🏠 HouseHunt: Finding Your Perfect Rental Home

Category: Full Stack Development **Team ID**: LTVIP2025TMID57855 **Team Size**: 4

Team Members:

- 😊💼 Team Leader: Mudiyam Nithin Simha

- 😊💻 Team Member: Momin Shameem

- 😊💻 Team Member: Kummaripavan Kumar

- 😊💻 Team Member: B Ramcharan

🔍 Project Overview

HouseHunt is a web-based rental platform built using the MERN stack. It facilitates end-to-end rental workflows between property owners and renters. The platform includes features like account management, property browsing with filters, secure bookings, and communication—all under a well-governed, responsive, and secure framework.

🛠 Skills Required

- HTML5, CSS3

- JavaScript (ES6)

- Bootstrap & Material UI

- React.js (Frontend)

- Node.js & Express.js (Backend)

- MongoDB (Database)

💡 Key Features

- **User Roles**: Renter, Owner, Admin

- **Secure Login/Registration**

- **Property Listings**: with dynamic filters

- **Real-Time Inquiry & Messaging**

- **Booking & Lease Flow**: managed entirely within the platform

- **Admin Controls**: user verification, content moderation

- **Responsive UI** using Bootstrap and Material UI

😊💼 Scenario-Based Flow

User: Alice – a Renter

1. Registers and logs in

2. Browses properties using filters (location, price, rooms, etc.)

3. Sends an inquiry for a selected apartment

4. Receives booking approval & negotiates lease with the owner

5. Confirms rental and moves in

User: Bob – a Property Owner

1. Registers and requests approval

2. Admin reviews and approves ownership

3. Bob adds and manages properties

4. Accepts/declines rental requests

Admin Role

- Approves new owners and monitors user activity

- Ensures compliance with platform policies

🏦 Technical Architecture

Frontend

- Built with **React.js**

- Styled with **Bootstrap** & **Material UI**

- Uses **Axios** for HTTP communication

- Fully responsive across devices

Backend

- **Node.js** with **Express.js** for RESTful APIs

- Modular route and controller architecture

- Implements business logic and authorization layers

Database

- **MongoDB**: stores user info, listings, transactions, and communication logs

- Mongoose ODM for schema design and validation

🔐 Security & Governance

- Token-based authentication (JWT or similar)

- Form validation & role-based access control

- Admin interface for oversight and moderation

- Data encryption and secure messaging

☑ Future Enhancements

- Integration with **Google Maps** for location visualization

- **Image upload** support for property listings

- **Payment gateway** for online rental processing

- Mobile app version using **React Native**

- Smart suggestions with **ML/AI** (e.g. best-match listings)