



# My Apartment

A Secure, Easy-to-Use Rental Property Listing Platform

---



## 1. Introduction

**My Apartment** is a modern, web-based rental listing platform that bridges the gap between property owners and tenants. It is designed to make **property discovery and rental management** simple, transparent, and secure.

On this platform, verified users can **post, edit, and manage** property listings such as houses, rooms, and flats.

It combines **secure authentication, interactive maps, image cloud storage, real-time chat**, and **advanced property search filters** into one seamless experience.

### Target Audience:

- **Property Owners / Landlords** who want a reliable way to list and promote their rental spaces.
  - **Tenants / Renters** searching for verified properties without the hassle of scams or incomplete information.
- 



## 2. Problem Statement

Despite the growing popularity of online property rental platforms, most existing solutions have several **pain points**:

1. **Complicated Listing Workflows** – Adding a property often involves multiple confusing steps.
2. **Lack of Property Verification** – Leads to fake listings, scams, and wasted time.
3. **No Direct Communication** – Most platforms don't allow real-time owner-tenant conversations.
4. **Weak Search Filters** – Limited or ineffective search options cause users to spend more time finding relevant listings.

### **Impact:**

These issues reduce efficiency, increase the risk of fraud, and cause **frustration** for both property owners and seekers.

---



## **3. Solution Provided**

**My Apartment** directly addresses these challenges with a **feature-rich, user-first design**:

- **Secure Onboarding** – OTP-based account verification ensures only genuine users can list properties.
  - **Guided Listing Process** – Step-by-step form to make listing creation simple and error-free.
  - **Location Accuracy** – Integration with **Mapbox** to pinpoint exact property locations.
  - **Real-Time Communication** – WebSocket-powered chat for instant interaction between owners and tenants.
  - **Trust Signals** – Ratings and reviews on each property to build credibility.
  - **Intelligent Search** – Filter properties by title, city, and location for faster discovery.
- 



## **4. Features (Detailed)**

### **4.1 Property Management**

- Add listings with **title, price, city, location, and detailed description**.
- Upload multiple high-quality images stored securely in **Cloudinary**.
- Edit or update any listing at any time.

### **4.2 Security & Verification**

- OTP verification via **NodeMailer** before account activation.
- **Passport.js** authentication using cookies for persistent and secure login.

## 4.3 Search & Discovery

- Search powered by **MongoDB Atlas Search** for fast, accurate results.
- Filters available for **title, city, and location**.
- View properties on an interactive **Mapbox** map.

## 4.4 User Interaction

- Post reviews and ratings for properties.
- Real-time chat between owner and tenant for quick queries and negotiations.

## 4.5 Error Handling & Validation

- Centralized **Express.js** error middleware for handling backend errors.
  - Client-side form validation for a smoother user experience.
  - Schema validation with **Joi** to ensure data integrity.
- 



## 5. Tech Stack

Category	Technologies
Frontend	HTML, CSS, Bootstrap, JavaScript
Backend	Node.js, Express.js
Database	MongoDB Atlas
Authentication	Passport.js (Cookies)
Integrations	NodeMailer, Mapbox, Cloudinary, WebSocket
Hosting	Render

---

## 6. System Architecture

### Workflow:

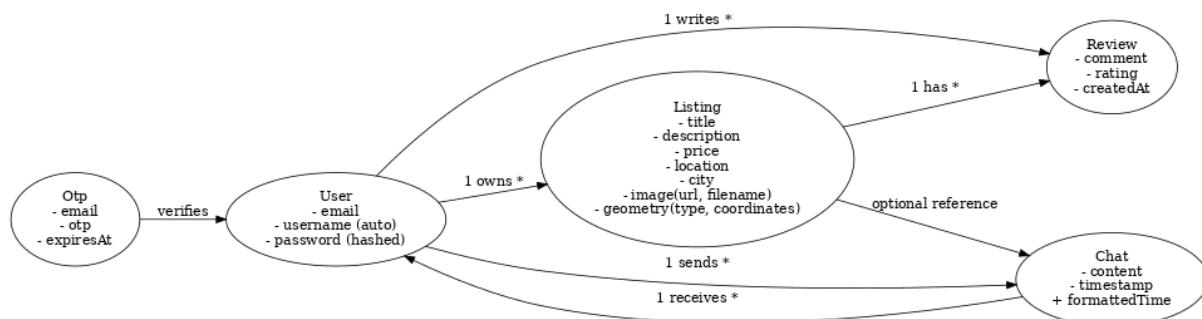
1. **Client** → Sends requests to backend via browser.
  2. **Express.js Server** → Handles routing, validation, and business logic.
  3. **MongoDB Atlas** → Stores and retrieves user, listing, and review data.
  4. **Third-Party Services** → Mapbox (maps), Cloudinary (images), NodeMailer (emails), WebSocket (chat)
- 

## 7. Database Design

### Entity Relationship (ER) Diagram (Text Description):

- A **User** owns multiple **Listings**.
- Each **Listing** can have multiple **Reviews**.
- A **Listing** can also be related to multiple **Chats**.
- A **User** can participate in multiple **Chats**.
- An **Otp** validates a **User** during registration or verification.

### Database ERD Image



## 7.2 Collections Overview

### Users Collection (User)

Stores registered user accounts.

- **username** (String) – Managed automatically by `passport-local-mongoose`.
  - **email** (String) – Unique email used for account verification.
  - **hash** (String) – Encrypted password hash (auto-managed).
- 

### Listings Collection (Listing)

Stores property details posted by owners.

- **title** (String) – Property title.
  - **description** (String) – Detailed property description.
  - **price** (Number) – Monthly rental price.
  - **location** (String) – Area or street name.
  - **city** (String) – City where the property is located.
  - **image** (Object) – Contains `{url, filename}` for Cloudinary image storage.
  - **reviews** (Array of IDs) – References to `Review` documents.
  - **owner** (ObjectId) – Reference to the `User` who owns the listing.
  - **geometry** (GeoJSON) – `{type: "Point", coordinates: [longitude, latitude]}` for mapping.
- 

### Reviews Collection (Review)

Stores ratings and feedback for listings.

- **comment** (String) – Review text provided by the user.

- **rating** (Number) – Rating value (1–5).
  - **createdAt** (Date) – Auto-generated timestamp of review creation.
  - **author** (ObjectId) – Reference to the [User](#) who wrote the review.
- 

### Chats Collection (Chat)

Stores messages exchanged between two users.

- **sender** (ObjectId) – Reference to the sending [User](#).
  - **receiver** (ObjectId) – Reference to the receiving [User](#).
  - **listing** (ObjectId) – Optional reference to the related [Listing](#).
  - **content** (String) – The message text.
  - **timestamp** (Date) – Auto-generated timestamp of when the message was sent.
- 

### OTP Collection (Otp)

Stores temporary one-time passwords for email verification.

- **email** (String) – Email address associated with verification.
  - **otp** (String) – One-time password code.
  - **expiresAt** (Date) – Expiration time (record auto-deletes after expiry).
- 

## ⚙ 8. Installation & Setup

```
# Clone repository  
git clone https://github.com/yourusername/my-apartment.git
```

```
# Enter project directory  
cd my-apartment
```

```
# Install dependencies  
npm install
```

```
# Start development server  
npm run dev
```

---

## 9. Environment Variables

Create a `.env` file in the project root and all credentails

---

## 10. Usage Guide

1. **Register/Login** → Complete OTP verification via email.
  2. **List a Property** → Fill in details and upload photos.
  3. **Search Listings** → Filter by title, city, location.
  4. **View Property Details** → Access map, images, and reviews.
  5. **Chat with Owner** → Send messages instantly.
-

## 拍照 11. Screenshots

### Home Page

The screenshot shows a web-based real estate platform. At the top, there's a navigation bar with a house icon, links for 'All Apartments' and 'Add New', and a search bar labeled 'Search By City...' with a 'Search' button. To the right is a user profile icon. Below the navigation, there are three large thumbnail images of houses, each with a 'Click to View Details' button. The first house is yellow with an orange roof, the second is white with a blue balcony, and the third is pink with a red roof. Each listing includes the name 'Dasji house' and its monthly rent price.

House Name	Rent Price
Dasji house	₹ 2,500 / Month
Dasji house	₹ 2,000 / Month
Dasji house	₹ 4,000 / Month

### View page

This screenshot shows a detailed view of a property listing for 'Dasji house'. The title 'Dasji house' is at the top in red. Below it is a large image of the house, which is a two-story yellow building with an orange roof and a car parked in front. The listing includes the owner's name ('Owned By: sonu'), price ('Price: ₹2,500/Month'), location ('Location: Sitamarhi, bihar'), city ('City: Bihar'), and a description. The description text is as follows:

2 BHK House for Rent near Railway Station Spacious and well-maintained 2 BHK house available for rent. Located just a short walk from the railway station, this home is perfect for families or working professionals. 🏠 Property Highlights: 2 Bedrooms, 1 Hall, 1 Kitchen Peaceful residential area Close to market, shos. and daily essentials Ground nearbv. ideal

[Read More](#)

At the bottom, there are two buttons: 'Edit Listing' and 'Delete'.

## View page

All Apartments Add New

Search By City... Search

Leave a Review

★★★★★

Comments

Submit Review

All Reviews

monu

★★★★★

Room owner is very cooperative

Home Location



## Add home/room/flat page

All Apartments Add New

Search By City... Search

Title

Enter Title

Description

Enter description

Image

Choose File No file chosen

Price

Enter Price

City

Enter City

Location

Enter Location

Add this

## Advance search filter



All Apartments Add New

sitmarhi

Search



Dasji house  
₹ 2,000 / Month



Dasji house  
₹ 2,500 / Month



Dasji house  
₹ 4,000 / Month



All Apartments Add New

sitmar

Search



Dasji house  
₹ 2,000 / Month



Dasji house  
₹ 2,500 / Month



Dasji house  
₹ 4,000 / Month

## Inbox page

The screenshot shows a user interface for managing an inbox. At the top, there is a navigation bar with a house icon, links for "All Apartments" and "Add New", a search bar labeled "Search By City...", and a search button. On the right side of the header is a user profile icon. Below the header, the title "Your Inbox" is displayed next to an envelope icon. A single message card is visible, showing a profile picture of a person with a blue circle, the name "monu", the apartment name "Dasji house", and the message content "Message: hi". To the right of the message card, the timestamp "2/8/2025, 6:36:44 pm" and a trash bin icon are present.

## Live chat page

The screenshot shows a live chat interface. At the top, there is a navigation bar with a house icon, links for "All Apartments" and "Add New", a search bar labeled "Search By City...", and a search button. On the right side of the header is a user profile icon. Below the header, the apartment details are shown: "Apartment: Dasji house" and "House Owner: monu (Offline)". The main area is a chat window showing a conversation between the user and the house owner. The user's messages are in pink boxes with the text "Hi" and the timestamp "2/8/2025, 5:24:08 pm". The house owner's messages are in green boxes with the text "hello" and the timestamp "2/8/2025, 6:11:56 pm", followed by "hi" and the timestamp "2/8/2025, 6:36:44 pm". At the bottom of the chat window, there is an input field for typing messages and a blue "Send" button. The footer of the page includes social media icons for Facebook, Instagram, LinkedIn, Email, and Twitter, the copyright notice "©My Apartment", and links for "Privacy" and "Terms". The URL "https://mymapartments.onrender.com/listings" is also visible at the bottom.

---

## 12. Future Enhancements

- Add **price range filters**.
  - Create an **admin dashboard** for managing users and listings.
  - Develop a **mobile application** version.
  - Implement **push notifications** for chat and listing updates.
- 

## 13. View Live here another project also

[My Apartment – Live View: https://myapartment.onrender.com/listings](https://myapartment.onrender.com/listings)

[Portfolio Website – Live View: https://sonukumarwebsite.netlify.app/](https://sonukumarwebsite.netlify.app/)

[My Music App – Live View: https://my-music-sss.netlify.app/](https://my-music-sss.netlify.app/)