

# Role-Based User Management System

## Using ReactJS, Node.js, MongoDB, and SCSS

### Introduction

This project is a full-stack Role-Based User Management System built with:

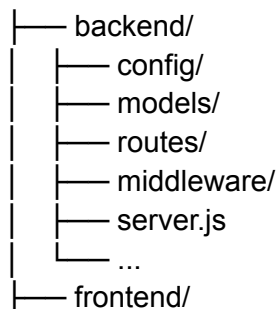
- Frontend: ReactJS (with functional components & hooks), styled using SCSS
- Backend: Node.js (Express.js)
- Database: MongoDB
- Authentication: JWT-based login, bcrypt password hashing
- Features:
  - Role-based access (Admin & User)
  - Admin can manage all users (create users, view all)
  - Users can access their own restricted dashboard
  - Responsive, modern UI using SCSS
  - Proper input validation & notification for UX

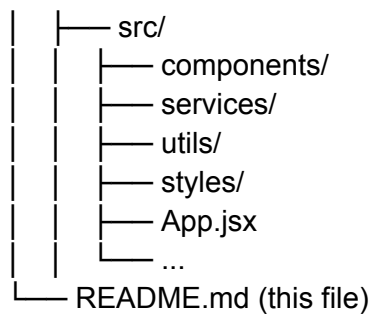
### Features

- Two user roles: Admin and User
- Admin can create new users and view all users
- Users can only access their own dashboard
- First Admin must be inserted directly in the database for security (no signup)
- Login: JWT-based secure authentication
- Logout: Clears session/localStorage
- Notifications: Success/error feedback via toasts
- SCSS Styling: Clean, consistent, and responsive

### Project Structure

role-based-user-management/





## Prerequisites

- Node.js (v14 or higher)
- npm (v6 or higher)
- MongoDB (local server or MongoDB Compass)
- Code Editor (Recommended: VS Code)

## Setup: Step-by-Step Guide

1. Clone the Repository or Create Folders.

2. Backend Setup : Create **config**, **models**, **routes**, and **middleware** folders.

3. MongoDB: Insert First Admin Using MongoDB Compass or mongosh.

4. Start the Backend :

cd backend

npm run dev

Check terminal for: **Server running on port 5000** and **MongoDB Connected: localhost**.

5. Frontend Setup : Create the directory structure and add the frontend source files as provided (components, styles, services, etc).

Add to **frontend/package.json**: "proxy": "<http://localhost:5000>"

6. Start the Frontend :

npm start

Access the app at <http://localhost:3000>

# How to Use the Application

## 1. Login:

- Go to <http://localhost:3000>
- Login as admin:
  - Email: [admin@example.com](mailto:admin@example.com)
  - Password: [password](#)

## 2. Admin Dashboard:

- Create users (choose admin/user role)
- See list of all users

## 3. Regular User:

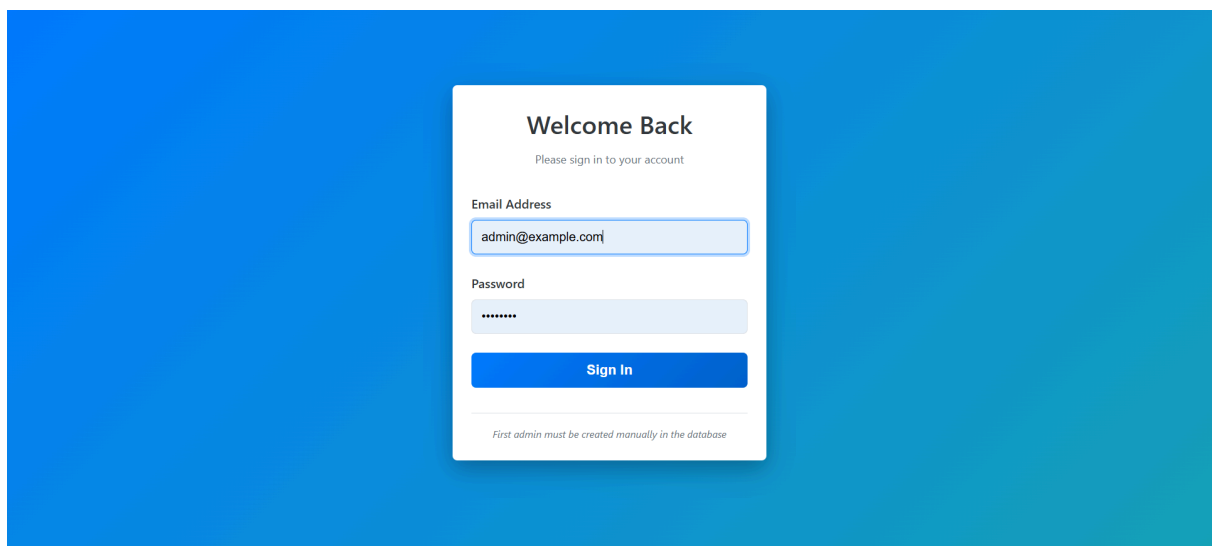
- If you create a user with role [user](#), log in with those credentials
- Regular users see only their basic dashboard

## 4. Security:

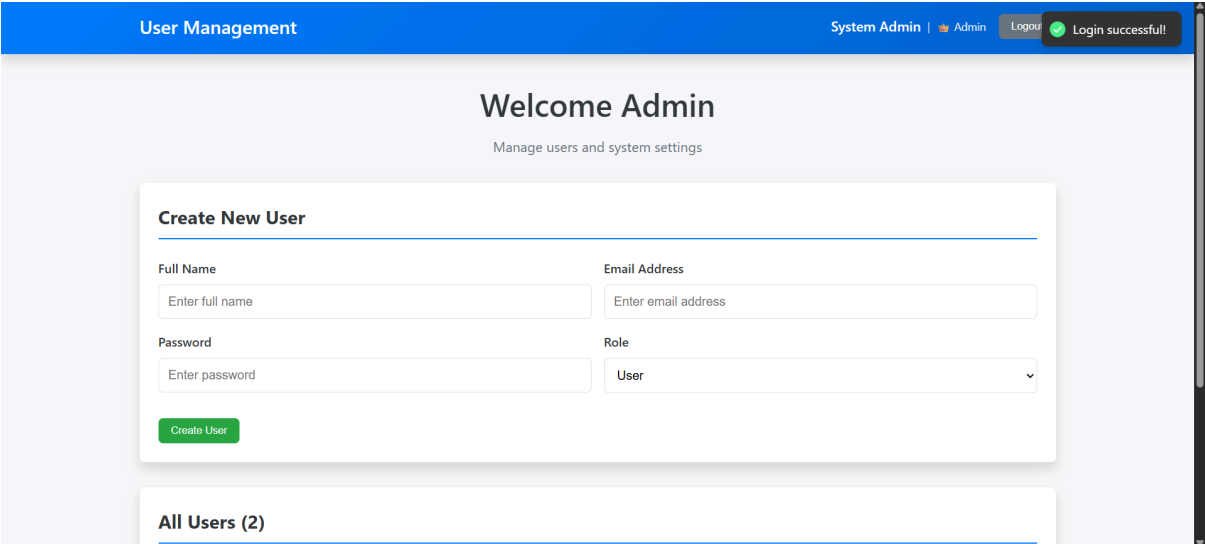
- Logout to end a session
- Admin-only pages are protected

# Result Screenshots

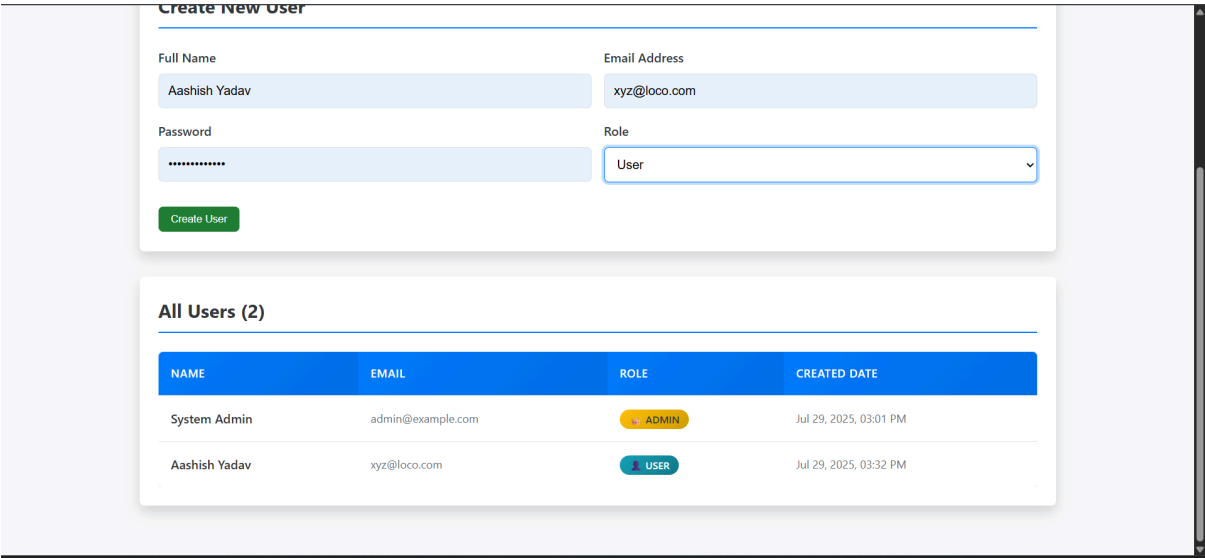
- Login page



- Admin welcome/dashboard



- User creation form & user table



- Regular user dashboard

