

# Full Stack Development with MERN

## Project Documentation

### 1. Introduction

**Project Title: Househunt: Finding Your Perfect Rental Home**

**Team Members:**

**Team Size: 2**

**Team Leader:**

**Kamisetti Jyothsna**– Full Stack Development (Backend APIs, Database Integration, Project Integration) & Frontend Support & Testing

**Pinninti Ujwala Akshaya**– Documentation & Presentation Support, Frontend Development (UI Components & Design) & Project Deployment

### 2. Project Overview

**Purpose:**

The purpose of this project is to develop a full-stack web application where users can search houses, rent houses, and view their areas and cost.

The system also provides an Admin Panel to manage houses and monitor all bookings.

This project demonstrates real-world MERN stack development skills.

**Features:**

 **User Features:**

- User Registration & Login
- View Available Houses
- Rent House
- Filter by address
- Search Properties
- View Available Properties

## **Admin Features:**

- Admin Login
- Add New House
- Edit House Details
- Delete House
- View All Houses
- View All Bookings

## **3. Architecture**

### **Frontend (Client – React)**

- Built using React.js
- Component-based architecture
- React Router for navigation
- CSS for styling
- Runs on: `http://localhost:3000`

Frontend communicates with backend using REST APIs.

### **Backend (Server – Node.js + Express)**

- Built using Node.js and Express.js
- REST API architecture
- Password hashing using bcrypt
- CORS enabled
- JSON request handling
- Runs on: `http://localhost:6001` Users
- House for rents
- House for sales

### **Database (MongoDB)**

MongoDB used as NoSQL database.

Collections:

- User
- House for Rent
- House for Sales

Schemas created using Mongoose.

## 4. Setup Instructions

- **Prerequisites:**
  - Node.js (v16 or above)
  - MongoDB (Local or Atlas)
  - Postman
  - VS Code

### Installation:

#### Step 1: Clone Project

```
git clone <your-repo-link>
```

#### Step 2: Backend Setup

```
cd server
```

```
npm install
```

```
npm start
```

Backend runs on:

<http://localhost:6001>

#### Step 3: Frontend Setup

```
cd client
```

```
npm install
```

```
npm start
```

Frontend runs on:

<http://localhost:3000>

## 5. Folder Structure

- Admin

admin/

|

├── src/

| ├── components/

| ├── pages/

| ├── styles/

| ├── App.js/

| └── index.js

- **Server:**

Owner/

|

├── schemas.js

├── index.js

└── package.js

## 6. Running the Application

1. Start MongoDB
2. Start Backend:  
npm start
3. Start Frontend:  
npm start

Open:

- Frontend → <http://localhost:3000>
- Backend → <http://localhost:6001>

## 7. API Documentation

- **Authentication:**

| Method | Endpoint  | Description   |
|--------|-----------|---------------|
| POST   | /register | Register user |
| POST   | /login    | Login user    |

- **Houses:**

| Method | Endpoint          |
|--------|-------------------|
| GET    | /fetch-Houses     |
| POST   | /add-Houses       |
| PUT    | /update-House/:id |
| DELETE | /delete-House/:id |

- **Bookings:**

| Method | Endpoint                |
|--------|-------------------------|
| POST   | /rent-house             |
| GET    | /fetch-bookings/:userId |
| GET    | /all-bookings           |
| DELETE | /cancel-booking/:id     |

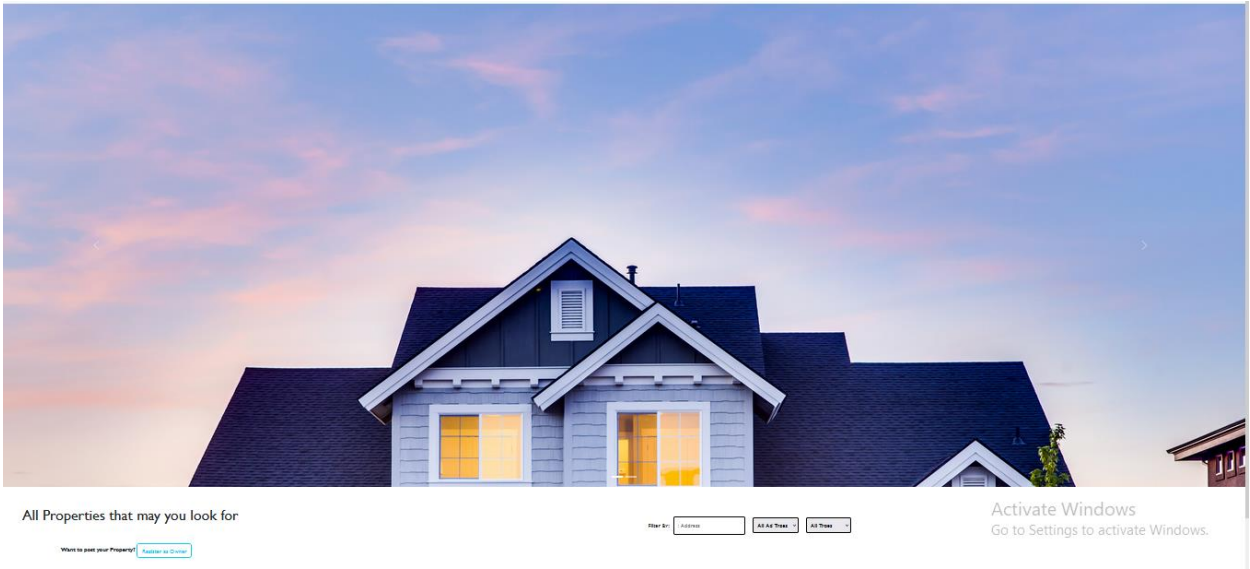
## 8. Authentication

- Passwords encrypted using bcrypt
- User login validation
- Admin access for House management
- Session stored in frontend (localStorage)

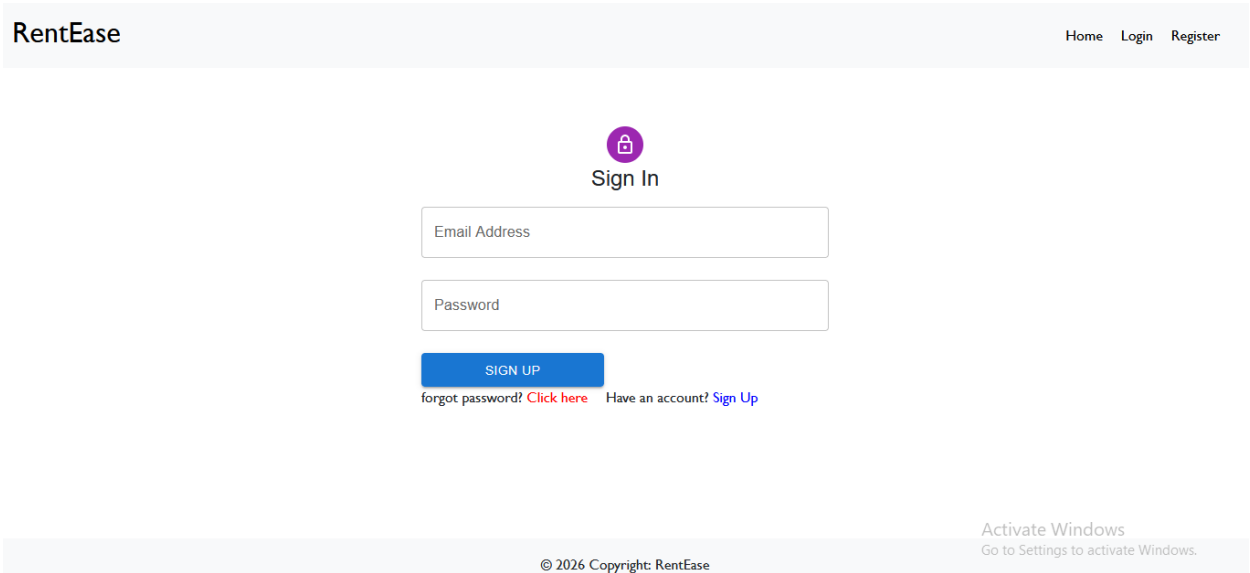
# 9. User Interface

• The UI includes:

- Home Page




- Login Page



- Registration Page

RentEase

HomeLoginRegister



Sign up

Renter Full Name/Owner Name

Email Address

Password

User Type

SIGN UP

Have an account? [Sign In](#)

Activate Windows

Go to Settings to activate Windows.

© 2026 Copyright: RentEase

## 10. Testing

- Manual testing performed for all APIs

### Verified:

- Registration
- Login
- Add Flight
- View Bookings
- Admin Operations

Tested using:

Browser

Postman

## 11. Screenshots or Demo

- Demo Video Link:

[https://drive.google.com/file/d/1-RnrqbmSex2uWHPPUE33BUgcx69J7pyz/view?usp=drive\\_link](https://drive.google.com/file/d/1-RnrqbmSex2uWHPPUE33BUgcx69J7pyz/view?usp=drive_link)

Screenshots included in documentation.

## 12. Known Issues

- No payment integration
- No email notifications
- Basic admin authentication
- UI responsiveness can be improved

## 13. Future Enhancements

- Payment Gateway (Razorpay/Stripe)
- Email Rent/Sale Confirmation



- Area Selection
- Role-based Authentication (JWT)
- Deployment on Render / AWS