NM PROJECT REPORT

HOUSE RENT APP WITH MERN

TEAM 6560

TEAM MEMBERS:
DARSHAN B – 2021503505
RAJKUMAR M -2021503039
LOGESH S – 2021503519
VARUN D – 2021503567
KAILASH NATHAN S – 2021503315

Purpose

The purpose of this project is to build a house rental application that allows users (renters and property owners) to interact efficiently and securely. Renters can search for available properties, apply filters, contact landlords, and finalize rentals, while property owners can list properties, manage bookings, and review renter applications. The platform ensures a seamless rental process from discovery to move-in.

- Create an intuitive user experience for property discovery and renting.
- Provide landlords with a robust dashboard for property management.
- Ensure a secure transaction and application process with real-time notifications.
- Use a scalable and high-performance MERN stack architecture to accommodate large numbers of users and data.

Features

- **Property Listings**: Detailed listings with photos, descriptions, rent amount, location, and amenities.
- **Search Filters**: Filters for location, property type, price, number of rooms, and more.
- **User Registration and Profiles**: Registration for both renters and landlords, with user profiles.
- **Contact and Inquiry**: In-app messaging between renters and landlords.
- **Booking and Confirmation**: End-to-end booking process with notifications and confirmation.
- **Admin Controls**: Admin verification for landlord accounts, policy enforcement, and monitoring.
- **Transaction Management**: Built-in features for lease agreements, payment processing, and security.
- **Landlord Dashboard**: Features for landlords to add, edit, delete properties, and manage bookings.
- **Analytics for Landlords**: Insights into rental trends, occupancy rates, and earnings.

Architecture

Frontend

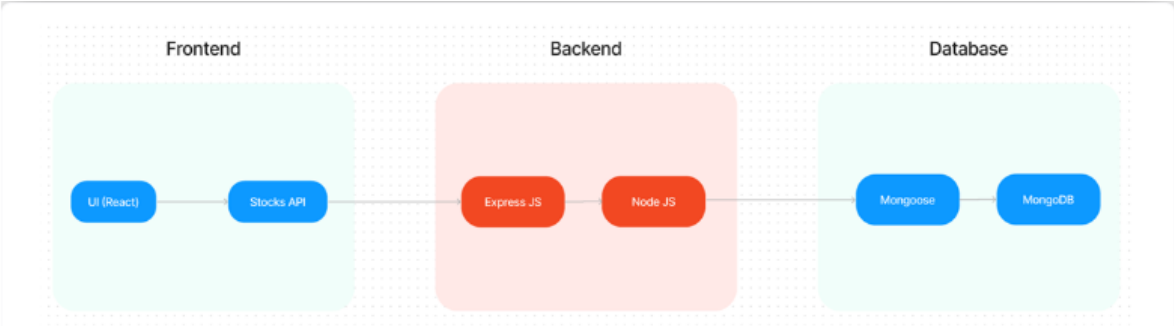
The frontend is built with React and utilizes Bootstrap and Material UI for styling and responsive design. The interface includes screens for renters to search and apply for properties, and landlords to manage their listings and view applications. Axios is used for API requests, ensuring smooth communication with the backend.

Backend

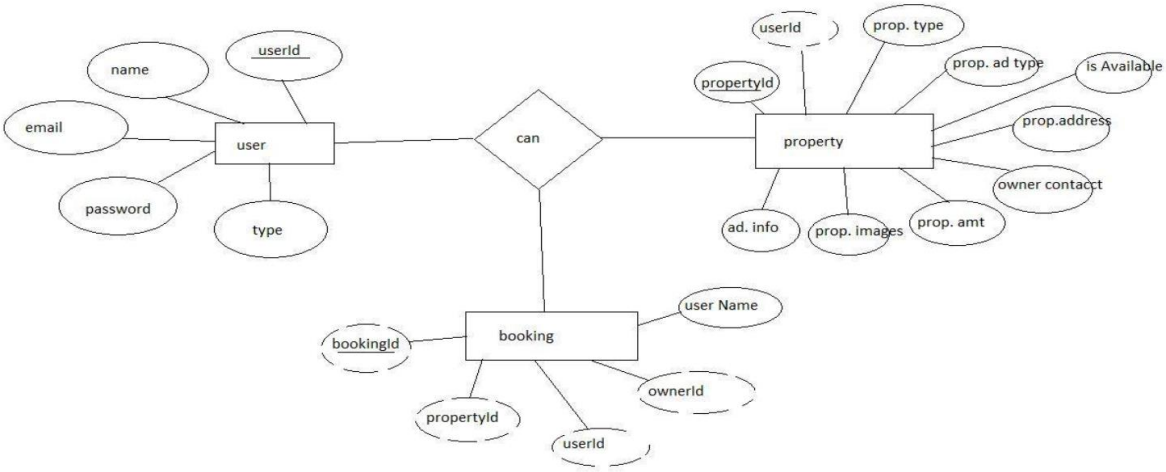
The backend is powered by Node.js and Express.js, handling server logic, API requests, and user authentication. The backend includes controllers for managing property listings, inquiries, user authentication, and admin approvals

Database

The database is MongoDB, chosen for its scalability and flexibility with document-based storage. MongoDB is ideal for handling user profiles, property listings, transactions, and messages between users.



ER-diagram



Setup Instructions

Prerequisites

- Node.js (version 14+)
- MongoDB (set up a local or remote MongoDB server)
- NPM (Node Package Manager)

Installation

- **Clone the Repository**: git clone <repository-url>
- **Navigate into Project Folder**: cd house-rent-app
- Install Dependencies.
- **Client**: cd client && npm install
- **Server**: cd server && npm install
- Set Environment Variables.
- Create a .env file in the server directory with values for MONGO_URI, JWT_SECRET, etc.

Authentication and Authorization

- **JWT (JSON Web Tokens):** Tokens are generated during user login and stored on the client side.
- **Sessions:** Sessions ensure that the user remains logged in between sessions.
- **Role-based Access Control**: Different access levels for renters, landlords, and admins

Folder Structure

Client

- **/src**: Contains all frontend files
    - **/components**: Reusable components (e.g., Navbar, Footer, Property Card)
    - **/pages**: Different pages for the app (Home, Property Details, Dashboard)
    - **/services**: Axios calls and API services
    - **/styles**: CSS or SCSS files for styling

Server

- **/controllers**: Handles business logic for routes (e.g., user authentication, property listing)
- **/models**: Defines MongoDB schemas (e.g., User, Property)
- **/routes**: Defines API endpoints (e.g., /auth, /properties)
- **/middleware**: Custom middleware (e.g., authentication, error handling)

Running the Application

Start the Frontend

- Navigate to the client directory: cd client
- Run the frontend server: npm start

Start the Backend

- Navigate to the server directory: cd server
- Run the backend server: npm start

Testing

- **Unit Testing**: Test components, services, and utility functions in both frontend and backend.
- **Integration Testing**: Test interaction between frontend components and backend APIs.
- **End-to-End Testing**: Use tools like **Jest**, **Mocha**, and **Cypress** for E2E testing of core functionalities (e.g., property search, user registration, messaging).

Known Issues

- **Map Integration**: Some location-based searches may have limited results without external API integration.
- **Notifications**: Push notifications may require browser permissions and may not work across all devices.

Future Enhancements

- **Enhanced Analytics**: Provide detailed insights for landlords on rental trends and tenant feedback.
- **Advanced Filters**: Add more filter options, such as pet-friendly properties, furnished/unfurnished, and lease duration.
- **Mobile App**: Develop a mobile version of the app for on-the-go property search and management.
- **Automated Lease Generation**: Enable automatic generation and signing of digital lease agreements within the app.
- **3D Property Tours**: Integrate 3D virtual tours for enhanced property viewing.

Conclusion

This house rent app provides a robust and streamlined platform that connects renters with property owners, simplifying the rental process from search to booking. Utilizing a MERN stack architecture, it ensures efficient data handling, secure transactions, and a user-friendly interface with advanced filtering and management features. This comprehensive setup not only addresses essential requirements like secure user authentication, admin oversight, and in-app messaging but also offers a scalable foundation for future enhancements and expanded functionality

References

Github Link: Click Now.

Demo Video Link: Click Now.