}

@media (prefers-reduced-motion: no-preference) {

  a:nth-of-type(2) .logo {

    animation: logo-spin infinite 20s linear;

  }

}

.card {

  padding: 2em;

}

.read-the-docs {

  color: #888;

}

**17.App.tsx**

import React, { Suspense, lazy } from 'react';

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

import { AuthProvider, useAuth } from './contexts/AuthContext';

import ProtectedRoute from './components/ProtectedRoute';

import DashboardLayout from './components/DashboardLayout';

import './App.css';

// Lazy load components for better performance

const Home = lazy(() => import('./pages/Home'));

const Login = lazy(() => import('./components/Login'));

const SignUp = lazy(() => import('./components/SignUp'));

const Dashboard = lazy(() => import('./pages/Dashboard'));

const Profile = lazy(() => import('./pages/Profile'));

const Settings = lazy(() => import('./pages/Settings'));

const Users = lazy(() => import('./pages/Users'));

const Analytics = lazy(() => import('./pages/Analytics'));

// Loading component

const LoadingSpinner: React.FC = () => (

  <div className="min-h-screen flex items-center justify-center">

    <div className="animate-spin rounded-full h-12 w-12 border-b-2 border-indigo-600"></div>

  </div>

);

// Dashboard wrapper to handle layout

const DashboardWrapper: React.FC<{ children: React.ReactNode }> = ({ children }) => (

  <DashboardLayout>{children}</DashboardLayout>

);

// Auth redirect component

const AuthRedirect: React.FC = () => {

  const { isAuthenticated } = useAuth();

  return isAuthenticated ? <Navigate to="/dashboard" replace /> : <Navigate to="/login" replace />;

};

function App() {

  return (

    <AuthProvider>

      <Router>

        <div className="App">

          <Suspense fallback={<LoadingSpinner />}>

            <Routes>

              {/\* Public routes \*/}

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

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

              <Route path="/signup" element={<SignUp />} />

              {/\* Protected dashboard routes \*/}

              <Route

                path="/dashboard"

                element={

                  <ProtectedRoute>

                    <DashboardWrapper>

                      <Dashboard />

                    </DashboardWrapper>

                  </ProtectedRoute>

                }

              />

              <Route

                path="/dashboard/profile"

                element={

                  <ProtectedRoute>

                    <DashboardWrapper>

                      <Profile />

                    </DashboardWrapper>

                  </ProtectedRoute>

                }

              />

              <Route

                path="/dashboard/settings"

                element={

                  <ProtectedRoute>

                    <DashboardWrapper>

                      <Settings />

                    </DashboardWrapper>

                  </ProtectedRoute>

                }

              />

              {/\* Admin only routes \*/}

              <Route

                path="/dashboard/users"

                element={

                  <ProtectedRoute requireAdmin={true}>

                    <DashboardWrapper>

                      <Users />

                    </DashboardWrapper>

                  </ProtectedRoute>

                }

              />

              <Route

                path="/dashboard/analytics"

                element={

                  <ProtectedRoute requireAdmin={true}>

                    <DashboardWrapper>

                      <Analytics />

                    </DashboardWrapper>

                  </ProtectedRoute>

                }

              />

              {/\* Catch all route \*/}

              <Route path="\*" element={<AuthRedirect />} />

            </Routes>

          </Suspense>

        </div>

      </Router>

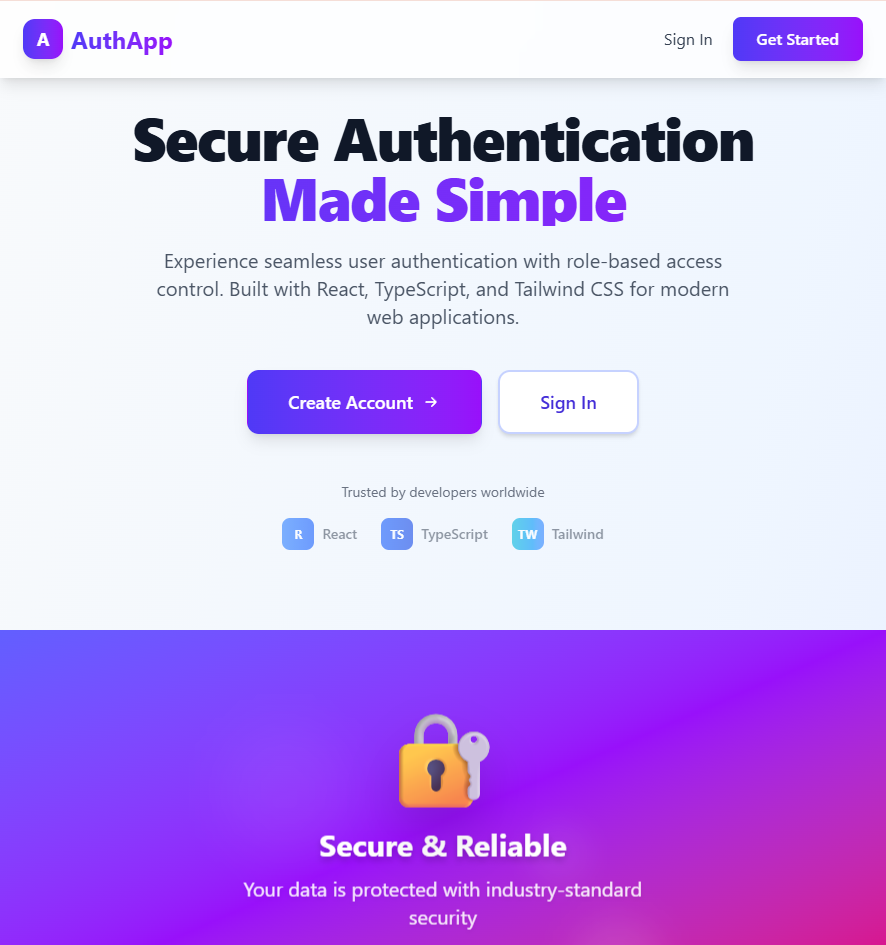
    </AuthProvider>

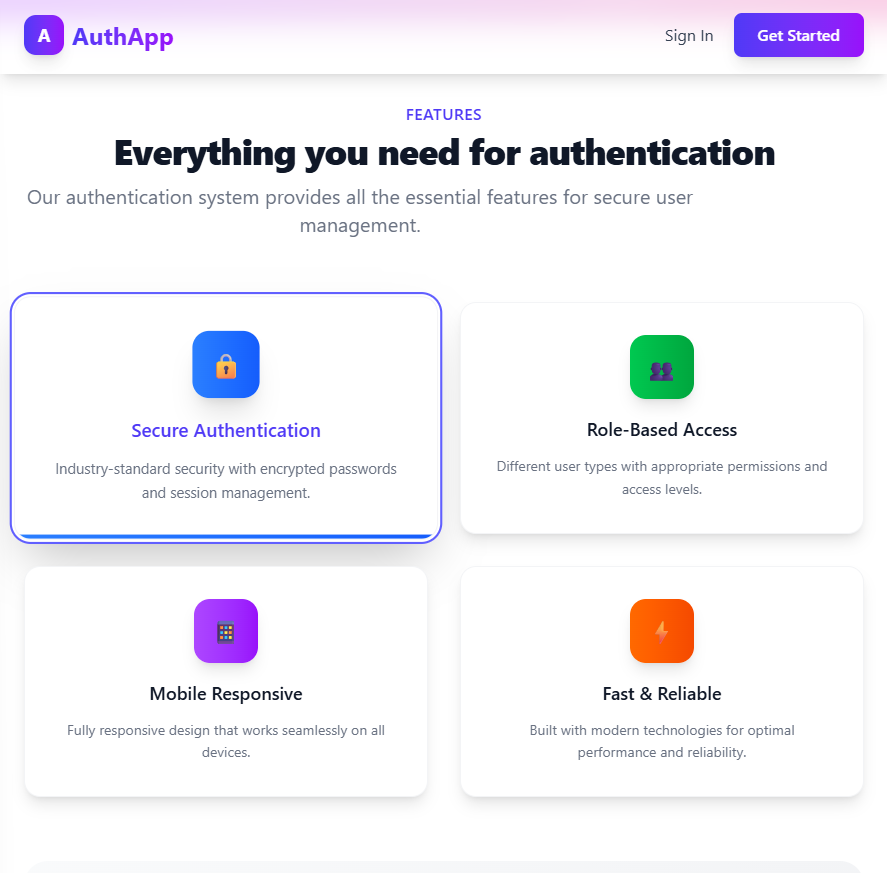
  );

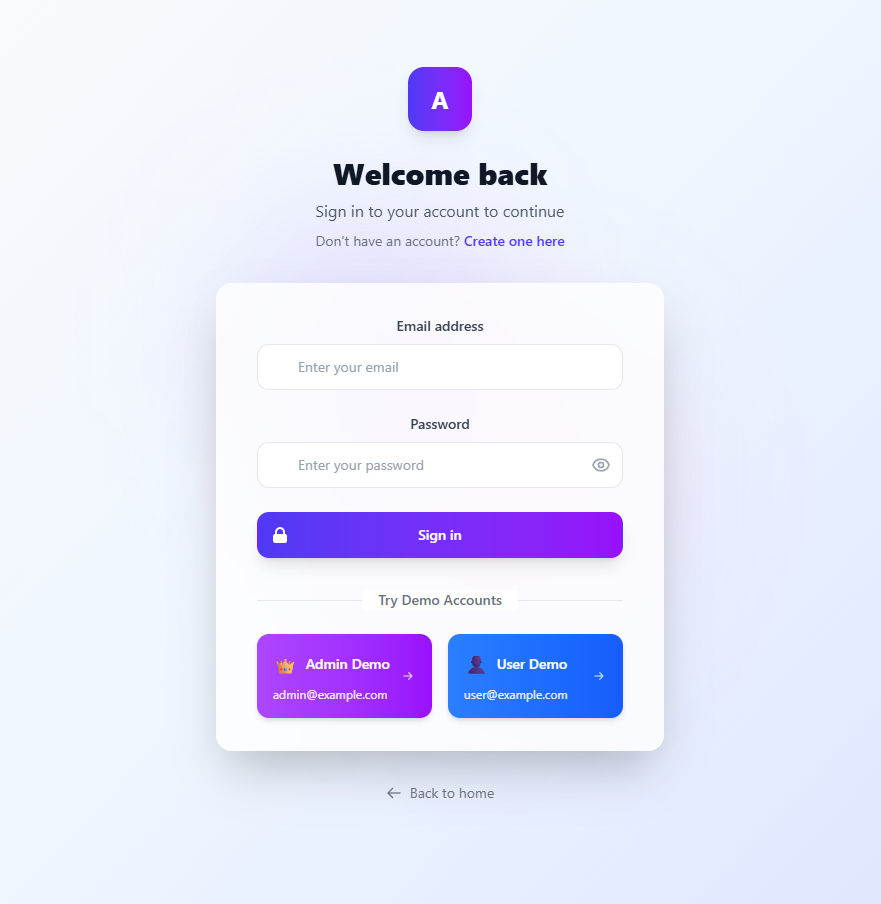
}

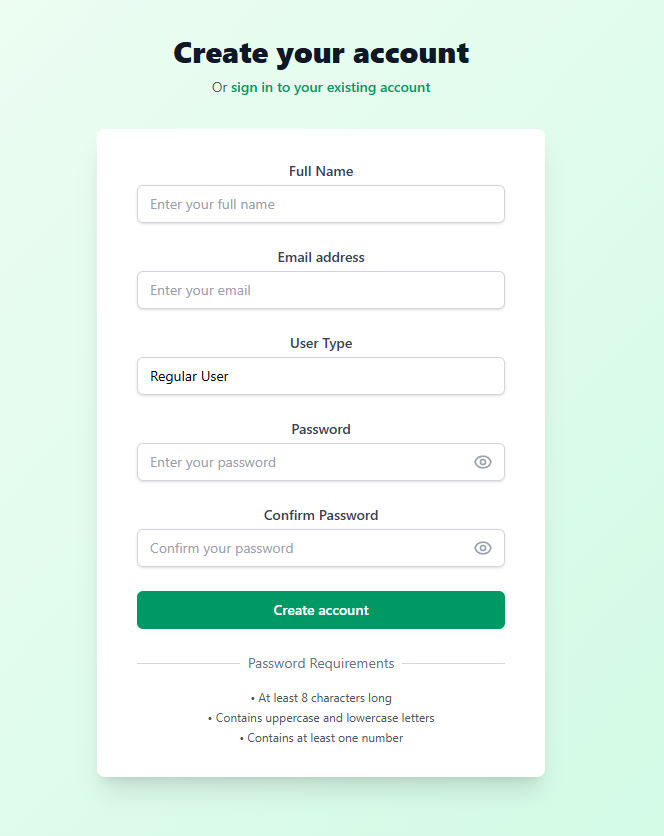
export default App;

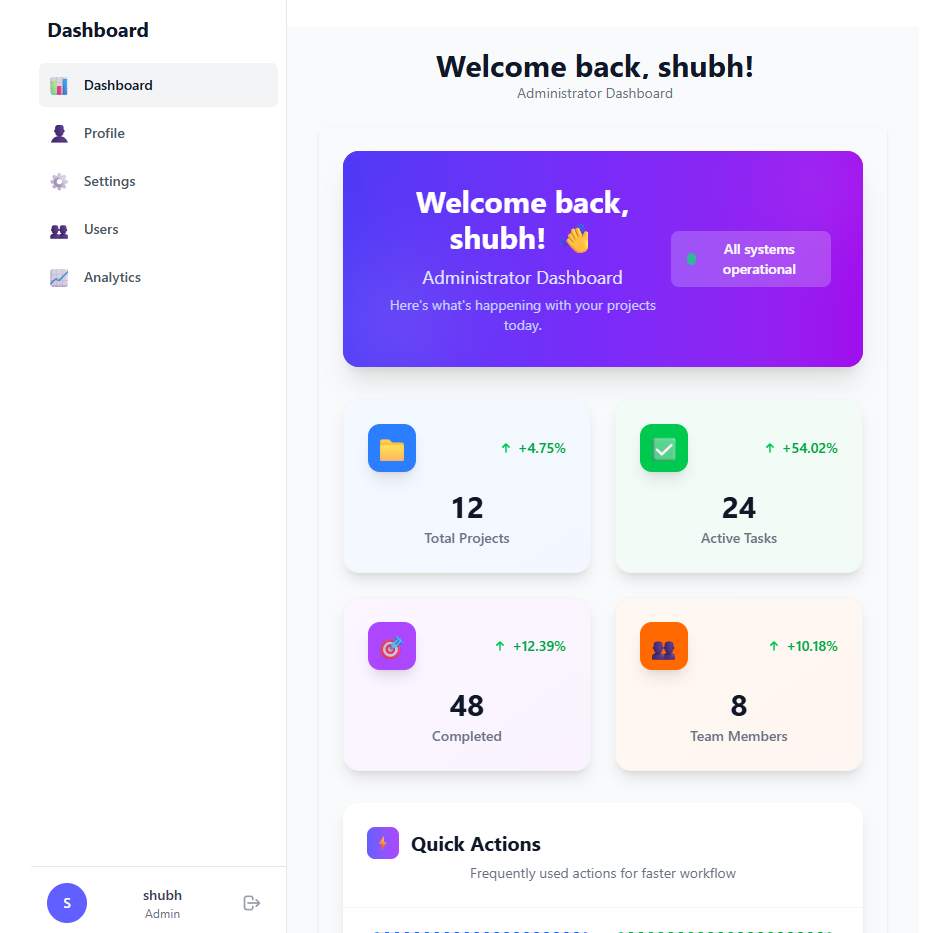
**ScreenShots of Website:**

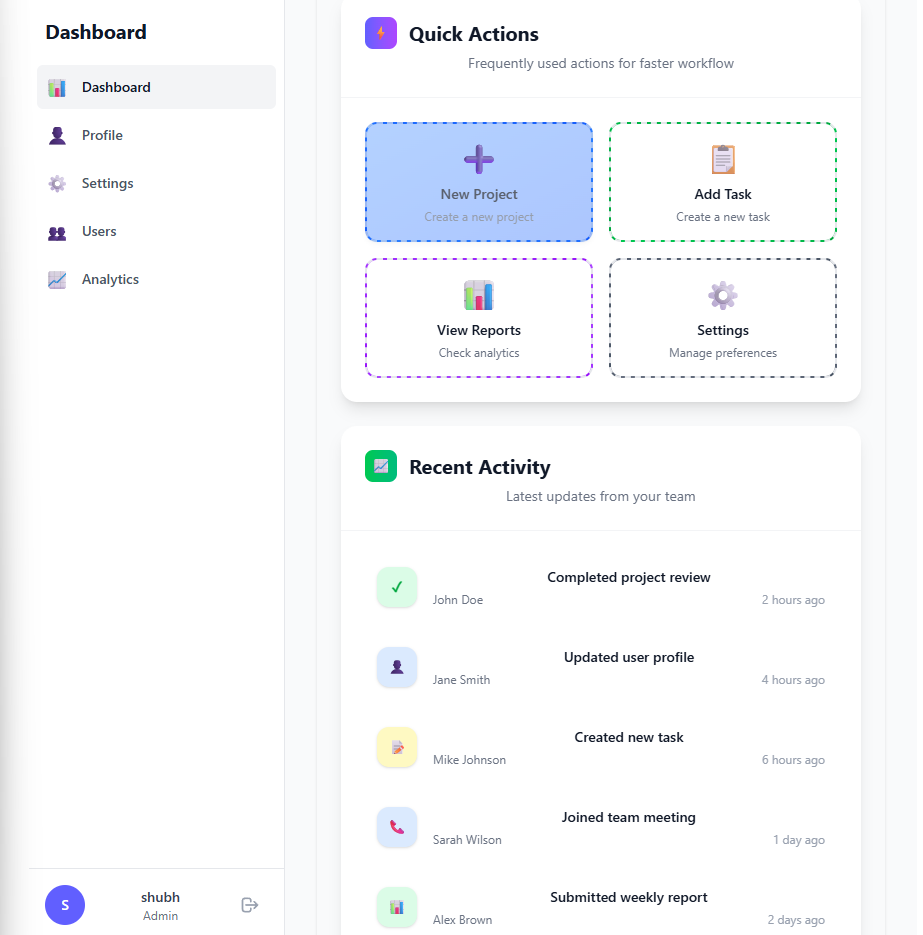
****

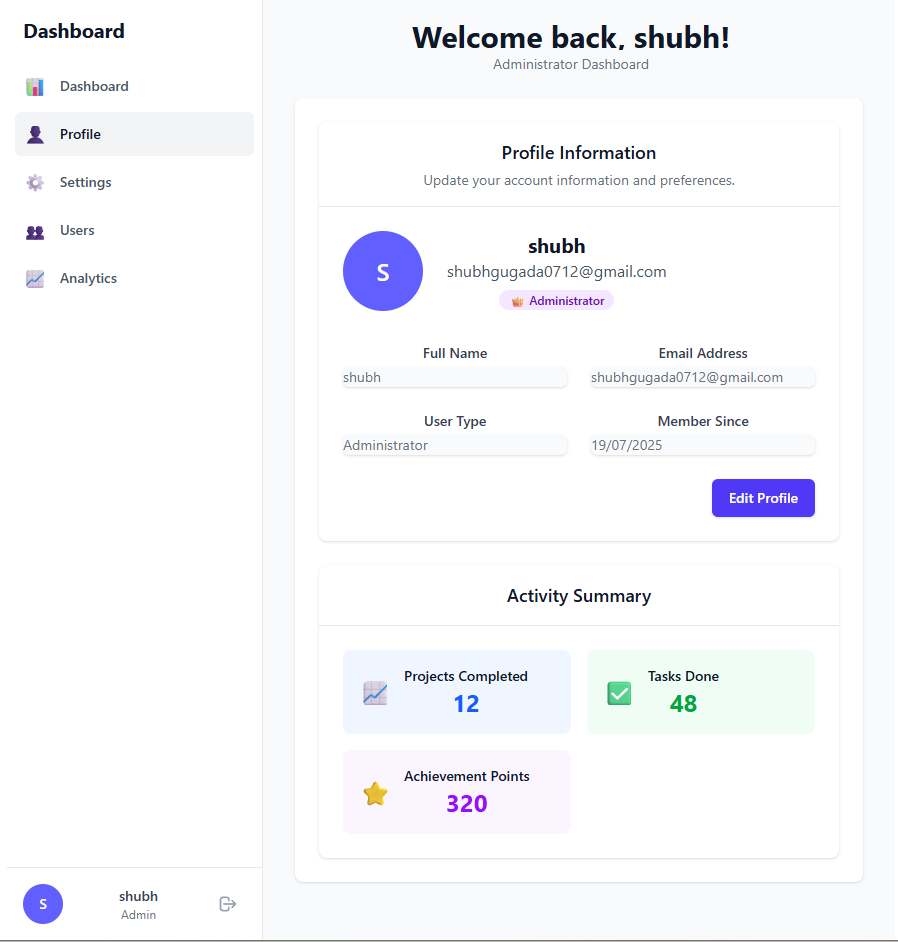
****

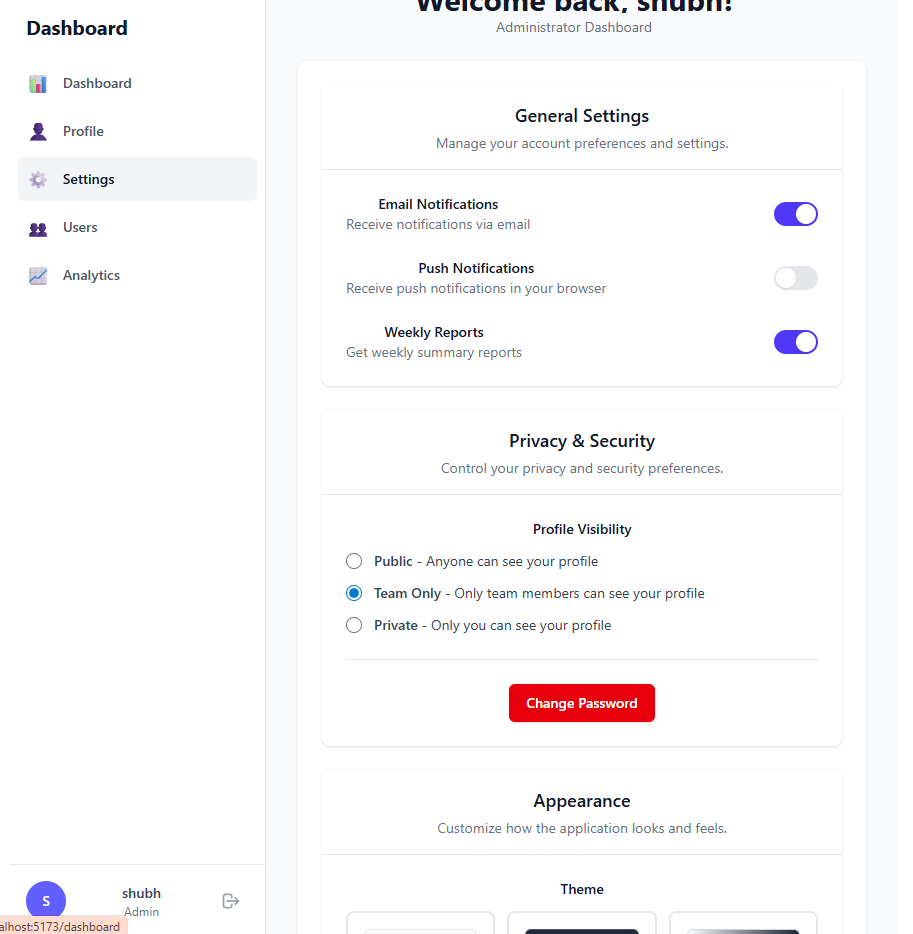
****

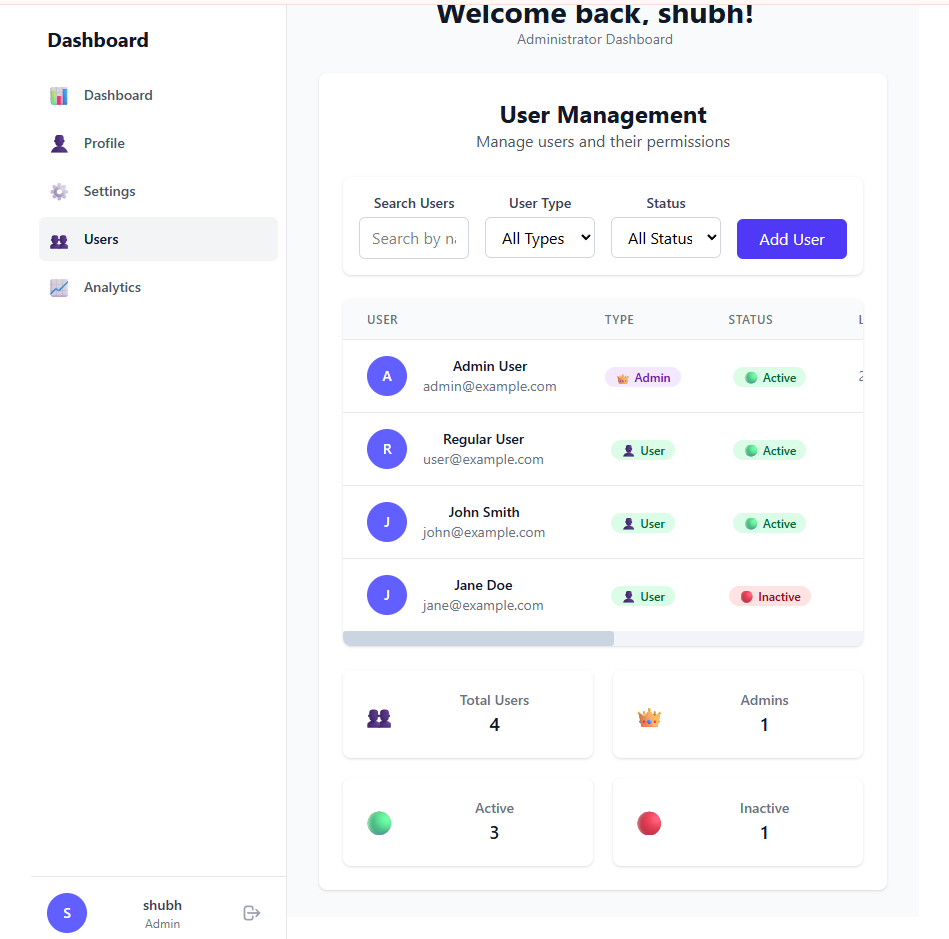
****

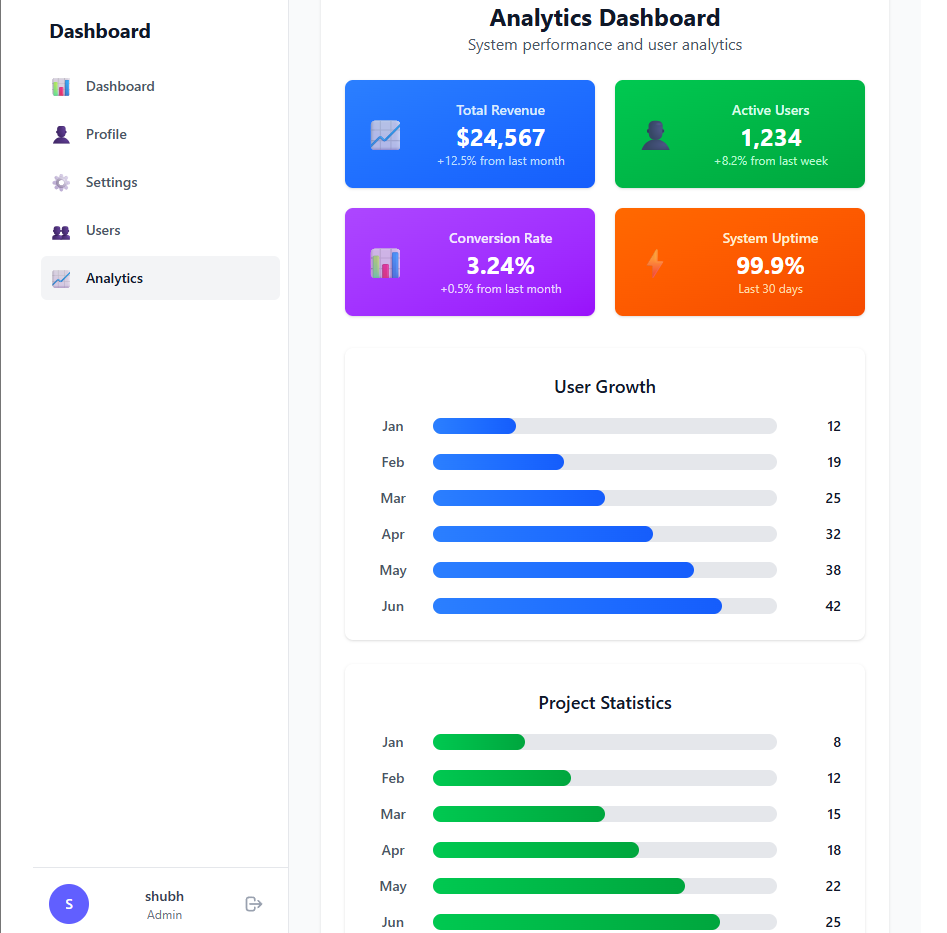
****

****

****

****

****

****