### 

### Internship Report

### on

### HouseHunt: Find Your Perfect Rental Home

### At

### Smartbridge

### Submitted by Team ID: LTVIP2025TMID56731

Team Leader: Team Members:

Lakshmi Charitha Sri Kuntamukkala Kalpana Pacha

Jasmin Shaik

Teja Sri Durga Thentu

1.Introduction

**HouseHunt: Finding Your Perfect Rental Home** is a comprehensive web application designed to streamline the process of renting residential properties. HouseHunt offers a user-centric platform where renters and property owners can seamlessly connect, communicate, and complete rental transactions in a secure, transparent environment.

In an age where digital convenience is essential, traditional methods of finding rental homes—such as classified ads or third-party brokers—can be time-consuming, inefficient, and often unreliable. HouseHunt solves these challenges by delivering an intuitive and feature-rich solution that simplifies the entire rental journey, from discovering a property to finalizing lease agreements.

The application is equipped with a range of key features:

* **Property Listings**: A centralized database of rental properties with high-quality images, pricing, descriptions, and location details.
* **Advanced Search Filters**: Enables users to refine searches based on budget, location, property type, size, and amenities.
* **Direct Communication**: Secure in-app messaging and inquiry forms facilitate smooth interactions between renters and property owners.
* **Role-Based Access**: Tailored dashboards and controls for Renters, Property Owners, and Admins.
* **Booking & Lease Management**: Tools to manage inquiries, confirm bookings, and negotiate lease agreements within the platform.

In summary, **HouseHunt** is not just a house rental app—it’s a smart digital companion for modern renters and property managers, offering a seamless, secure, and scalable way to find or list the perfect rental home.

1.1 Project Overview

**HouseHunt** is a full-stack rental housing application built with the MERN (MongoDB, Express.js, React.js, Node.js) stack. It aims to provide a seamless experience for renters and property owners to connect, communicate, and complete the rental process securely and efficiently.

The platform supports three main user roles:

* **Renters**: Users looking to find and rent properties.
* **Owners**: Individuals or agencies listing properties for rent.
* **Admins**: Platform moderators responsible for user verification and governance.

1.2 Project Purpose

The purpose of the **HouseHunt** project is to create a user-friendly, scalable, and secure digital platform that simplifies and modernizes the residential property rental process for both renters and property owners. By leveraging modern web technologies and an intuitive user experience, HouseHunt aims to address the common pain points associated with traditional property rental methods—such as lack of transparency, inefficient communication, and limited accessibility.

-> Streamline Property Discovery

-> Enable Seamless Communication

-> Support Role-Based

**2. Features**

**2.1 Property Listings:**

* A dynamic listing of all available rental units.
* Includes photos, location, rent amount, and property specifications

2.2 **Advanced Search Filters**

* Filter by location, rent range, number of bedrooms, amenities, and property type (apartment, house, room, etc.).

**2.3 Inquiry & Booking System**

* Direct communication between renters and owners.
* Built-in messaging or inquiry forms.
* Booking status management (e.g., pending, confirmed).

**2.4 User Registration & Authentication**

* Separate flows for **Renters** and **Owners**.
* Secure login, signup, and password management.

**2.5 User Profile Management**

* Renters can:
  + View their booking history.
  + Manage saved properties.
* Owners can:
  + View property statistics.
  + Track tenant interactions.

**2.6 Admin Approval System**

* Review and approve owner registrations.
* Monitor user activities and enforce platform policies.

**2.7 Rental Agreement & Documentation**

* Support for digital lease agreements.
* Lease term negotiation through chat.
* Record of rental history and agreement summaries.

**2.8 Notifications System**

* Email or in-app alerts for:
  + New messages.
  + Booking confirmations or rejections.
  + Admin approvals or actions.
  + Status updates on listings or inquiries.

**3. Technical Architecture**

* **Client-server model**: React frontend communicates with Express backend via REST APIs.
* **Frontend**:

->React.js with Bootstrap and Material UI for responsive UI

->Axios for making RESTful API calls to the backend

->Role-based dashboard components for Renter, Owner, and Admin

* **Backend**:

->**Node.js** with **Express.js** to handle routing and server logic

->RESTful APIs for user authentication, property management, messaging, etc.

* **Database**: MongoDB with Mongoose stores users, properties, bookings; hosted on MongoDB Atlas.
* **Image Storage**: Cloudinary or AWS S3 used for property images.
* **Admin Panel**: Manages user approvals, platform monitoring, and policy enforcement.
* **Security**: Passwords hashed with bcrypt, JWT for auth, Helmet for security headers, CORS enabled.

**4.Assumptions**

* Users are renters, owners, or admins.
* Owners must be approved to add properties.
* Users register/login to book or inquire.
* Bookings are requests, not guaranteed leases.
* Lease agreements finalized via app messaging.
* Admin enforces policies and approves accounts.
* Only available properties can be booked.

**5.Dependencies**

* **Frontend:** React, React Router, Axios, Bootstrap/Material-UI.
* **Backend:** Express, MongoDB (Mongoose), JWT, Bcrypt, Multer, Cloudinary (optional).
* **Tools:** MongoDB Atlas, hosting services (Heroku/Vercel), Postman.
* **Deployment**: Frontend on Vercel/Netlify, backend on Heroku/Render, database on MongoDB Atlas.

**6. Security & Compliance**

* Password hashing using bcrypt
* JWT-based session management
* Admin approval to prevent spam/fraud
* GDPR-compliant data handling (optional for expansion)

**7. Future Enhancements**

* Payment gateway integration (Stripe, Razorpay)
* In-app chat system
* Mobile app version (React Native)
* Push notifications
* Rating & Review system for properties and owners

**8. Conclusion**

**HouseHunt** streamlines the rental process by bringing tenants and landlords together on one platform, providing tools to communicate, manage, and finalize rental transactions. With a modern tech stack and a scalable architecture, this project sets a solid foundation for real-world deployment and further enhancements.

**9. Results**

Output screenshots to include:

* Home Page
* Sin Up Page
* Properties Page
* Inquiry page
* Dashboard Page









