# **FULL STACK DEVELOPMENT WITH MERN PROJECT**

### 🏠 **House Hunt: Finding Your Perfect Rental Home**

**Team ID:** LTVIP2025TMID60989

**Team Members:**

* Team Leader: [**NUTHALAPATI. TEJASWINI**] – Full Stack Developer & Project Coordinator
* Frontend Developer: [**ORUGANTI SUPRAJA**] – React.js & UI Specialist
* Backend Developer: [**MEDIBOYINA MOUNIKA**] – Node.js & API Specialist
* Database Administrator: [**KUPPALA VENKATA SAI PAVITHRA**]– MongoDB Schema Designer
* Tester:[**NALLABROLU TIRUMALA VENKATA PAVANI**]- Test the app and modules

## **📑 Table of Contents**

1. Team Details
2. Introduction
3. Problem Statement
4. Objectives
5. Scope of the Project
6. Technical Architecture
7. ER Diagram
8. Features
9. Prerequisites
10. Project Flow (Milestones)
11. Database Schemas
12. Testing
13. Future Enhancements
14. Conclusion

**Collaboration Tools & Practices:**

* GitHub for version control and code reviews
* Agile methodology with Scrum sprints
* Slack, Trello, Jira for communication and project management
* CI/CD pipelines for automated testing and deployment

## **Introduction**

HouseHunt is a next-generation rental management ecosystem designed to simplify the **property discovery, listing, and rental process** while ensuring **trust, transparency, and efficiency**. In today’s digital world, renting a home should be as seamless as online shopping — quick, reliable, and secure. However, the current market still faces multiple inefficiencies, ranging from scattered property details to fraud and lack of verification.

### **Current Market Challenges**

* **For Tenants:** Property details are often incomplete, outdated, or misleading, forcing tenants to waste valuable time visiting multiple platforms or relying on brokers.
* **For Landlords:** Property owners face difficulty showcasing their rentals with detailed and attractive listings, reducing their chances of connecting with genuine tenants.
* **For Administrators:** Ensuring safety, transparency, and fraud prevention is challenging without centralized monitoring and verification mechanisms.

### **HouseHunt’s Value Proposition**

HouseHunt aims to bridge these gaps by creating a **centralized, user-friendly, and future-ready ecosystem**.

* Tenants can explore **verified properties** with **personalized recommendations** based on preferences, location, and budget.
* Landlords can manage **multimedia-rich listings** and access **analytics dashboards** to track engagement and inquiries.
* Administrators can monitor listings, verify tenant/landlord credentials, and remove fraudulent content, ensuring a safe and trusted environment.

### **Why HouseHunt is Different**

Unlike traditional property portals, HouseHunt focuses not only on **listing management** but also on **experience and trust-building**. Some unique aspects include:

* **AI-driven recommendations** to match tenants with the most relevant properties.
* **Predictive analytics** to forecast rental demand in specific areas.
* **Virtual property tours** and potential **AR/VR integration** to save time and effort.
* **Integrated rent payment system** for end-to-end rental lifecycle management.

### **Long-Term Vision**

HouseHunt is built with scalability in mind. The project envisions:

* Expanding to **mobile-first platforms** for tenants and landlords on the go.
* Supporting **multi-language interfaces** to cater to diverse users.
* Creating a **trust ecosystem** with landlord ratings, tenant verification, and fraud-prevention tools.

By combining technology, security, and convenience, HouseHunt is not just a property search platform — it is a **complete rental management ecosystem** for the future of urban housing.

## **Problem Statement**

The rental ecosystem is fragmented and inefficient:

* Tenants waste weeks navigating multiple unreliable platforms.
* Brokers increase costs while reducing transparency.
* Landlords struggle to connect with genuine tenants.
* Listings are often outdated, incomplete, or misleading.
* No centralized authority ensures **trust, safety, and accountability**.

Hence, there is a strong need for a **unified, scalable, and intelligent rental platform**.

## **Objectives**

* Develop a **centralized digital platform** for rentals.
* Provide tenants with **advanced search tools** (budget, amenities, location filters).
* Enable landlords to **list properties with multimedia support**.
* Empower admins with **monitoring, fraud detection, and reporting tools**.
* Ensure the system is **secure, scalable, and mobile-friendly**.
* Reduce dependency on brokers and increase transparency.

**Additional Objectives:**

* Personalized property recommendations.
* Transparent rental history and landlord ratings.
* Seamless integration with mobile apps for wider accessibility.

## **Scope of the Project**

### **Tenant Features**

* Registration, login, and profile management.
* Search rental properties with advanced filters.
* Save favorites/wishlist.
* Direct chat with landlords.
* Notifications for new listings.
* Appointment booking (future).

### **Landlord Features**

* Add and manage property listings with images/videos.
* Manage property availability (available, rented).
* Dashboard with analytics (views, inquiries, trends).
* Direct communication with tenants.

### **Admin Features**

* Manage tenants and landlords.
* Approve/reject suspicious properties.
* Detect and remove fraudulent content.
* Monitor chats/reports for safety.
* Generate usage and activity reports.

## **Technical Architecture**

**Frontend (Client Side):**

* React.js + TailwindCSS
* Responsive, mobile-first design
* SPA (Single Page Application) with reusable components

**Backend (Server Side):**

* Node.js + Express.js
* RESTful APIs for users, properties, favorites, bookings, messages
* JWT authentication for security

**Database:**

* MongoDB with Mongoose ODM
* Collections: Users, Properties, Bookings, Favorites, Messages

**Deployment:**

* AWS EC2 for server hosting
* AWS S3 for property image storage
* GitHub CI/CD pipeline for automation

**Future Architecture Goals:**

* Containerization with Docker
* Orchestration using Kubernetes
* Microservices for modular scalability

## **ER Diagram**

**Entities & Attributes**

* User: userId, name, email, role, password
* Property: propertyId, title, description, location, rent, amenities, landlordId
* Booking: bookingId, userId, propertyId, date, status
* Favorites: favId, userId, propertyId
* Message: msgId, senderId, receiverId, content, timestamp

**Relationships**

* One landlord → Many properties
* One tenant → Many bookings/favorites
* Many-to-many → Messaging between users

## **Features**

**Tenant:**

* 🔍 Search properties by filters
* ❤️ Save favorites
* 💬 Chat with landlords
* 📅 Book property visits
* 🔔 Get notifications

**Landlord:**

* 🏡 Add/manage property listings
* 📸 Upload multimedia content
* 📊 Track performance analytics
* 📩 Manage tenant inquiries

**Admin:**

* 👤 Manage users
* ✅ Approve/reject listings
* 🕵️ Detect/remove fraudulent content
* 📈 Generate reports

## **Prerequisites**

* Node.js + npm
* MongoDB Atlas / Compass
* Express.js
* React.js
* TailwindCSS
* Git & GitHub
* VS Code IDE
* Chrome/Firefox Browser

## **Project Flow (Milestones)**

1. **Setup:** Install Node.js, MongoDB, React, Git. Setup GitHub repo.
2. **Backend Development:** Express server, routes (user, property, booking, favorites, messages), MongoDB integration.
3. **Frontend Development:** React pages, components, property search & filters, chat UI.
4. **Integration & Testing:** Connect frontend & backend, test workflows.
5. **Deployment:** Deploy on AWS with CI/CD pipeline.

## **Database Schemas**

* **User Schema:** name, email, password, role, createdAt
* **Property Schema:** title, description, location, rent, amenities, landlordId, images
* **Booking Schema:** bookingId, userId, propertyId, date, status
* **Favorites Schema:** favId, userId, propertyId
* **Message Schema:** msgId, senderId, receiverId, content, timestamp

## **Testing**

* **Unit Testing:** Jest for backend APIs
* **Integration Testing:** API + frontend connectivity
* **System Testing:** End-to-end workflows
* **UAT:** With real test users
* **Security Testing:** JWT authentication, password encryption, NoSQL injection prevention

## **Future Enhancements**

* AI-based property recommendations
* Virtual 3D and AR property tours
* Integrated rent payment system (UPI/cards/net banking)
* Verified tenants via Aadhaar/KYC
* Mobile app (Android/iOS)
* Voice-based search & chatbot
* Long-term: Blockchain-based rental agreements

## **Conclusion**

HouseHunt is **more than a rental app** — it is a **complete rental management ecosystem** designed to ensure trust, convenience, and efficiency.

With **scalable MERN architecture, AI-driven recommendations, immersive property tours, and integrated payments**, HouseHunt has the potential to transform the rental housing market, reduce broker dependency, and bring transparency and efficiency for all stakeholders.