

# 1. INTRODUCTION

## 1.1 Project Overview

HouseHunt is a web-based real estate rental platform designed using the MERN stack. It simplifies the process of finding, listing, and managing rental properties by offering intuitive tools for renters, owners, and admins.

## 1.2 Purpose

The primary goal of HouseHunt is to streamline the rental process for all parties-renters, property owners, and real estate professionals-by providing a centralized, user-friendly application for property discovery, booking, and management.

# 2. IDEATION PHASE

## 2.1 Problem Statement

Traditional house-hunting is often tedious and lacks transparency. There is a need for a centralized system that connects renters with legitimate property owners, supports virtual tours, and simplifies booking and approvals.

## 2.2 Empathy Map Canvas

Users: Renters, Property Owners, Admin

Needs: Easy access to verified property listings, simplified communication, booking history tracking

Pain Points: Fake listings, delayed approvals, lack of centralized tools

Goals: Smooth rental process, secure platform, responsive user experience

## 2.3 Brainstorming

Ideas considered included a mobile-only version, tenant-focused features, and integration with financial services. The final decision was to build a web platform supporting Renters, Owners, and Admins.

### **3. REQUIREMENT ANALYSIS**

#### **3.1 Customer Journey Map**

1. Renter: Browses properties - views info - sends booking request - waits for owner-s approval
2. Owner: Gets approved by Admin - adds property - responds to booking
3. Admin: Approves Owners - Monitors activity

#### **3.2 Solution Requirement**

- Responsive UI
- Role-based access control
- Real-time property updates
- Booking status and history

#### **3.3 Data Flow Diagram**

Refer Appendix for DFD

#### **3.4 Technology Stack**

Frontend: React.js, Material UI, Bootstrap, Axios, Moment.js, Ant Design

Backend: Node.js, Express.js, MongoDB, Mongoose

Others: Multer, CORS, JWT, Dotenv, BcryptJS

### **4. PROJECT DESIGN**

#### **4.1 Problem Solution Fit**

The system addresses inefficiencies in house rental workflows by automating discovery, communication, and booking processes.

#### **4.2 Proposed Solution**

A multi-role platform with distinct dashboards for Renters, Owners, and Admins, featuring listing

creation, virtual previews, and booking management.

4.3 Solution Architecture

Client-server model with frontend communicating via RESTful APIs. Backend uses Express and MongoDB for data processing and storage.

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Milestone	Description	Tools Used
----- ----- -----		
M1	Project Setup	VS Code, Git, npm
M2	Backend Development	Express, Mongoose
M3	Database Schema Design	MongoDB
M4	Frontend Development	React.js, Material UI
M5	UI Integration	Ant Design, Bootstrap
M6	Testing & Deployment	Postman, Localhost

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Tool Used: Postman

Testing Scope:

- API Response Time
- Concurrent Booking Scenarios
- Role-Based Access Test

Result: Application responded efficiently under load with minimal delay in booking status updates.

7. RESULTS

7.1 Output Screenshots

- User Registration and Login

- Property Listing Interface
- Admin Dashboard
- Owner Property Management View
- Booking Status Page

## **8. ADVANTAGES & DISADVANTAGES**

Advantages:

- Real-time property updates
- Virtual tours and detailed listings
- Role-based access enhances security
- Scalable architecture

Disadvantages:

- Not mobile-optimized yet
- No inbuilt payment gateway
- Relies on manual Admin approval

## **9. CONCLUSION**

HouseHunt successfully streamlines the real estate rental process with an efficient platform for tenants, property owners, and administrators. With its advanced features and clean architecture, it improves transparency and user engagement.

## **10. FUTURE SCOPE**

- Mobile App version using React Native
- Payment Integration for deposits
- AI-based recommendation engine
- Chat support for renters and owners

## **11. APPENDIX**

- ER Diagram

- Data Flow Diagram
- Project Structure (Frontend & Backend Folder Layout)
- Milestone Setup Screens
- UI Screenshots (from Project Demo)