

HOUSE RENT APPLICATION

1. INTRODUCTION

Project Title : House Rent Application(Rentify)

Team Members

1. Devendran M (Backend Development)
2. Ragulan S (Frontend Development)
3. Asvini Rajan SS (Database Management)
4. Akash YR (User Authentication)
5. Sajan M (Testing)

2. PROJECT OVERVIEW

Purpose:

The House Rent Application aims to provide a seamless online platform for tenants and property owners to interact effectively. By leveraging the MERN (MongoDB, Express.js, React, Node.js) stack, the application offers an intuitive, secure, and responsive experience for users to explore rental properties, manage bookings, and enjoy a hassle-free renting journey. The goal is to replicate the efficiency of traditional property renting processes while making them accessible from anywhere, at any time.

Features:

User Registration and Authentication:

Users can sign up, log in, and authenticate their identity securely. This allows different roles (renter/tenant, owner, admin) to access their respective features with proper authorization.

Property Listings & Search:

Display a wide selection of properties for rent with sorting and filtering options (by city, rent amount, property type, etc.). Users can search based on location, price, and other relevant criteria.

Property Selection:

Allows users to view detailed information about each property, including address, city, rent amount, property type, landlord's name and contact, property description, location link, and images.

Booking and Reservation Management:

Users (renters/tenants) can book properties and check availability. Property owners can manage bookings, approve or reject requests, and track reservations.

CRUD Operations for Property Owners:

Owners can perform Create, Read, Update, and Delete operations for their properties, allowing them to manage their listings efficiently. This includes editing property details, uploading new property images, and managing their rental offers.

Responsive Design:

The application is fully responsive, ensuring an optimal experience across devices (desktop, tablet, mobile).

3. ARCHITECTURE

The architecture of the House Rent Application is designed to provide a seamless user experience by efficiently integrating the frontend, backend, and database. The application leverages the MERN stack (MongoDB, Express.js, React.js, and Node.js) to deliver a scalable, secure, and user-friendly solution.

Frontend (React)

- **Components:** The frontend is developed using React.js, which includes reusable components like property listings, property details, booking forms, user profile, Tenant Details, Owner Details and Admin Dashboards.
- **UI Framework:** Preline UI is used to create a visually appealing, consistent, and responsive interface, simplifying development with prebuilt components.
- **State Management:** React's internal state management is used, with tools like Context API to handle global states (e.g., user authentication status, booking data).

- **Routing:** React Router is used for handling navigation between pages (e.g., home, property details, booking history).

Backend (Node.js and Express.js)

- **Express.js Server:** The backend is powered by Node.js and Express.js, which serve the necessary APIs to handle client requests, including fetching property data, processing bookings, and handling user authentication.
- **RESTful API:** The backend exposes RESTful APIs for: Managing books (CRUD operations)
- Managing user data (signup, login, authentication).

Database (MongoDB)

- **Schema:** The database is designed using MongoDB, which stores: Properties: Address, city, rent amount, property type, description, landlord information (name, email), location link, property image, posted date.
- **Users:** User profile information, role (tenant, owner, admin), authentication credentials (hashed passwords).
- **Bookings:** Booking details, user details, property information, booking date, and status.
- A User can have multiple Bookings. Each Booking corresponds to a Property.

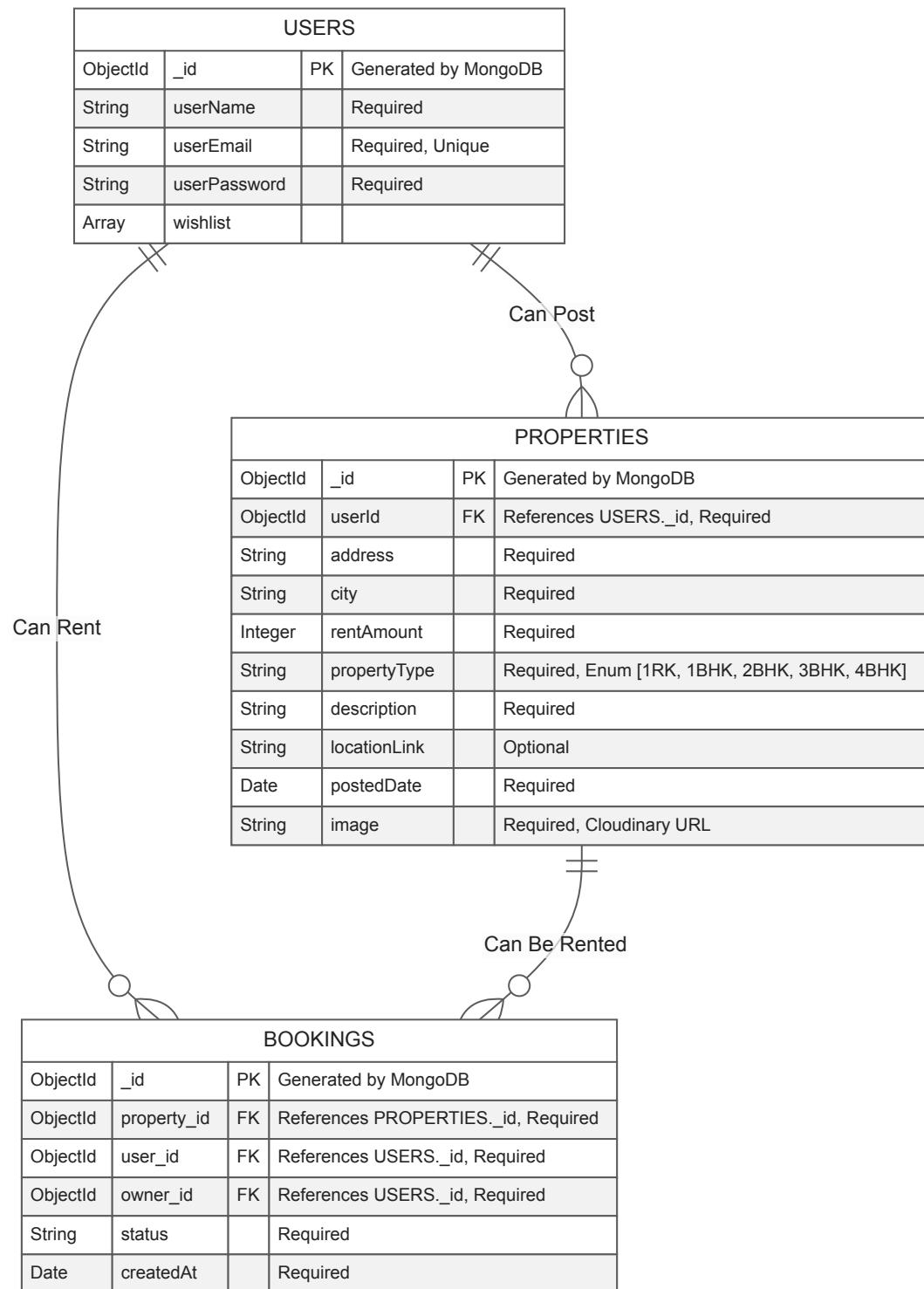
Interactions:

The frontend communicates with the backend through HTTP requests handled by Axios, a promise-based HTTP client. Backend APIs query and update the MongoDB database as needed. For example:

- When a tenant searches for properties, the backend retrieves matching property records from MongoDB and sends them to the frontend.
- When a property owner updates a listing, the backend modifies the corresponding record in MongoDB.
- When a booking request is submitted, the backend creates a new booking entry, associating it with the respective user and property.

By combining React.js, Node.js, Express.js, and MongoDB, the architecture of the House Rent Application ensures a robust, scalable, and maintainable system. This architecture supports real-time data updates, secure user interactions, and efficient property and booking management.

ER Diagram



4. SETUP INSTRUCTIONS

Prerequisites

Before setting up the House Rent application, ensure you have the following installed:

1. Node.js (v14 or above)
2. MongoDB (Local or Remote instance, e.g., MongoDB Atlas)
3. npm (Node Package Manager)

Installation

1. Clone the Repository

- Start by cloning the repository to your local machine:

```
git clone https://github.com/devendran-odin/HouseRent.git
```

2. Backend Setup

- Navigate to the Backend directory:

```
cd Backend
```

- Install the required dependencies:

```
npm install
```

- Start the backend server

```
npm start
```

- The backend should now be running on <http://localhost:5000> (default port).

3. Frontend Setup

- Navigate to the Frontend/HouseRent-App directory:

```
cd Frontend/HouseRent-App
```

- Install the required dependencies:

```
npm install
```

- Start the frontend server:

```
npm run dev
```

The frontend will now be available at <http://localhost:5173>

4. Environment Variables

In Backed Root Folder Setup an .env file

```
PORT=5000  
  
MONGO_URI=<mongo connection string>  
  
ADMIN_EMAIL=rentifyadmin@gmail.com  
ADMIN_PASSWORD=admin@123  
  
JWT_SECRET_KEY=<jwt secret key>  
SENDGRID_API_KEY=<sendgrid_api_key>  
CLOUDINARY_CLOUD_NAME=<cloudinary_cloud_name>  
CLOUDINARY_API_KEY=<cloudinary_api_key>  
CLOUDINARY_API_SECRET=<cloudinary_api_secret>
```

In Frontend/HouseRent-App Folder Setup an .env file

```
VITE_BACKEND_URL=http://localhost:5000
```

5. FOLDER STRUCTURE

Client (Frontend)

- Frontend/HouseRent-App
- src/
- components/: Contains shared React components like **Footer**, **Home**, **PropertyCard**, etc.
- Pages/: Contains pages specific to renters/tenants such as **Home**, **Property Listings**, **Booking History**.
- Contains pages for admin users, including **Manage Users**, **Manage Properties**, and **Admin Dashboard**.
- Pages/: Contains pages for property owners, such as **Add Property**, **My Listings**, and **Manage Bookings**.
- public/: Contains static assets like **images**, **icons**, and **favicon**.
- App.jsx: Main component that handles routing and rendering the application.
- main.js: Entry point for the React application.

Server (Backend)

- Backend/
- Contains **Mongoose models** for **User**, **Property**, and **Booking schemas**.
- routes/: Contains API route definitions for managing properties, bookings, and users.
- Contains configuration files for database connection, environment variables, and server settings.
- server.js: Main entry point that sets up **Express**, connects to MongoDB, and includes controllers to handle incoming API requests.

This folder structure ensures the project is organized and modular, making it easy to scale and maintain.

The screenshot shows a file explorer interface with a dark theme. At the top, there are four small icons: a document, a folder, a circular arrow, and a square. Below these are two blue circular dots.

The project structure is as follows:

- HOUSERENT**
 - Backend**
 - controllers
 - middleware
 - models
 - node_modules
 - routes
 - uploads
 - .env
 - .gitignore
 - package-lock.json
 - package.json
 - server.js
- Frontend/HouseRent-App**
 - node_modules
 - public
 - src**
 - assets
 - Components
 - context
 - Pages**
 - App.jsx
 - index.css
 - main.jsx
 - .env
 - .gitignore
 - eslint.config.js
 - index.html
 - package-lock.json
 - package.json
 - postcss.config.js
 - tailwind.config.js
 - vite.config.js

At the bottom of the list is a blue circular dot.

At the very bottom of the screen, there is a dark bar containing a small icon and some text.

6. RUNNING THE APPLICATION

Frontend:

Navigate to the `Frontend/HouseRent-App` directory and run:

```
npm run dev
```

Backend:

Navigate to the `Backend` directory and run:

```
npm start
```

Ensure MongoDB is running locally or use a cloudbased MongoDB instance (e.g., MongoDB Atlas)

7. API DOCUMENTATION

1. User APIs

a) User Login

- Endpoint: `POST /api/users/login`
- Request Body:

```
{
  "userEmail": "devsanthosh720@gmail.com",
  "userPassword": "12345678"
}
```

- Response:

```
{  
  "message": "Login successful",  
  "userId": "6739b3782b49bc07690c0ffd",  
  "token": "token"  
}
```

b) User Register

- Endpoint: POST /api/users/signup
- Request Body:

```
{  
  "username": "your_test_username",  
  "useremail": "test@example.com",  
  "password": "your_test_password"  
}
```

- Response:

```
User registered successfully!
```

c) Get All Users

- Endpoint: GET /api/users/getAllUsers
- Response:

```
[  
  {  
  
    "id": "userId",  
    "userName": "User Name",  
    "userEmail": "user@example.com"  
  
  },  
  {  
  
    "id": "userId",  
    "userName": "User Name",  
    "userEmail": "user@example.com"  
  
  },  
]
```

d) Delete User

- Endpoint: DELETE /api/users/:id
- Response:
- Status 200 : User deleted successfully
- Status 500 : Internal server error

2. Property APIs

a) Get All Property

- **Endpoint:** GET /api/properties/getAllProperty
- Response:

```
[  
  {  
  
    "_id": "673c82bb5f57113c540e37ac",  
    "userId": "6739b3782b49bc07690c0ffd",  
    "address": "No 123, ABC Street, Chennai-34",  
    "city": "Chennai",  
    "rentAmount": 13500,  
    "propertyType": "2BHK",  
    "description": "A Spacious 2BHK Home Near the Abc School",  
    "locationLink": "https://maps.app.goo.gl/h3QtpmgTTUXeRG4z8",  
    "postedDate": "2024-11-19T00:00:00.000Z",  
    "image":  
      "https://res.cloudinary.com/dvtznimfk/image/upload/v1732018870/a1dzfn6ojyg  
      v6vutfmex.jpg",  
      "__v": 0  
  
  },  
  
  {  
  
    "_id": "673c843c5f57113c540e37bb",  
    "userId": "6739b3782b49bc07690c0ffd",  
    "address": "Door no 12, Second floor, Abc Apartments, Mumbai",  
    "city": "Mumbai",  
    "rentAmount": 24000,  
    "propertyType": "3BHK",  
    "description": "Stunning 3BHK with Premium Amenities at Abc  
    Apartments, Mumbai",  
    "locationLink": "https://maps.app.goo.gl/SfPk66gYAmWZqZTy6",  
    "postedDate": "2024-11-18T00:00:00.000Z",  
    "image":  
      "https://res.cloudinary.com/dvtznimfk/image/upload/v1732018870/a1dzfn6ojyg  
      v6vutfmex.jpg",  
      "__v": 0  
  }]
```

```
"https://res.cloudinary.com/dvt nzimfk/image/upload/v1732019256/ig0clmcwdqxjl54onypt.avif",
    "__v": 0

},
]
```

b) Get Property by ID

- Endpoint:

```
GET api/properties/:id
```

- Response:

```
{
  "_id": "673c843c5f57113c540e37bb",
  "userId": "6739b3782b49bc07690c0ffd",
  "address": "Door no 12, Second floor, Abc Apartments, Mumbai",
  "city": "Mumbai",
  "rentAmount": 24000,
  "propertyType": "3BHK",
  "description": "Stunning 3BHK with Premium Amenities at Abc Apartments, Mumbai",
  "locationLink": "https://maps.app.goo.gl/SfPk66gYAmWZqZTy6",
  "postedDate": "2024-11-18T00:00:00.000Z",
  "image": "https://res.cloudinary.com/dvt nzimfk/image/upload/v1732019256/ig0clmcwdqxjl54onypt.avif",
  "__v": 0
}
```

c) Add Property

- Endpoint:

```
POST /api/properties/addProperty
```

- Request Body:

```
{
  "userId": "6739b3782b49bc07690c0ffd",
  "address": "Door no 12, Second floor, Abc Apartments, Mumbai",
```

```
"city":"Mumbai",
"rentAmount":24000,
"propertyType":"3BHK",
"description":"Stunning 3BHK with Premium Amenities at Abc Apartments, Mumbai",
"locationLink":"https://maps.app.goo.gl/SfPk66gYAmWZqZTy6",
"postedDate":"2024-11-18T00:00:00.000Z",
"image":"https://res.cloudinary.com/dvtznimfk/image/upload/v1732019256/ig0clmwdqxjl54onypt.jpg",
}
```

- Response:

“successfully added”

d) Delete Property

- Endpoint:

DELETE /api/properties/:id

- Response:
- Status 200 : Property deleted successfully
- Status 500 : Internal server error

3. Wishlist APIs

a) Add Item to Wishlist

- Endpoint:

POST /api/wishlist/add

- Request Body:

{

```
"propertyId": "propertyId",
"user": "userId",
}
```

- Response:

```
"Property added to wishlist"
```

b) Remove Item from Wishlist

- Need to send the token in header to verify the user
- Endpoint:

```
DELETE /api/wishlist/:property_id
```

- Request Body:

```
{
  "propertyId": "propertyId",
  "userId": "userId",
}
```

- Response:

```
"Property removed from wishlist"
```

c) Get all Wishlist for specific user

- Need to send the token in header to verify the user
- Endpoint:

```
GET /api/wishlist/getAll
```

- Response

```
[
  {
    "_id": "673c8eb1f2a8fb095d4d9dda",
    "userId": {
      "_id": "6739e39cf0904df870f3c90f",
      "userName": "Akash",
      "userEmail": "akash@gmail.com"
    },
    "address": "No 98, RR st, Chennai-45",
    "city": "Chennai",
    "rentAmount": 4500,
    "propertyType": "1RK",
    "description": "1RK near the Marina Beach",
    "locationLink": "https://maps.app.goo.gl/Ly3rNsRaozXcTf3z7",
    "postedDate": "2024-10-29T00:00:00.000Z",

    "image": "https://res.cloudinary.com/dvtnzimfk/image/upload/v1732021932/glv
0c8wk5dlq3ghaif91.png",
    "__v": 0
  },
]
]
```

4. Filter API

a) Get the Properties based on Filters

- Endpoint:

```
POST /api/properties/filters
```

- Request Body

```
{
  "bhk": "2BHK",
  "location": "Chennai",
  "priceRange": "10001-15000",
}
```

- Response

```
[
  {
    "_id": "673c82bb5f57113c540e37ac",
    "userId": "6739b3782b49bc07690c0ffd",
```

```
        "address": "No 123, ABC Street, Chennai-34",
        "city": "Chennai",
        "rentAmount": 13500,
        "propertyType": "2BHK",
        "description": "A Spacious 2BHK Home Near the Abc School",
        "locationLink": "https://maps.app.goo.gl/h3QtpmggtUXeRG4z8",
        "postedDate": "2024-11-19T00:00:00.000Z",
        "image":
      "https://res.cloudinary.com/dvtnzimfk/image/upload/v1732018870/a1dzfn6ojyg
      v6vutfmex.jpg",
        "__v": 0
    }
]
```

5. Booking APIs

a) Requesting to Book an Property

- Endpoint:

```
POST /api/bookings/request
```

- Request Body

```
{
  propertyId: "property_id",
  requesterId: "user_id"
}
```

- Response

```
Booking request sent successfully.
```

b) Get all the Requested Bookings

- Endpoint
- Need to send the auth token in header

```
POST /api/bookings/getAllRequest
```

- Response

```
[  
  {  
    "_id": "6742c044e79b6ae751219a99",  
    "owner": {  
      "userName": "Akash",  
      "userEmail": "akash@gmail.com"  
    },  
    "propertyId": "673c8eb1f2a8fb095d4d9dda",  
    "createdAt": "2024-11-24T05:57:24.845Z",  
    "status": "Pending"  
  },  
]  
]
```

c) Get all the Pending Approvals

- Endpoint
- Need to send auth token in header

GET /api/bookings/approvals

- Response

```
[  
  {  
    "_id": "674810b8b4125ec8f882163e",  
    "tenantId": "6739b7a988be4343ce9916a3",  
    "tenantName": "Arun",  
    "tenantEmail": "arun@gmail.com",  
    "propertyId": "673c8eb1f2a8fb095d4d9dda",  
    "requestedDate": "2024-11-28T06:42:00.977Z"  
  },  
]  
]
```

8. AUTHENTICATION

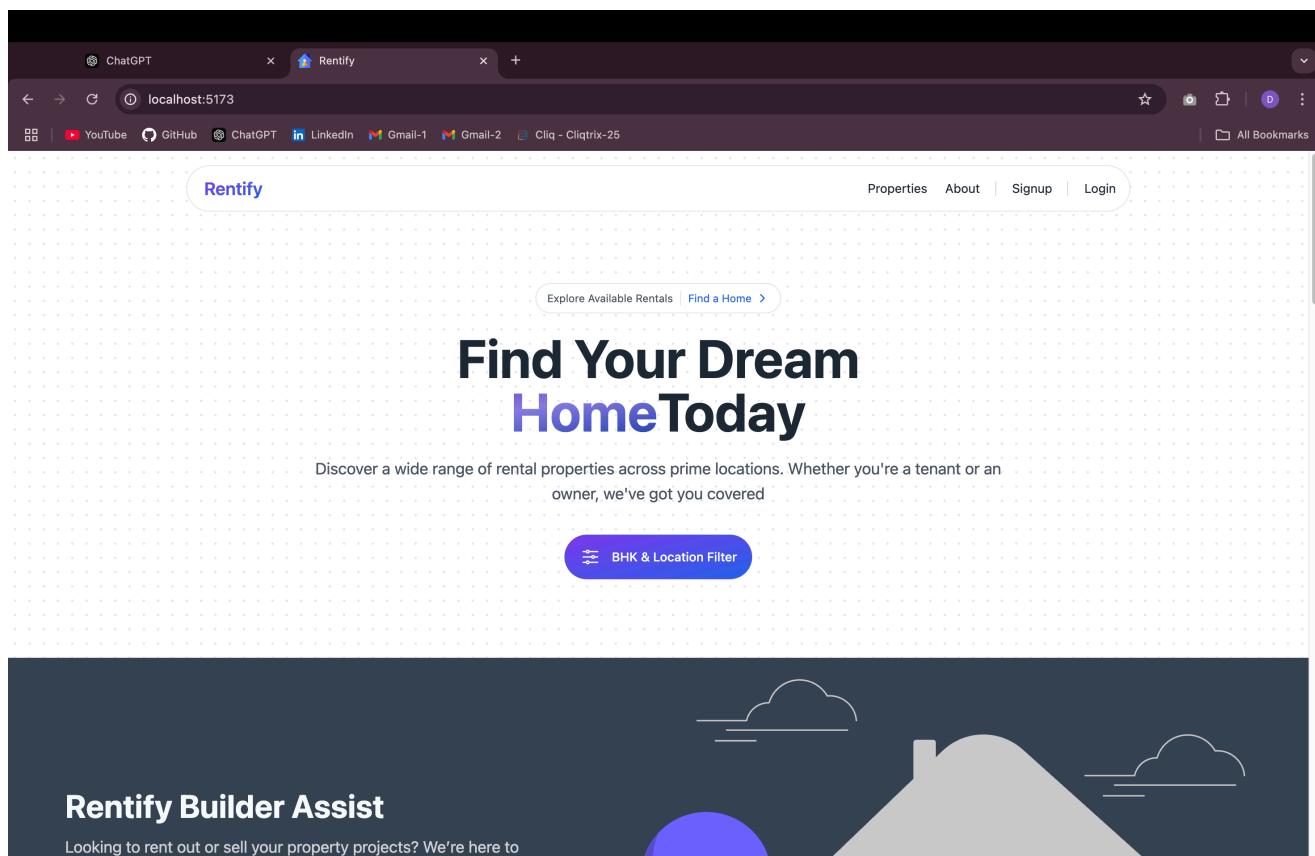
- **User and Admin Authentication:**
 - The code implements authentication via login and registration endpoints for both users and admins.
 - Credentials are verified against the database during login.

- The system ensures unique email addresses during registration to avoid duplication.

• **Secure Access Management:**

- Successful authentication grants users access to their respective accounts.
- Unauthorized access is prevented by validating user identities.
- User sessions are managed securely using **JWT (JSON Web Tokens)**, ensuring safe and persistent access.
- JWTs are issued upon successful login and stored in the user's session, allowing secure, stateless communication between the client and server.
- The JWT is used for subsequent API requests to validate the user's identity and permissions.

9. USER INTERFACE



localhost:5173

Rentify

Discover a wide range of properties for rent or sale.

Filter Preferences

Refine your search based on your requirements

BHK Type: 2 BHK

Location: Bengaluru

Price Range: ₹10,001 - ₹15,000

Cancel Apply Filters

Rentify Builder Assist

Looking to rent out or sell your property projects? We're here to help!

localhost:5173/# work email Request demo

localhost:5173/properties

Rentify

PROPERTIES POSTED

See All



₹13,500 Per Month
A Spacious 2BHK Home Near the ABC School

Add to Favorites Request to Book



₹24,000 Per Month
Stunning 3BHK with Premium Amenities at ABC Apartments, Mumbai

Add to Favorites Request to Book



₹9,800 Per Month
1BHK Home Surrounded by Nature in Kerala

Add to Favorites Request to Book



₹6,800 Per Month
1RK Home Near Connaught Place, Delhi

Add to Favorites Request to Book



₹28,000 Per Month
Spacious 4BHK Home Near IT Parks in Bangalore

Add to Favorites Request to Book



₹4,500 Per Month
1RK near the Marina Beach

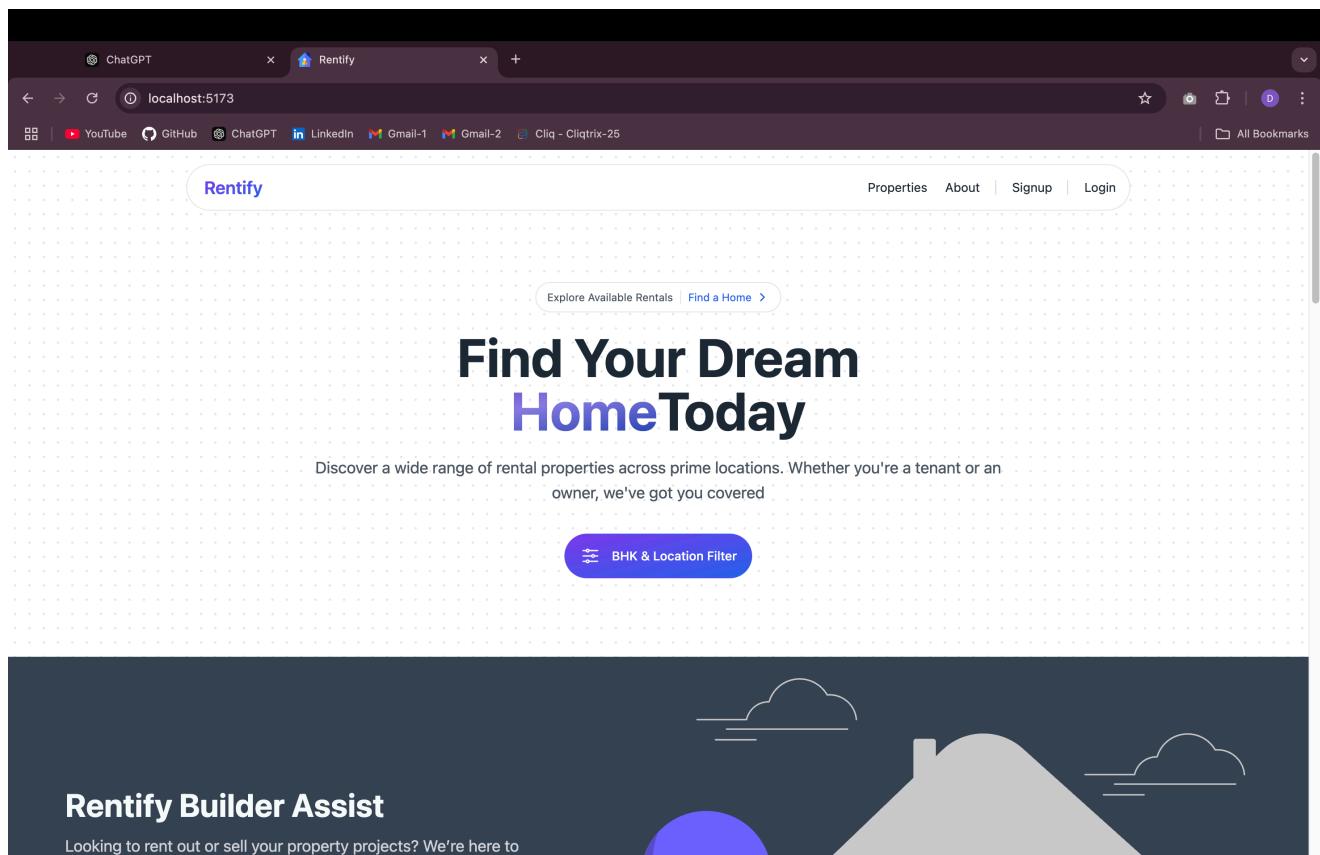
Add to Favorites Request to Book

10. TESTING

Tools Used: Manual testing, Postman for API testing, and user experience testing for interface usability.

- **API Testing:** Verifying correct API responses, error handling, and CRUD operations (user registration, property listings, bookings).
- **User Authentication:** Testing login, registration, and session management for secure access to user data.
- **Authorization Testing:** Ensuring proper access control for users, admins, and property owners based on roles.
- **Frontend Testing:** Verifying React components render correctly and respond to user actions (search, booking, etc.).
- **Responsive Design:** Testing frontend behavior across different devices and screen sizes for a smooth user experience.
- **Integration Testing:** Ensuring frontend and backend integration works seamlessly, handling edge cases and invalid inputs.

11. SCREENSHOTS



Rentify

Properties | About | Signup | Login

Find Your Home
Discover a wide range of rental properties tailored to meet your preferences and budget. Whether you're looking for an apartment or a house, we've got you covered.

Owner Verification
All property owners are verified before listing their homes to ensure safety and reliability. This process guarantees that you're dealing with legitimate owners.

Bookmark Favorites
Save your favorite listings to your profile for easy access later. Keep track of properties you love without needing to search again.

Easy Booking
Our straightforward booking system allows you to schedule viewings and make inquiries with just a few clicks, streamlining your search for a new home.

24/7 Support
Our dedicated customer support team is available around the clock to assist you with any questions or concerns, ensuring a seamless renting experience.

User-Friendly
Enjoy a clean and intuitive interface designed for easy navigation. Find what you need quickly and efficiently, whether on desktop or mobile..

Rentify

Properties | About | **Signup** | Login

Sign up

Already have an account? [Sign in](#)

Username

Email address

Password

Confirm Password

Sign up

Sign up successful!

Rentify

Properties A Login successful!

Sign in

Don't have an account yet? [Sign up](#)

Email address
ajay@gmail.com

Password
.....

Sign in

As a Tenant

Rentify

Properties About [Profile](#)

Ajay
ajay@gmail.com

As a Tenant → As a Owner → Logout →

Requested Favorites

OWNER DETAILS	REQUESTED DATE	STATUS	VIEW PROPERTY
Christina Bersh christina@site.com	10 Oct 2024	Accepted	More Info
David Harrison david@site.com	04 Aug 2024	Rejected	More Info
Anne Richard anne@site.com	July 10 2024	Pending	More Info

As a Owner

The screenshot shows the Rentify web application interface for an owner. At the top, there's a header bar with tabs for 'Properties', 'About', and 'Profile'. Below the header is a navigation bar with links for 'Ajay' and 'ajay@gmail.com', and buttons for 'As a Tenant →', 'As a Owner →' (which is highlighted with a green border), and 'Logout →'. The main content area has three tabs: 'My Properties' (selected), 'Pending Approvals', and 'Add New Property'. Below these tabs is a table listing three properties:

PROPERTY ID	POSTED DATE	VIEW PROPERTY	ACTION	STATUS
#7854321	10 Jun, 2024	More Info	Edit	Available
#7854321	10 Jun, 2024	More Info	Edit	Occupied
#7854321	10 Jun, 2024	More Info	Edit	Pending

The screenshot shows the 'Post a New Property' form. At the top, there are tabs for 'My Properties', 'Pending Approvals', and 'Add New Property' (which is highlighted with a blue border). Below the tabs is a title 'Post a New Property' and a subtitle 'Fill in the details below to list your property for rent.' The form consists of several input fields:

- Address:** e.g., No 123, ABC Street, Chennai
- City:** Select City
- Rent Amount (in ₹):** e.g., 15000
- Property Type:** Select Type
- Description:** e.g., A beautiful 3-bedroom apartment

At the bottom of the form is a link 'Google Maps Link'.

ChatGPT Rentify localhost:5173/profile

Rentify

Properties About Profile

Rent Amount (in ₹)

e.g., 15000

Property Type

Select Type

Description

e.g., A beautiful 3-bedroom apartment

Google Maps Link

e.g., https://maps.google.com/?q=...

Property Image

Choose file No file chosen

Posted Date

dd/mm/yyyy

Post Property

ChatGPT Rentify localhost:5173/properties

Rentify

Properties About Profile

PROPERTIES POSTED

See All



₹13,500 Per Month

A Spacious 2BHK Home Near the ABC School

Heart Add to Favorites Request to Book



₹24,000 Per Month

Stunning 3BHK with Premium Amenities at ABC Apartments, Mumbai

Heart Add to Favorites Request to Book



₹9,800 Per Month

1BHK Home Surrounded by Nature in Kerala

Heart Add to Favorites Request to Book



₹6,800 Per Month

1RK Home Near Connaught Place, Delhi

Heart Add to Favorites Request to Book



₹28,000 Per Month

Spacious 4BHK Home Near IT Parks in Bangalore

Heart Add to Favorites Request to Book



₹4,500 Per Month

1RK near the Marina Beach

Heart Add to Favorites Request to Book

ChatGPT Rentify

localhost:5173/view/673c843c5f57113c540e37bb

Properties About Profile

Rentify



Stunning 3BHK with Premium Amenities at Abc Apartments, Mumbai

Door no 12, Second floor, Abc Apartments, Mumbai

Rent Per Month ₹24,000

Property Type 3BHK, Mumbai

View on Map

Posted on: 18th Nov 2024

Add to Favorites

Request to Book

ChatGPT Rentify

localhost:5173

Properties About Profile

Rentify

Discover a wide range of properties across India.

Filter Preferences

Refine your search based on your requirements

BHK Type: 2 BHK

Location: Bengaluru

Price Range: ₹10,001 - ₹15,000

Cancel Apply Filters

Rentify Builder Assist

Looking to rent out or sell your property projects? We're here to help!

localhost:5173/# work email Request demo

The screenshot shows the Rentify web application interface. At the top, there's a navigation bar with tabs for 'Properties', 'About', 'Signup', and a success message 'Logout successful!'. Below the header is a purple header bar with the user's name 'Ajay' and email 'ajay@gmail.com', along with three buttons: 'As a Tenant →', 'As a Owner →', and 'Logout →'. The main content area has two tabs: 'Requested' (which is selected) and 'Favorites'. Below these tabs is a table with columns for 'OWNER DETAILS', 'REQUESTED DATE', 'STATUS', and 'VIEW PROPERTY'. The table contains three rows of data:

OWNER DETAILS	REQUESTED DATE	STATUS	VIEW PROPERTY
Christina Bersh christina@site.com	10 Oct 2024	Accepted	More Info
David Harrison david@site.com	04 Aug 2024	Rejected	More Info
Anne Richard anne@site.com	July 10 2024	Pending	More Info

Mobile Responsive

The image displays two mobile phone screens side-by-side, illustrating the Rentify application's responsive design across different platforms.

Left Screen (iOS Mockup):

- The top navigation bar includes the 'Rentify' logo, a menu icon, and a search bar with placeholder text 'Explore Available Rentals | Find a Home >'.
- The main headline reads 'Find Your Dream Home Today'.
- A descriptive paragraph states: 'Discover a wide range of rental properties across prime locations. Whether you're a tenant or an owner, we've got you covered'.
- A blue button labeled 'BHK & Location Filter' with a filter icon is located below the text.
- A dark call-to-action section titled 'Rentify Builder Assist' contains the text: 'Looking to rent out or sell your property projects? We're here to help!' and a 'Request demo' button.

Right Screen (Android Mockup):

- The top navigation bar includes the 'Rentify' logo, a close button, and a menu icon.
- The main content area shows a user profile for 'Dev' with the email 'devsanthosh720@gmail.com'.
- Three buttons are available: 'As a Tenant →', 'As a Owner →', and 'Logout →'.
- The bottom navigation bar features tabs for 'Requested' (selected) and 'Favorites'.

The screenshot shows a property listing for a 4BHK home. At the top, the Rentify logo is visible. Below it is a large image of a modern, two-story house with illuminated windows at dusk. The listing title is "Spacious 4BHK Home Near IT Parks in Bangalore". Below the title is the address "No 3, Adani Street, Bangalore-45". A section for "Rent Per Month" shows "₹28,000". Another section for "Property Type" shows "ADUK Bangalore".

Spacious 4BHK Home Near IT Parks in Bangalore

No 3, Adani Street, Bangalore-45

Rent Per Month
₹28,000

Property Type
ADUK Bangalore

The screenshot shows a list of properties under the heading "PROPERTIES POSTED". It features a large image of a modern, two-story house with a garage and a driveway. The price "₹13,500 Per Month" is displayed below the image. A brief description follows: "A Spacious 2BHK Home Near the Abc School". Two buttons are present: "Add to Favorites" and "Request to Book".

PROPERTIES POSTED

See All

₹13,500 Per Month

A Spacious 2BHK Home Near the Abc School

Add to Favorites Request to Book

12. KNOWN ISSUES

1. Payment Gateway Integration

- The payment gateway setup is incomplete, preventing payment processing for rent and deposits. Full functionality will be available once the API integration is finalized.

2. Scalability Challenges

- The app may face slow load times and downtime as the user base grows. Backend optimization with load balancing and caching is planned.

3. Error Handling Mechanisms

- Limited error handling causes generic messages and potential crashes. Enhanced exception handling and user-friendly messages are in progress.

4. Image Upload Limitations

- Users face file size and format constraints during image uploads. Plans include compression, validation, and progress indicators.

5. Security Concerns

- Basic security features are in place, but advanced protections like encryption and vulnerability defenses are pending implementation.

13. FUTURE ENHANCEMENTS

- **AI-Driven Property Suggestions**
 - Develop an AI-based recommendation engine that analyzes user preferences and search patterns to suggest suitable properties.
- **Enhanced Admin Panel**
 - Upgrade the admin dashboard with advanced features for managing properties, users, and bookings more efficiently.
- **Social Media Account Integration**
 - Add login options using platforms like Google and Facebook to simplify and speed up user registrations.
- **Comprehensive Reviews & Ratings**
 - Implement a detailed review system allowing users to provide feedback, upload images, and rate properties.
- **Live Chat Functionality**
 - Introduce a real-time chat feature to enable instant communication between property owners and tenants.
- **Intelligent Notifications & Updates**
 - Develop a notification system to keep users informed about property availability, booking updates, and important alerts.