

MERN Role-Based Admin Panel Documentation

Project Overview

This is a full-stack MERN (MongoDB, Express.js, React.js, Node.js) application that enables role-based user access and management. The system is designed with three types of users:

- **Admin:** Can manage users, assign roles, and handle products and view orders
 - **Manager:** Can view their team, view and manage their team's orders.
 - **Employee:** Can view products and place orders.
-

Tech Stack

- **Frontend:** React.js, TypeScript, Tailwind CSS, Axios, React Router
 - **Backend:** Node.js, Express.js, TypeScript, MongoDB, Mongoose
 - **Auth:** JWT (JSON Web Tokens)
 - **Deployment:** netlify (Frontend), Render (Backend)
-

Project Structure

```

.
├── change-networks
│   └── client
│       ├── .env
│       ├── .gitignore
│       ├── eslint.config.js
│       ├── index.html
│       ├── package-lock.json
│       ├── package.json
│       ├── public
│       │   ├── _redirects
│       │   ├── favicon-32x32.png
│       │   └── vite.svg
│       ├── README.md
│       └── src
│           ├── api
│           │   └── axios.ts
│           ├── App.tsx
│           ├── assets
│           │   ├── logo.webp
│           │   └── outfit
│           │       └── fonts

```

```
├── Outfit-Bold.ttf
├── Outfit-Medium.ttf
├── Outfit-Regular.ttf
├── react.svg
├── components
│   └── ConfirmBox.tsx
├── Layout
│   ├── Header.tsx
│   ├── Sidebar.tsx
│   ├── OrderHistory.tsx
│   ├── OrderStats.tsx
│   ├── StatCard.tsx
│   ├── SuccessBox.tsx
├── constants
│   └── sidebarConfig.ts
├── context
│   ├── AuthContext.tsx
│   ├── OrderContext.tsx
│   ├── ProductContext.tsx
│   └── UsersContext.tsx
├── index.css
├── main.tsx
├── pages
│   ├── admin
│   │   ├── AdminDashboard.tsx
│   │   ├── Orders.tsx
│   │   └── UserManagement.tsx
│   ├── employee
│   │   ├── EmployeeDashboard.tsx
│   │   ├── EmployeeOrders.tsx
│   │   └── EmpOrders.tsx
│   ├── Login.tsx
│   ├── manage
│   │   └── ManageProducts.tsx
│   ├── manager
│   │   ├── ManagerDashboard.tsx
│   │   ├── ManagerTeam.tsx
│   │   └── TeamOrders.tsx
│   ├── OrderDetails.tsx
│   ├── PlaceOrder.tsx
│   ├── ProductDetails.tsx
│   └── productsList.tsx
├── routes
│   └── ProtectedRoute.tsx
├── types
│   └── index.ts
├── vite-env.d.ts
├── tsconfig.app.json
├── tsconfig.json
├── tsconfig.node.json
├── vite.config.ts
├── server
├── .env
├── package-lock.json
├── package.json
├── src
│   ├── config
│   │   └── db.ts
│   ├── controllers
│   │   ├── authController.ts
│   │   ├── orderController.ts
│   │   └── productController.ts
```

```
├── userController.ts
├── middleware
│   ├── authMiddleware.ts
│   └── roleMiddleware.ts
├── models
│   ├── Order.ts
│   ├── Product.ts
│   └── User.ts
├── routes
│   ├── authRoutes.ts
│   ├── orderRoutes.ts
│   ├── productRoutes.ts
│   └── userRoutes.ts
├── server.ts
├── utils
│   ├── generateToken.ts
│   └── hashPassword.ts
├── tsconfig.json
├── .gitignore
├── app.rest
└── README.md
```

User Roles & Permissions

1. Admin

- ☒ Add/manage users
- ☒ Assign roles (manager or employee)
- ☒ View all orders
- ☒ Add/edit products

2. Manager

- ☒ View team members
- ☒ View/manage orders from team
- ☒ Add/edit products

3. Employee

- ☒ View products
 - ☒ Place orders
-

API Endpoints

Authentication

Method	Endpoint	Access	Description
POST	/api/auth/login	Public	Login user
POST	/api/auth/register	Admin	Register new user

Users

Method	Endpoint	Role	Description
GET	/api/users	Admin	List all users
POST	/api/users	Admin	Add user with a specific role
PUT	/api/users/:id	Admin	Update user
GET	/api/users/manager/:id	Manager	List team members assigned

Products

Method	Endpoint	Role	Description
GET	/api/products	All Roles	List all products
POST	/api/products	Admin/Manager	Add new product
GET	/api/products/:id	All Roles	View Product
PUT	/api/products/:id	Admin/Manager	Update product
DELETE	/api/products/:id	Admin/Manager	Delete product

Orders

Method	Endpoint	Role	Description
POST	/api/orders	Employee	Place a new order
GET	/api/orders/employee/my-orders	Employee	Employee View orders placed by him
GET	/api/orders/manager/team-orders	Manager	View team orders
PATCH	/api/orders/:id	Manager	Update order status

Routing Overview

- "/login": Login Page
- "/admin/manage-team": Admin user management
- "/admin/orders": Admin order list
- "/team/manage-products": Product management by Admin/Manager
- "/products": Product list
- "/products/:id": Product detail view
- "/orders/:id": Order detail view
- "/admin": Admin dashboard
- "/employee": Employee dashboard
- "/employee/place-order": Place new order
- "/employee/orders": View employee orders

- `"/manager"`: Manager dashboard
- `"/manager/team"`: Manager team view
- `"/manager/orders"`: Manager order management

Design

- Built using **Tailwind CSS**
 - Responsive layouts supporting **mobile**, **tablet**, and **desktop**.
 - Sidebar + Header navigation based on user role..
 - Toast notifications for failure operations and custom popup for success
-

Installation & Setup

Backend

- `cd server`
- `npm install`
- Update [server.ts](#) with `app.use(cors())` to allow all origin requests
- `npx tsc`
- `npm start`

Add **.env**:

```
PORT=5000
MONGO_URI=your_mongo_uri
JWT_SECRET=your_jwt_secret
```

Frontend

- `cd client`
- `npm install`
- `npm run dev`

Add **.env** file with **VITE_API_URL=http://localhost:5000/api**

Deployment

Frontend (Netlify)

- Connect GitHub repo
- Build command: `npm run build`

- Publish directory: `dist`
- Add environment variable: `VITE_API_URL=<backend-url>`

Backend (Render)

- Create Web Service and connect repo
 - Build: `npm install && npx tsc`
 - Start: `node dist/server.js`
 - Environment variables: `MONGO_URI`, `JWT_SECRET`
-

Live Link

Frontend: <https://changenetworks.netlify.app/>

Backend: <https://role-based-admin-panel-t38v.onrender.com>

Sample Access Credentials

Admin Access

Email: `admin@gmail.com`

Password: `Admin@2025`

Manager Access

Email: `pearlkinny2025@gmail.com`

Password: `Pearl@2025`

Employee Access

Email: `employee@gmail.com`

Password: `Employee@2025`