1. **Introduction**

* **Project Title: HOUSE RENT APPLICATION USING MERN**
* **Team Members:**
  + Yaashwan S K
  + Libni Macklends K
  + Iliyas R
  + Rajesh V

1. **Project Overview**

* **Purpose:** The app allows users to search for rental properties, book them, and manage listings, while property owners can list, manage, and rent their properties. The project implements modern web technologies to ensure responsiveness, security, and scalability.
* **Features**:

1. User Authentication and Authorization.
2. Property Listings and Management.
3. Booking and Availability Management.
4. Responsive UI and Design.
5. Real-Time Date Management.
6. **Architecture**

* **Frontend:** The frontend is built with React, enhanced by libraries like Axios, Material UI, and Bootstrap to create an interactive and responsive user interface. such as property information, booking details, and user data, while JSON Web Tokens (JWT) handle secure access to restricted routes
* **Backend:** The backend uses Node.js and Express to build RESTful API endpoints that handle business logic, data processing, and secure user authentication. JWT-based authentication ensures only authorized users can access certain features, while bcrypt encrypts passwords for secure storage.
* **Database:** The database uses MongoDB, with Mongoose as an ORM for schema modelling and efficient interaction with data collections. The database stores all essential app information in structured collections, including users, property listings, and bookings, with MongoDB Atlas recommended for scalable, cloud-hosted data management.

1. **Setup Instructions**

* **Prerequisites:**

**1. Node.js (version 22.11.0)**

**2. MongoDB (version 8.0.1)**

* **Installation:**

**1. Navigate to the project directory.**

**2. Install dependencies:**

* **Frontend: cd client && npm install**
* **Backend: cd server && npm install**

**3. Set up environment variables:**

* **MongoDB URI**

1. **Folder Structure**

* **Client:** The client folder is dedicated to the frontend, built with React, and includes a public folder for static assets like images, and an src folder for the main application code**.**
* **Server:** The server folder hosts the Node.js and Express codebase. It includes a config folder for configuration files, such as the MongoDB connection setup. Inside server, an auth folder handles authentication and authorization middleware, and a models folder holds Mongoose schemas for entities like users, properties, and booking.

1. **Running the Application:**

● To start the application locally:

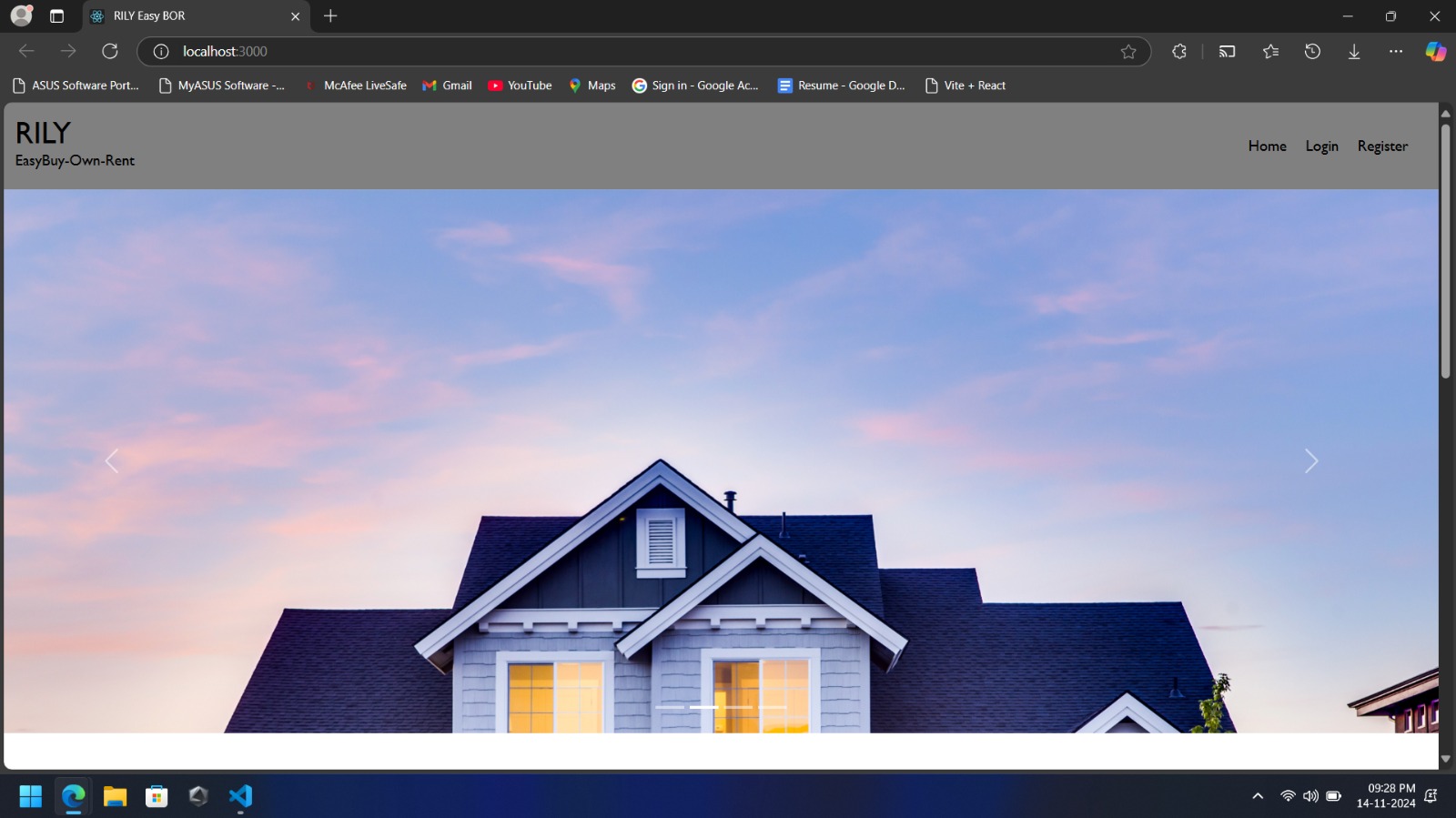
○ Frontend: Run npm start in the client directory.

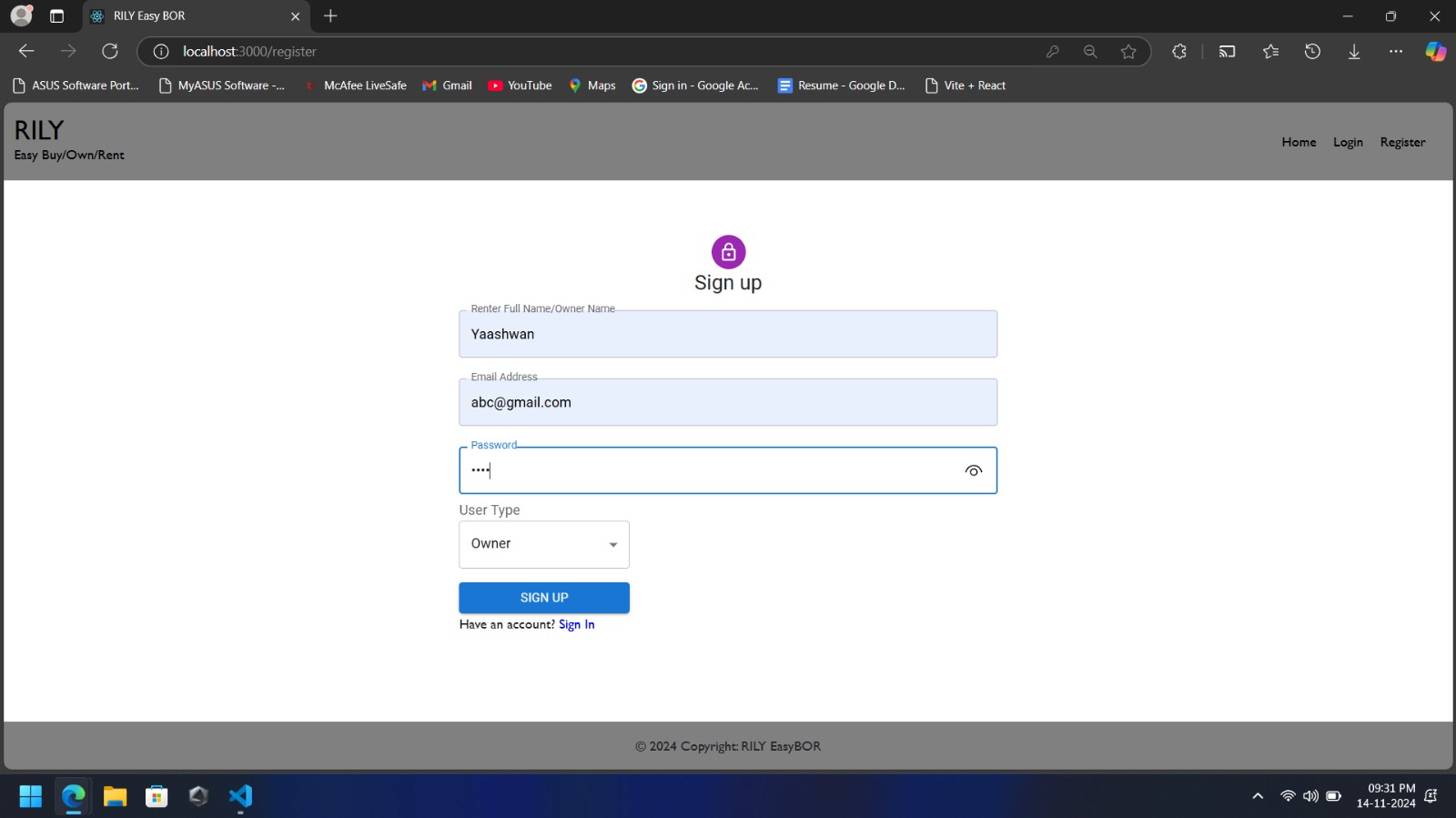
○ Backend: Run npm start in the server directory.

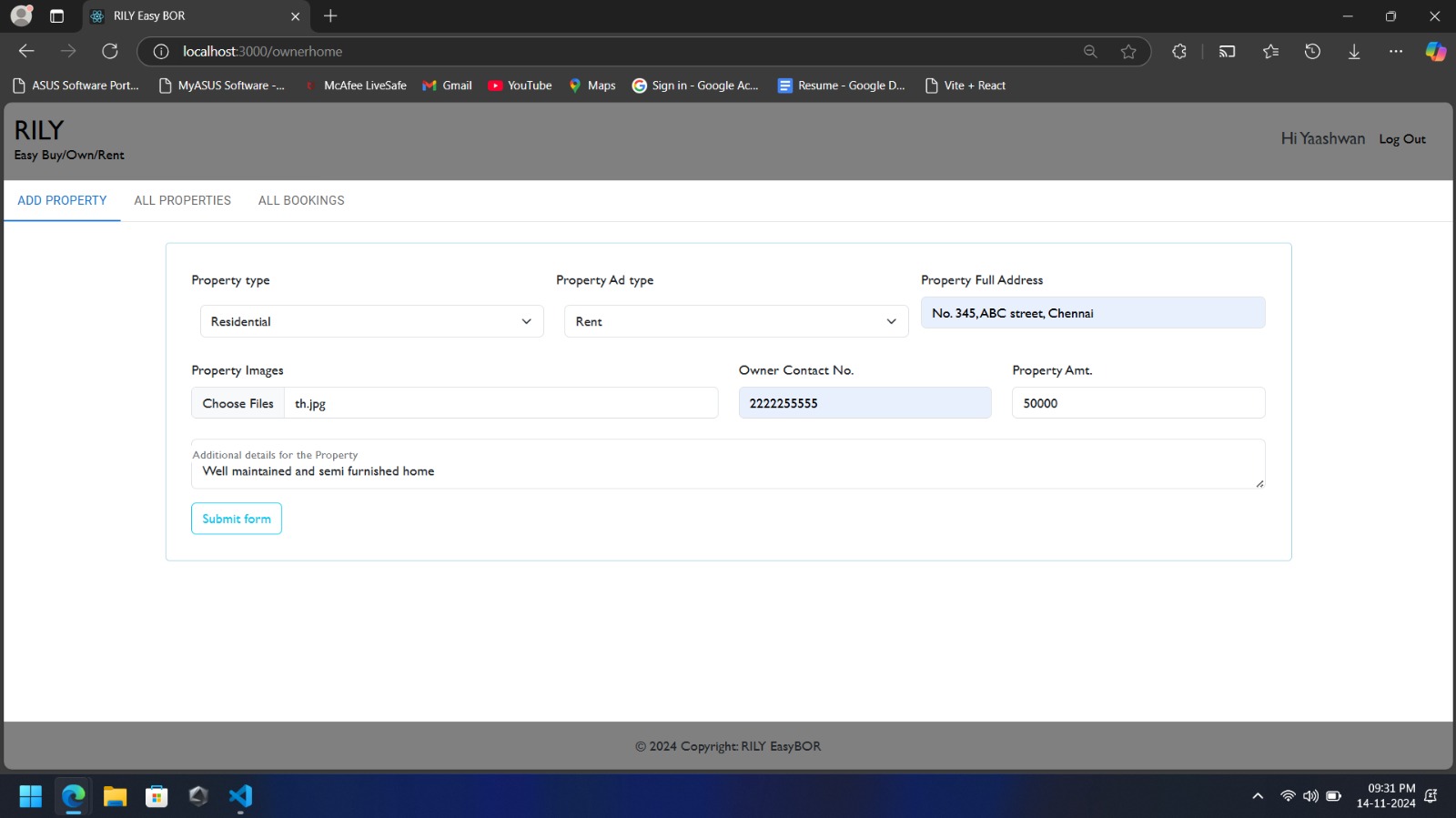
7. **Authentication**

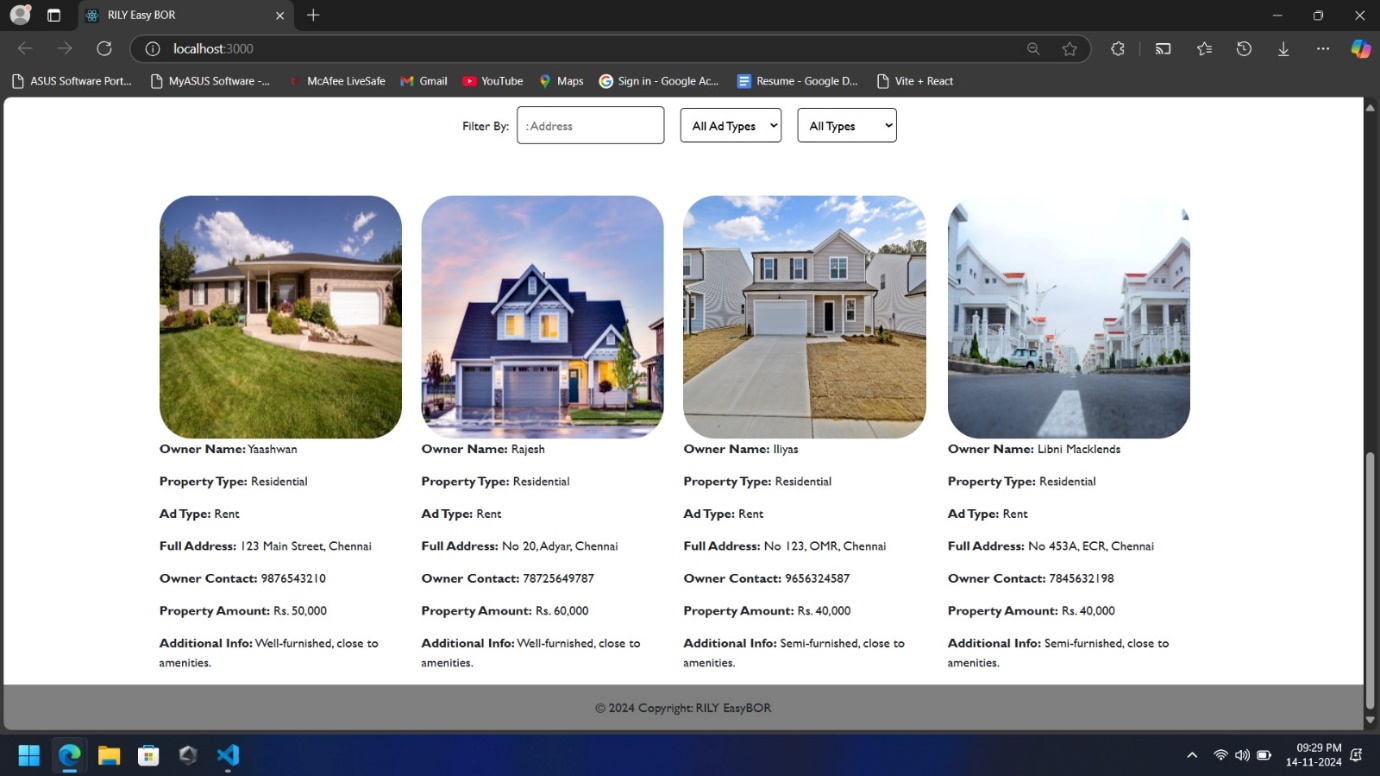
* Authentication is managed using JWT tokens. Upon login, a token is issued and stored on the client side to maintain session state. Middleware is implemented to verify tokens for protected routes, ensuring secure access to user-specific and sensitive data.

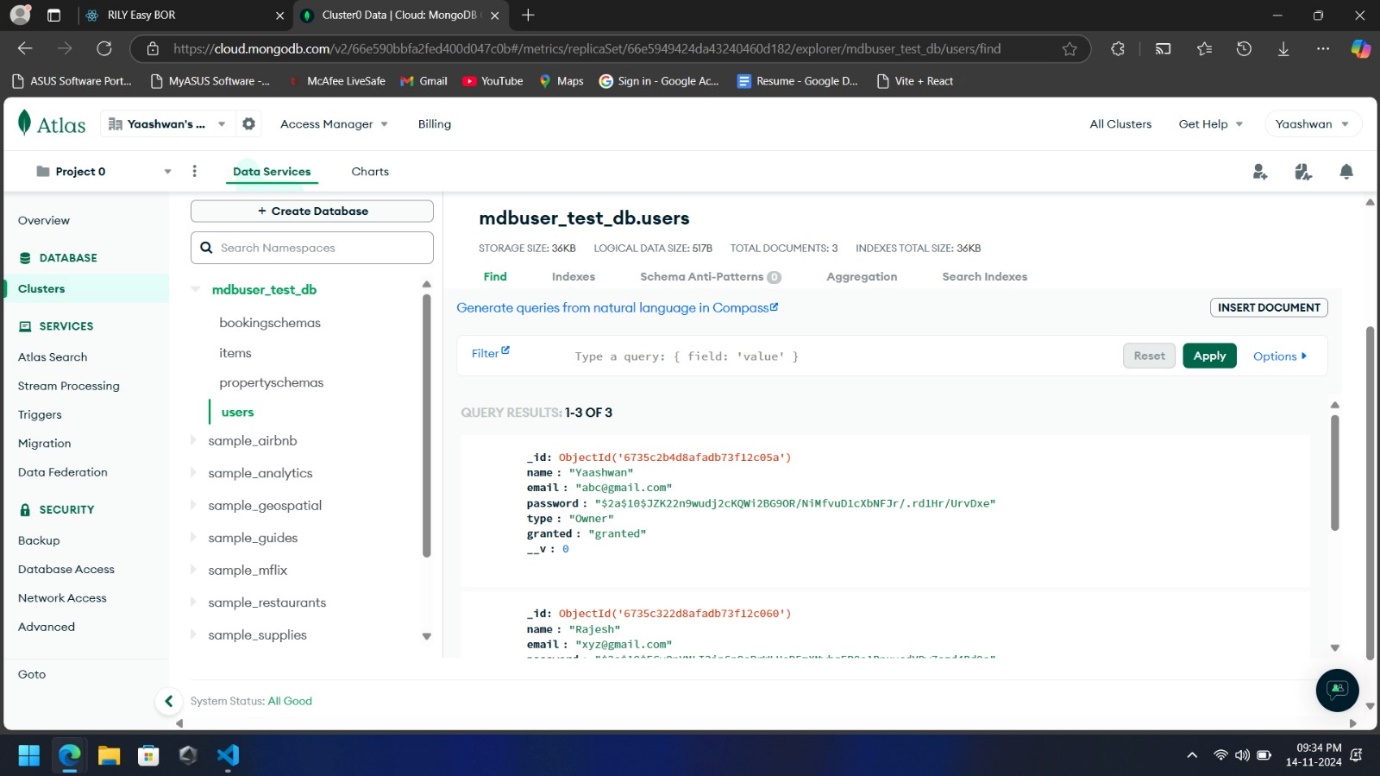
8. **Screenshots and Demo**











10. **Known Issues**

* Managing frontend state across components can become complex, and concurrent bookings may cause double bookings. To secure the app, it’s essential to protect against vulnerabilities like data exposure and injection attacks.

**11**. **Future Enhancements**

* **Real-Time Updates:** Implement WebSocket’s for live updates on property listings and booking statuses.
* **Advanced Search Filters:** Add filters for amenities, price range, and location-based suggestions**.**
* **Payment Integration:** Integrate payment gateways like Stripe or PayPal for seamless bookings**.**
* **Admin Dashboard:** Develop an admin panel for managing users, properties, and bookings**.**
* **Push Notifications:** Enable push notifications for booking confirmations and new property listings**.**