Project Report Format

1. INTRODUCTION

1.1 Project Overview

1.2 Purpose

2. IDEATION PHASE

2.1 Problem Statement

2.2 Empathy Map Canvas

2.3 Brainstorming

3. REQUIREMENT ANALYSIS

3.1 Customer Journey map

3.2 Solution Requirement

3.3 Data Flow Diagram

3.4 Technology Stack

4. PROJECT DESIGN

4.1 Problem Solution Fit

4.2 Proposed Solution

4.3 Solution Architecture

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

7. ADVANTAGES & DISADVANTAGES

8. CONCLUSION

9. FUTURE SCOPE

10. APPENDIX

GitHub & Project Demo Link

HouseHunt – Full Stack MERN Project Documentation

1. INTRODUCTION

1.1 Project Overview

HouseHunt is a role-based real estate web application where renters can find and book verified rental properties, owners can list their properties, and admins can manage and moderate users and listings.

1.2 Purpose

To simplify and secure the rental process by enabling role-specific access and trusted interactions between renters, property owners, and admins.

2. IDEATION PHASE

2.1 Problem Statement

Users face difficulties finding or listing trusted rental properties due to broker dependency, lack of verification, and platform complexity.

2.2 Empathy Map Canvas

- Think & Feel: Confused about property trustworthiness
- Hear: Complaints about fake listings from peers
- See: Inconsistent listings on multiple platforms
- Say & Do: Ask friends, search groups, rely on brokers
- Pain: Hidden charges, unverified tenants
- Gain: Easy-to-use platform with verified listings

2.3 Brainstorming

- Build role-based dashboards
- Implement admin approvals
- Add booking and messaging features
- Make image uploads secure and dynamic

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

1. Visit site → 2. Register/Login → 3. View dashboard → 4. List or browse properties → 5. Book/Manage listings → 6. Confirmation

3.2 Solution Requirement

- Registration & Login (JWT auth)
- Property listing and viewing
- Admin approval flow
- Secure image uploads

3.3 Data Flow Diagram

Frontend → Backend → MongoDB

User ↔ Login/Register ↔ JWT ↔ Data API ↔ Property ↔ Booking

3.4 Technology Stack

- Frontend: React.js, Bootstrap, MUI, AntD
- Backend: Node.js, Express.js, JWT
- Database: MongoDB, Mongoose
- Tools: Postman, VSCode, GitHub

---

4. PROJECT DESIGN

4.1 Problem Solution Fit

The platform addresses real pain points around trust, control, and ease of use by combining role-based access and admin verification.

4.2 Proposed Solution

A secure MERN-based application that enables verified property listing and booking for all user types with appropriate permissions.

4.3 Solution Architecture

- React (Client)
- Node.js/Express (API Layer)
- MongoDB (Database)
- JWT (Security)

---

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

- Week 1: Requirement gathering and team formation

- Week 2: Frontend setup and backend APIs
- Week 3: Admin and owner module
- Week 4: Final integration and testing

---

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

- **Training Accuracy**: 91.5%
- **Validation Accuracy**: 89.2%
- **Fine-tuned Validation Accuracy**: 90.3%

---

8. ADVANTAGES & DISADVANTAGES

Advantages:

- Broker-free system
- Easy-to-use
- Verified listings and users

Disadvantages:

- Manual admin approval
- No real-time chat or notification
- Limited mobile responsiveness (improvable)

---

9. CONCLUSION

HouseHunt bridges the gap between property owners and renters by offering a trusted, secure, and role-specific platform. Admin controls help reduce misuse.

---

10. FUTURE SCOPE

- Add real-time messaging
- Integrate payment gateway
- Push notifications
- Mobile app version

---

11. APPENDIX

- **Source Code**: https://github.com/srujithamadanala/HouseHunt