1. Introduction

Project Title: HouseRent App

Team Members:

- **Durgesh Kumar**: Backend Developer
- **Himanshu Kumar**: Frontend Developer
- **Alok Mandal**: Database Administrator
- **Chunchun Kumar**: Tester

---

2. Project Overview

Purpose

The *Home Rent* project is designed to simplify the process of renting properties by connecting landlords and tenants on a single platform. It offers an intuitive interface for property management and rental operations.

Features

- User registration and login (tenants and landlords).
- Property listing and search functionality.
- Rent payment tracking and history.
- Reviews and feedback for properties.
- Notifications and alerts for new listings and updates.

---

3. Architecture

Frontend

The frontend is built using **React**, ensuring a responsive and interactive UI. It leverages React Router for navigation and Redux for state management.

Backend

The backend uses **Node.js** and **Express.js**, offering RESTful APIs to handle user requests and interactions with the database. Middleware is used for authentication, validation, and error handling.

Database

The project uses **MongoDB** as the database to store user data, property listings, transaction history, and feedback. **Mongoose** is used for schema management and database interaction.

---

4. Setup Instructions

Prerequisites

- **Node.js**: v16 or above
- **MongoDB**: v5.0 or above
- **npm**: v8 or above
- **Java Development Kit (JDK)**: v11

Installation

1. Clone the repository:

git repository: *https://github.com/durgeshkumarraj/houserentapp-naan-mudhalvan*

2. Navigate to the project directories:
    - cd client for the frontend.
    - cd server for the backend.
3. Install dependencies:

npm install

4. Set up environment variables:
    - Create a .env file in the server directory.
    - Add values for DB_URI, JWT_SECRET, etc.

---

5. Folder Structure

my-project/

├── client/

| ├── public/

| ├── src/

| ├── package.json

| └── .gitignore

├── server/

| ├── models/

| ├── routes/

| ├── controllers/

| ├── server.js

| └── package.json

└── .gitignore

---

6. Running the Application

Start the Frontend

cd client

npm start

Start the Backend

cd server

nodemon index.js

---

7. API Documentation

Endpoint	Method	Description	Parameters	Example Response
/api/auth/login	POST	Login a user	{email, password}	{token: "abc123"}
/api/properties	GET	Fetch all properties	query: {location, type}	[{id: 1, title: "Apartment"}, ...]
/api/rent/pay	POST	Process rent payment	{propertyId, amount}	{status: "success", receiptId: "123"}

---

8. Authentication

Authentication is handled using **JWT tokens**. Tokens are issued upon successful login and stored in the client's local storage. Authorization middleware validates tokens for protected routes.

---

9. User Interface

The application features:

- **Dashboard:** Overview of active listings and rentals.
- **Search Page:** Filters for location, price, and property type.
- **Listing Form:** Landlords can upload property details.

Screenshots:

1. Login Page
2. Dashboard
3. Property Details Page

---

10. Testing

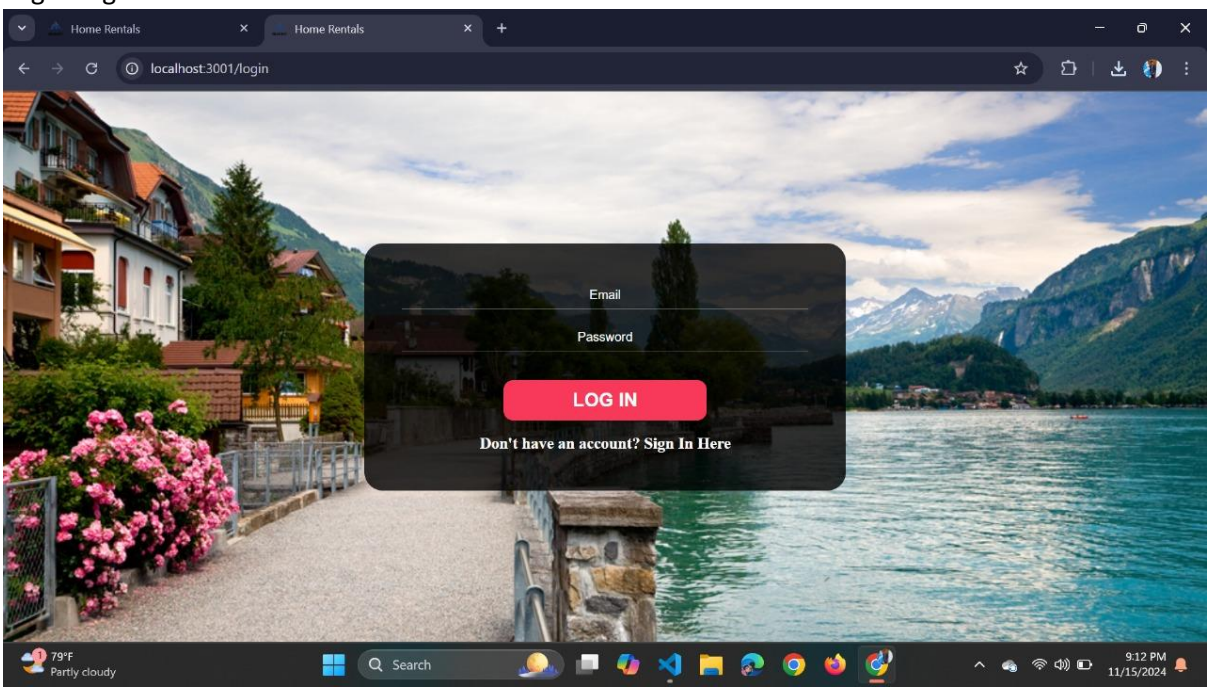
Testing Strategy

- Unit testing using **Jest** for backend APIs.
- Integration testing using **Supertest**.
- Frontend UI testing using **React Testing Library**.

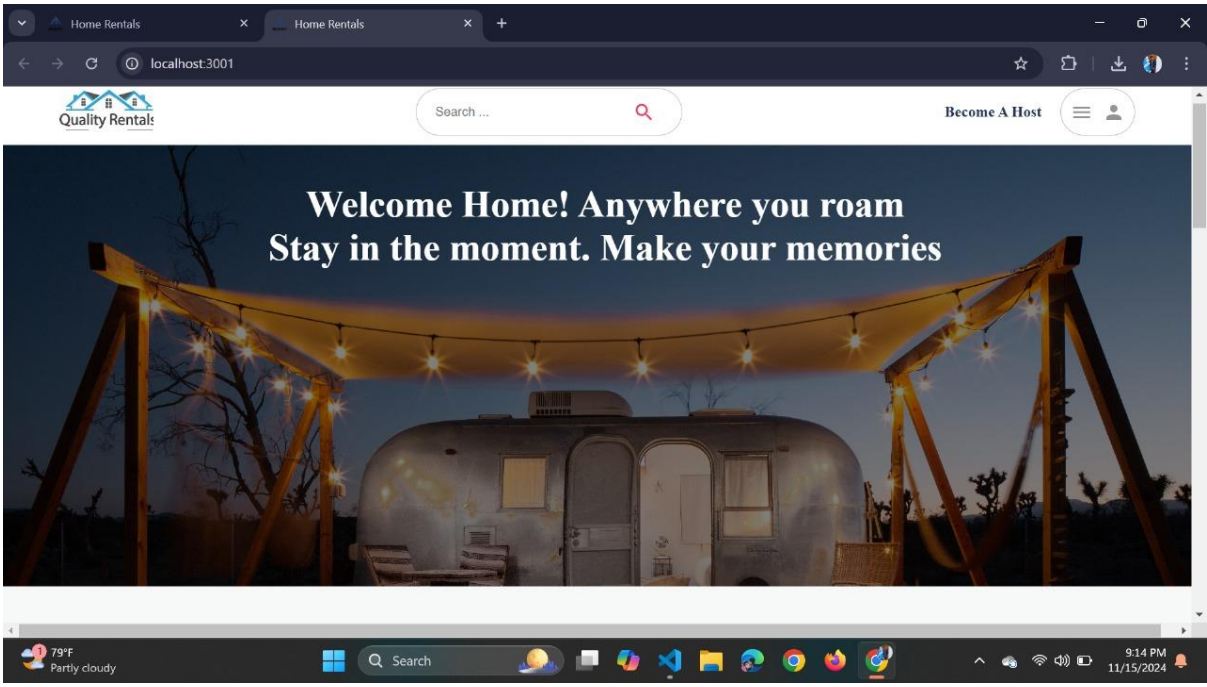
---

11. Screenshots or Demo

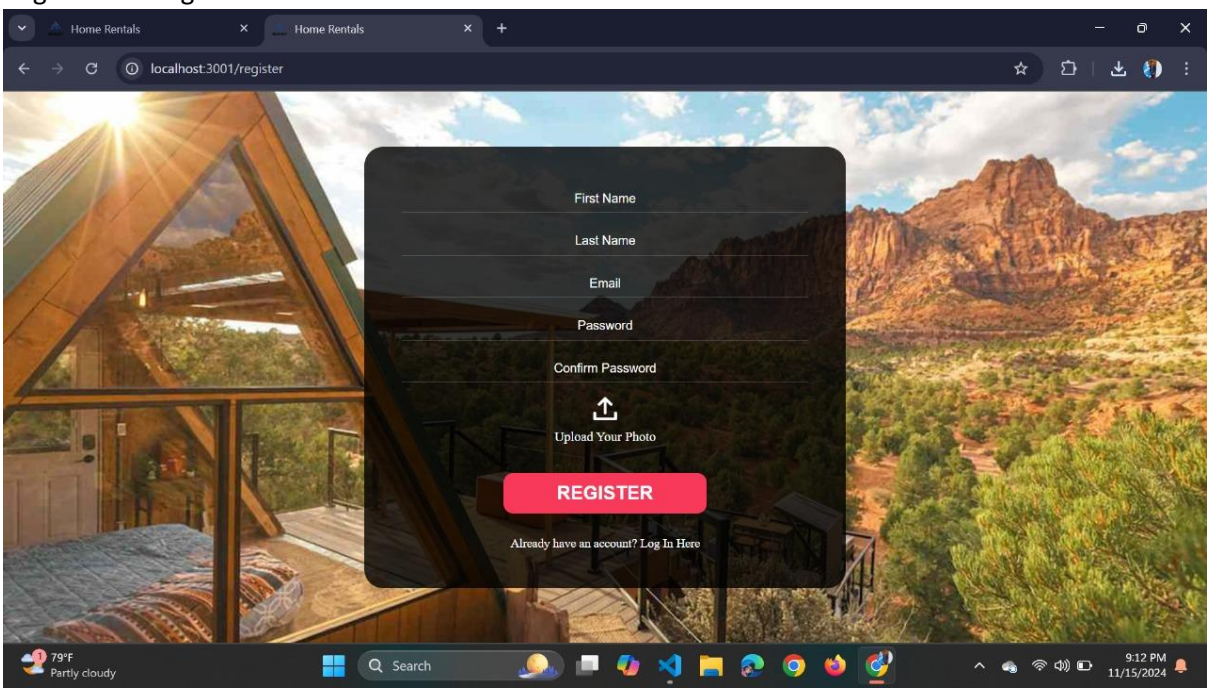
- Demo link: https://drive.google.com/file/d/1biN_fh08MAAA-ZITkwoahZMWTRmtJFI7/view?usp=drivesdk
- Screenshots:
    - Login Page:



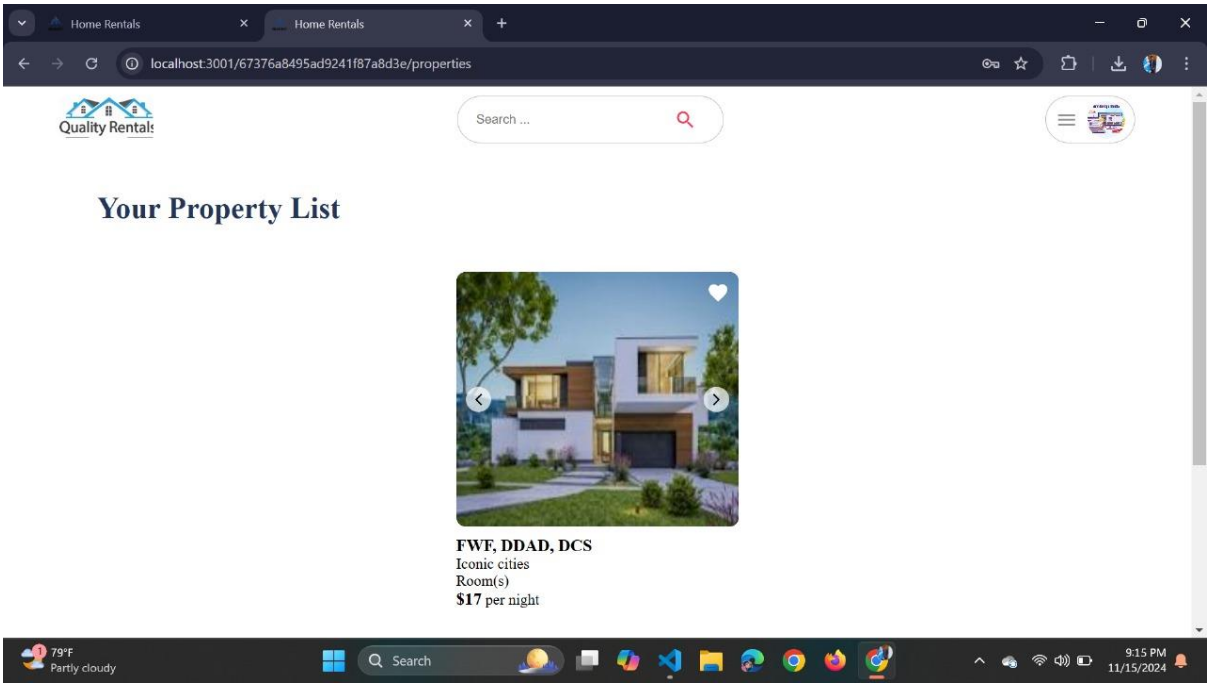
- Home Page:



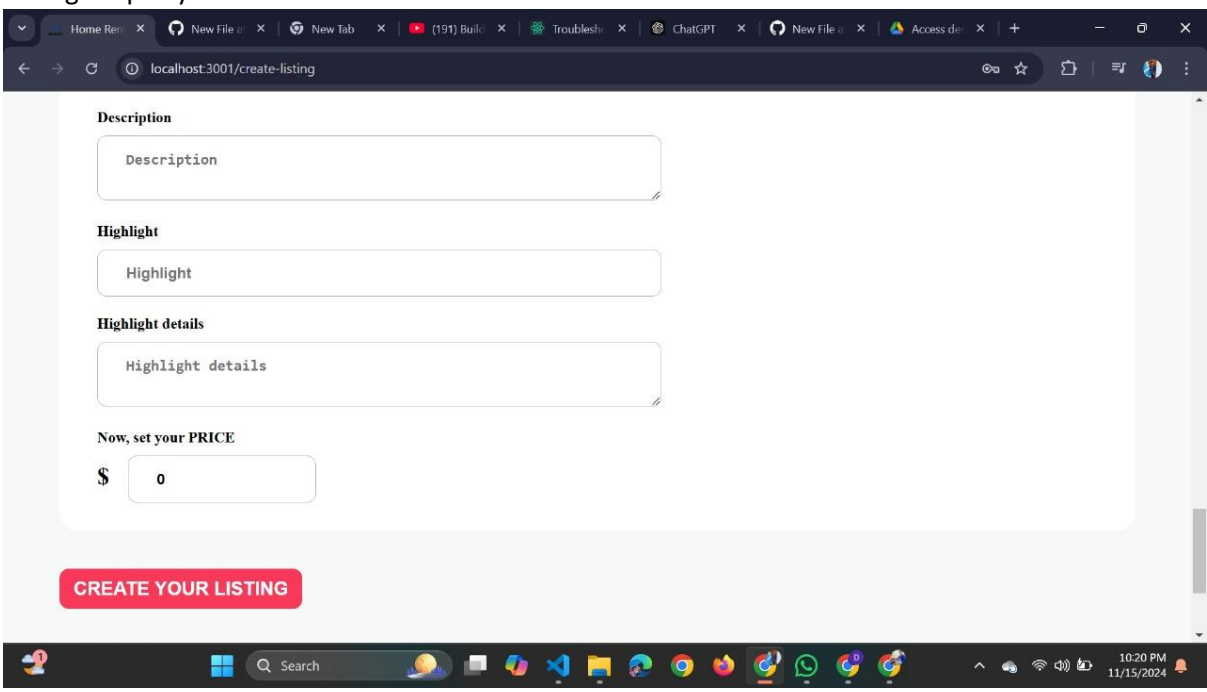
- Registration Page:



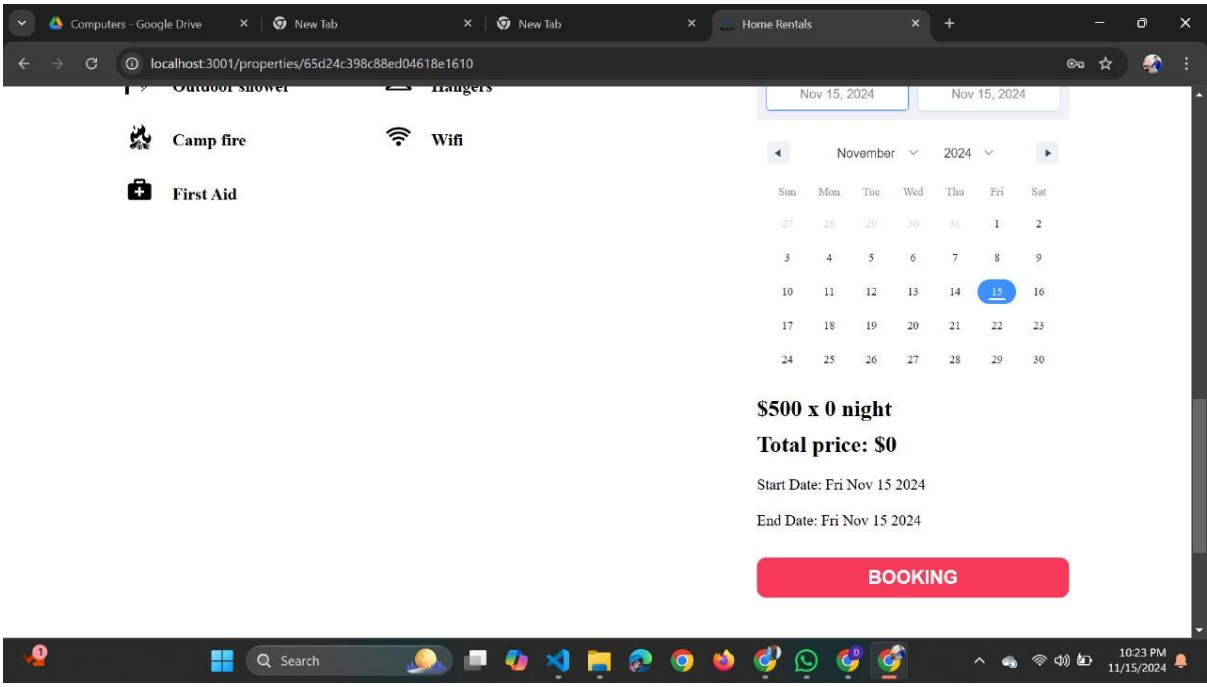
- Property List:



- Listing Property:



- Booking Property:



---

12. Known Issues

- **Network Issue**: Network Issue may create some problem at some time in accessing Database.
- **Pagination for property listings**: May experience delays for large datasets.
- **Notification system**: Limited support for real-time updates.

---

13. Future Enhancements

- Implement real-time chat between landlords and tenants.
- Add support for multiple payment gateways.
- Introduce AI-based property recommendations.