**Role-Based Access Control (RBAC) System**

**Overview**

This project implements a comprehensive **Authentication**, **Authorization**, and **Role-Based Access Control (RBAC)** system for managing user access in a secure and scalable manner. The solution ensures that users can only access resources and perform actions permitted by their roles. It uses **Node.js**, **Express**, and **MongoDB** for the backend and **React** for the frontend.

**Key Features**

**Backend**

* **Authentication**: Secure user registration, login, and logout using **JWT (Access and Refresh Tokens)**.
* **Authorization**: Enforces permissions for routes and resources based on user roles.
* **RBAC**: Implements role-based restrictions with three predefined roles: **Admin**, **Moderator**, and **User**.
* **Email Verification**: Users must verify their email before gaining full access.
* **Security Best Practices**:
  + Passwords hashed with **bcrypt**.
  + Tokens stored securely with proper expiration handling.

**Frontend**

* **Dynamic Role-Based Navigation**: Users see different content and accessible pages based on their roles.
* **React Context API**: Manages authentication state globally.
* **Protected Routes**: Ensures only authorized users can access restricted pages.
* **Responsive and Intuitive UI**.

**Tech Stack**

**Backend**

* **Node.js**
* **Express**
* **MongoDB**
* **JWT** for authentication
* **Nodemailer** for email services

**Frontend**

* **React**
* **React Router** for navigation
* **Axios** for API communication

**Installation and Setup**

**Prerequisites**

* Node.js and npm installed
* MongoDB cluster (or local instance)
* Verified email credentials for testing

**Backend Setup**

1. Clone the repository:

git clone <repository-url>

cd backend

1. Install dependencies:

npm install

1. Create a .env file:

PORT=4000

CORS\_ORIGIN=http://localhost:3000

MONGO\_URI=<Your MongoDB Connection String>

ACCESS\_TOKEN\_SECRET=<Your Access Token Secret>

ACCESS\_TOKEN\_EXPIRY=1d

REFRESH\_TOKEN\_SECRET=<Your Refresh Token Secret>

REFRESH\_TOKEN\_EXPIRY=10d

EMAIL=<Your Email Address>

APP\_PSWD=<Your Email Password>

EMAIL\_VERIFICATION\_SECRET=<Your Email Verification Secret>

1. Start the server:

npm start

1. The backend will be live at [http://localhost:4000](http://localhost:4000/).

**Frontend Setup**

1. Navigate to the frontend directory:

cd frontend

1. Install dependencies:

npm install

1. Start the development server:

npm start

1. Access the frontend at [http://localhost:3000](http://localhost:3000/).

**Demo**

**Users and Roles**

**Admin:**

Email: admin@example.com

Password: AdminPass123

**Moderator:**

Email: moderator@example.com

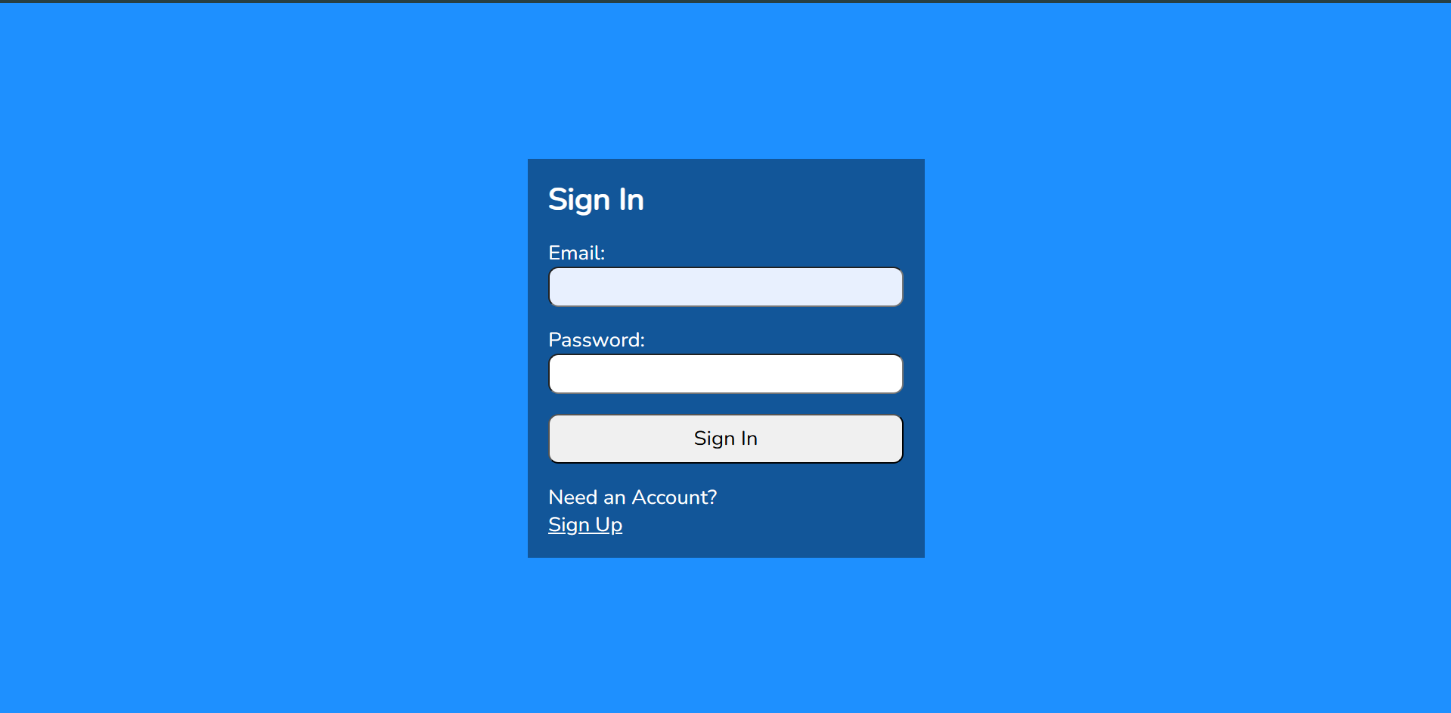
Password: ModPass123

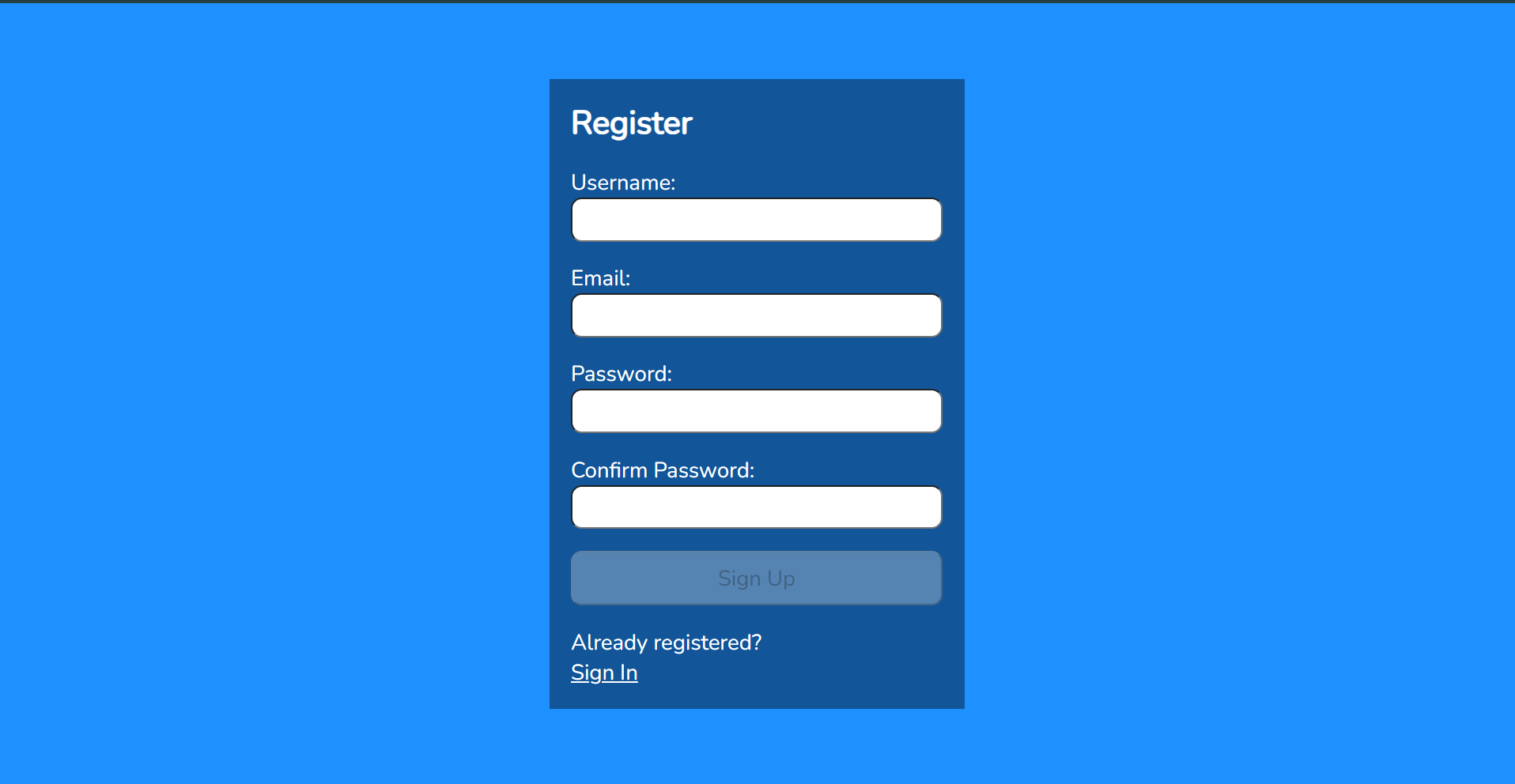
**User:**

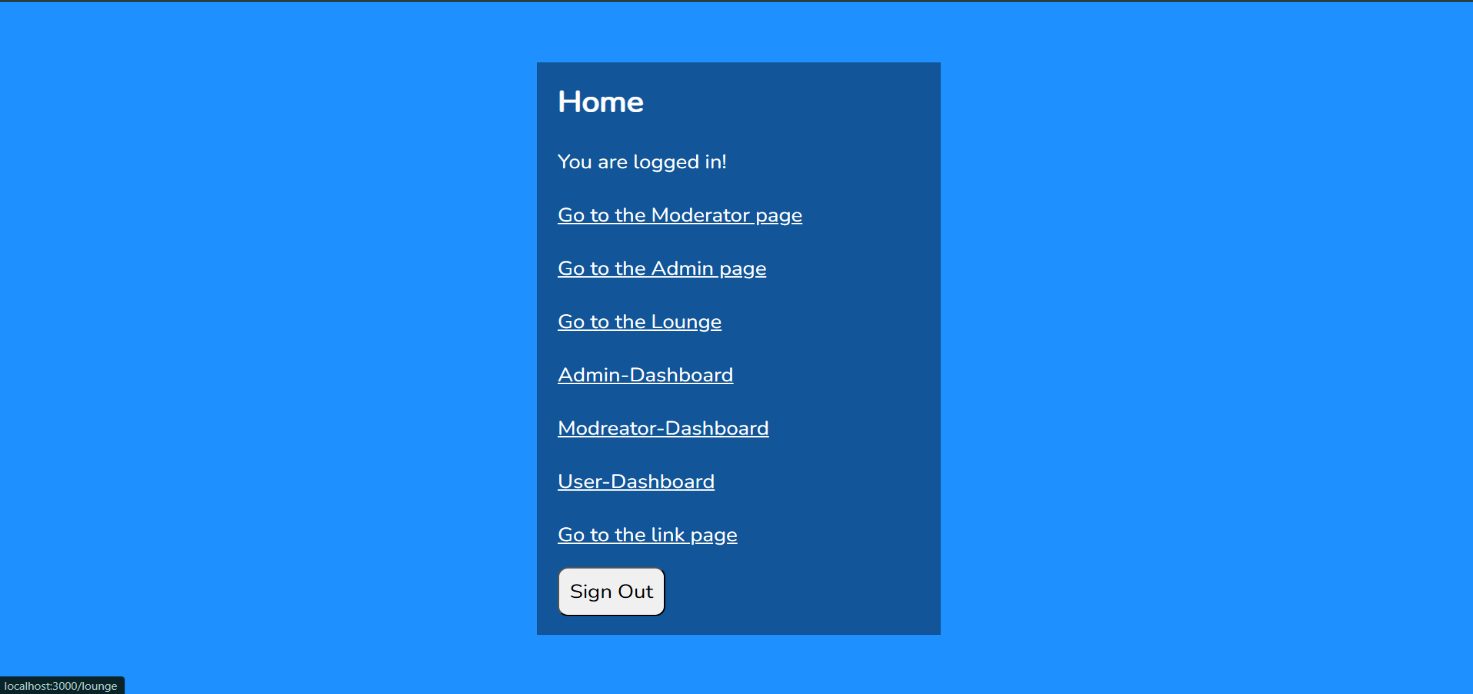
Email: user@example.com

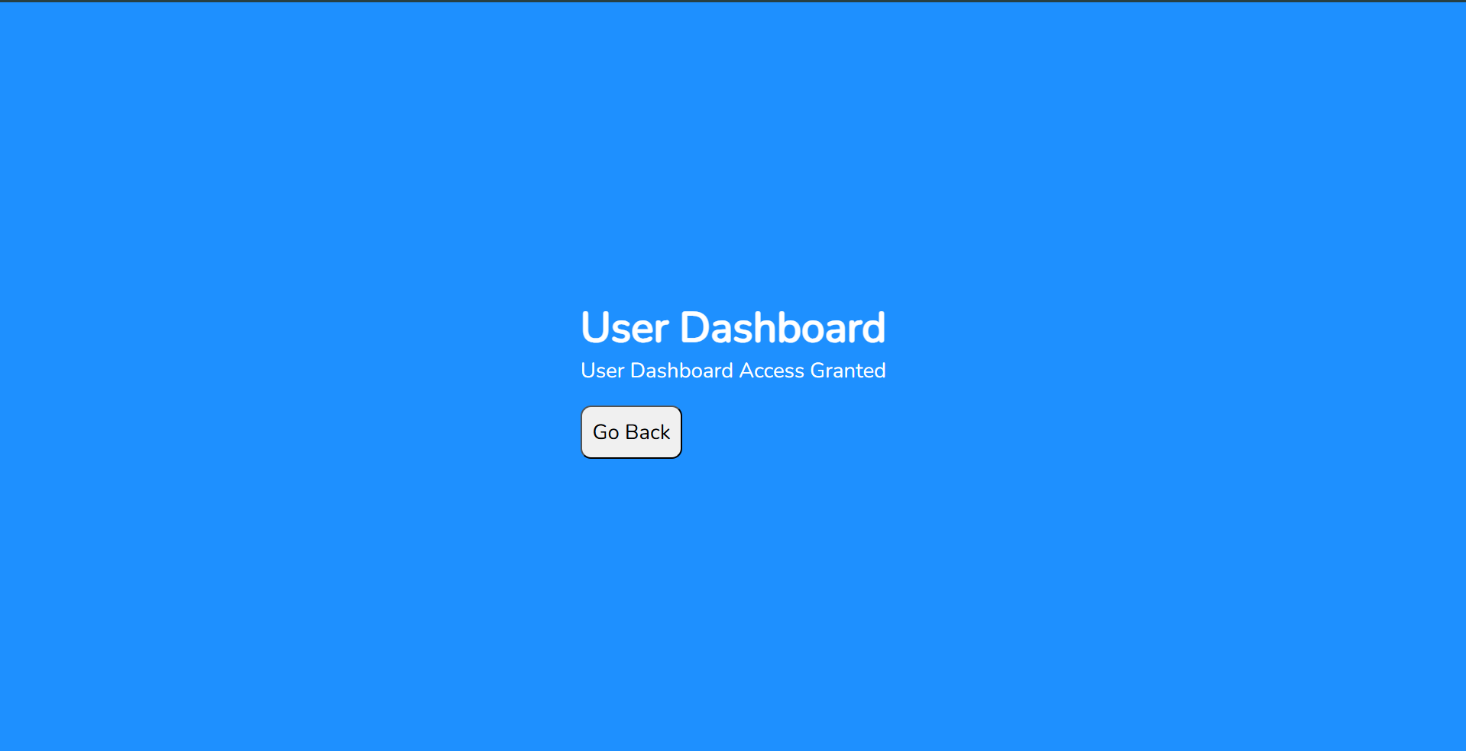
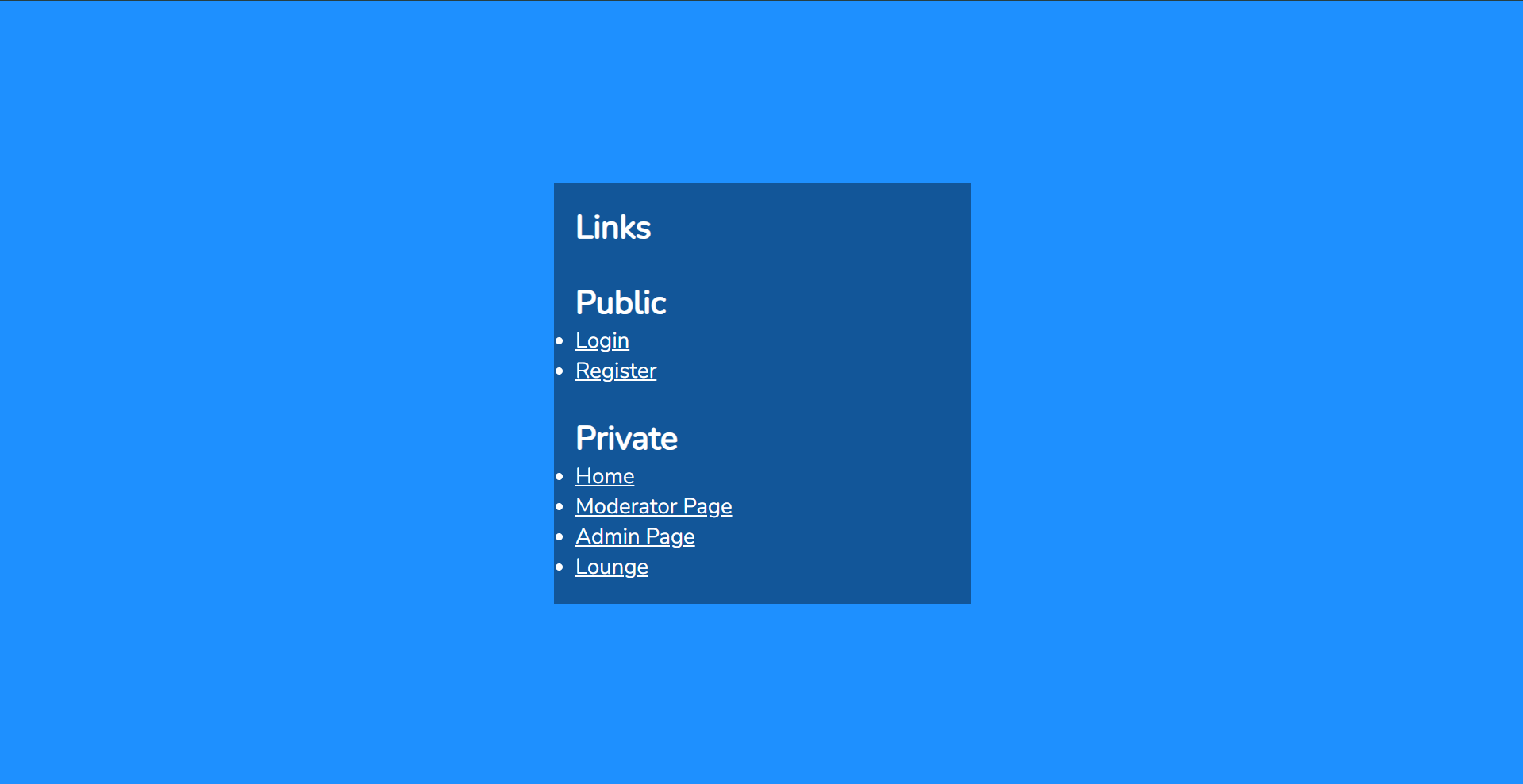
Password: UserPass123

**ScreenShots:**







 ****