**## User Management System Documentation**

**## Introduction**

This documentation outlines the development and functionality of our website

**## Installation and Setup**

->Unzip the document.

-> Go to the working directory.

-> open command prompt in the working directory and run following commands

**npm install**

**npm start**

**## Here are The Credentials to Login to the System**

**Super Admin**

email: superadmin@gmail.com

password: Superadmin@123

**Admin**

email: admin@gmail.com

password": Admin@123

**User**

email: user@gmail.com

password": User@123

**### Technology Used**

React.js, Material UI, Redux-Toolkit (for state-management)

**###System Flows**

**User Logs In:**

All users, including Admin Users, Super Admins, and Regular Users, log in with their credentials.

**Admin User Actions:**

After logging in, Admin Users are redirected to the Admin Dashboard, where they can:

Create new Users.

Edit existing User profiles.

Create new Types of Users.

Edit existing Types of Users.

Admin Users cannot delete Users or Types of Users.

**Super Admin Actions:**

After logging in, Super Admins are redirected to the Super Admin Dashboard, where they can:

Create new Users.

Edit existing User profiles.

Delete Users.

Create new Types of Users.

Edit existing Types of Users.

Delete Types of Users.

**Regular User Actions:**

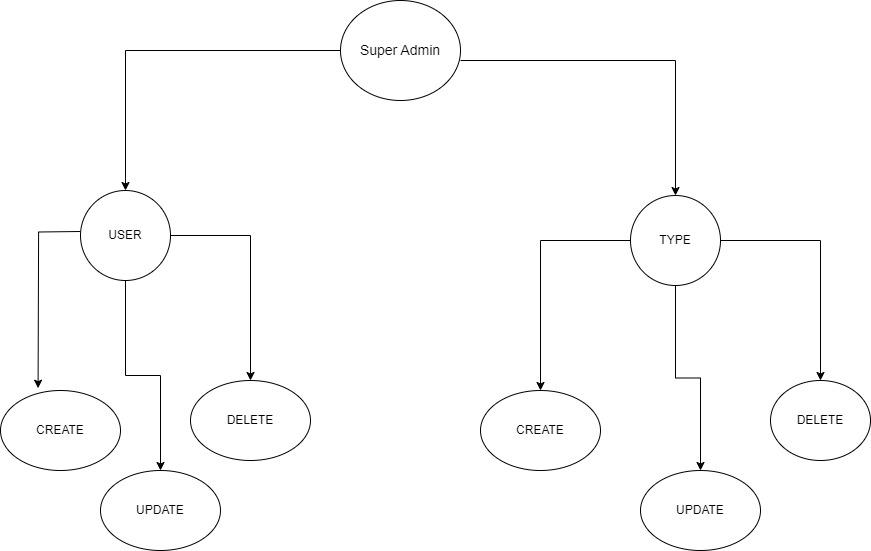
After logging in, Regular Users are redirected to their User Dashboard, where they can:

View their personal information.

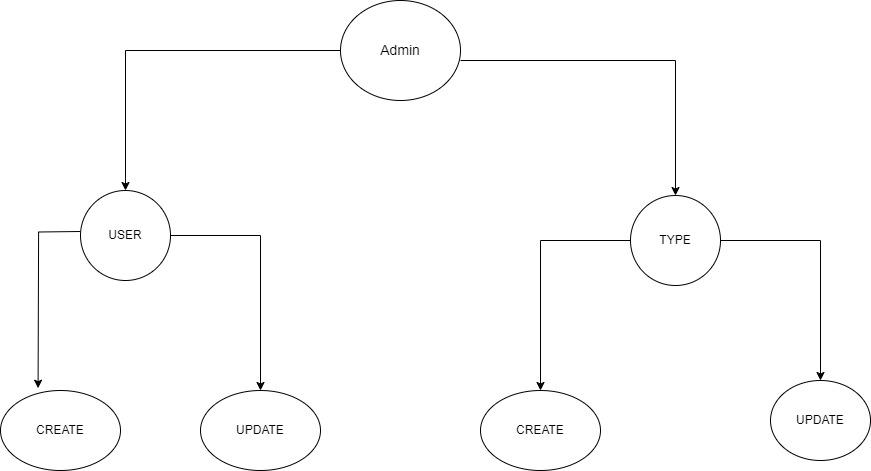
See the Types assigned to them.

**### Diagrams**

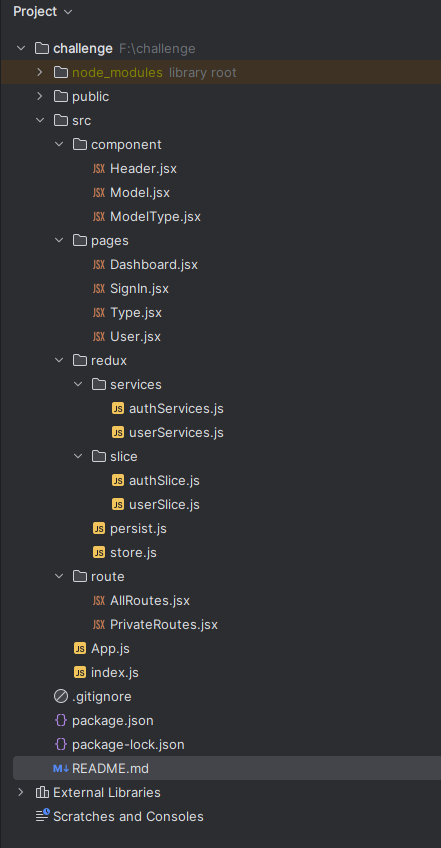
**#Super Admin**



**#ADMIN**



**### Code Structure**

****

**component**

**-> Header.jsx**: Explanation of the Header component, its role, and usage.

**-> Model.jsx**: Information about the Model component and how it's used.

**-> ModelType.jsx**: Description of the ModelType component and its purpose.

**Pages**

**-> Dashboard.jsx**: Description of the Dashboard component and its functionality.

**-> SignIn.jsx**: Details about the SignIn page, its layout, and user interactions.

**-> Type.jsx**: Explanation of the Type page and its functionality.

**-> User.jsx**: Information about the User page, what it displays, and how it's used.

**redux**

**-> service**: A folder containing service modules for interacting with the data.

**-> authServices.js**: Explanation of the authServices module for handling

authentication-related operations.

**-> userServices.js**: Details about the userServices module for managing user-related data.

**-> slice**: This directory contains Redux slices, which define the state and reducers for different

parts of the application.

**-> authSlice.js**: Information about the authSlice, which handles authentication state.

**-> userService.js**: Description of the userService slice, responsible for managing user data.

**-> persist.js**: Explanation of the persist.js file, which handles state persistence using Redux

Persist.

**-> store.js**: Details about the Redux store configuration and initialization.

**route**

**AllRoutes.jsx**: Overview of the AllRoutes component, which contains all application routes.

**PrivateRoutes.jsx**: Explanation of the PrivateRoutes component, which handles authenticated routes and redirects.