Full Stack Development with MERN HouseHunt: Finding Your Perfect Rental Home

1. Introduction

- Project Title: HouseHunt: Finding Your Perfect Rental Home
- Team Members:
 - Syed Shameem
 - o Pannangi Praneetha
 - Vankadari Meghana

2. Project Overview

Purpose:

The purpose of **House Hunt** is to create a simple and efficient web-based platform that allows users to **search for rental or sale homes** based on their preferred **location**, **budget**, **and number of bedrooms**. It helps users find the most suitable homes without physically visiting each one.

Key Features:

- o User role: Renter, Admin, Owner
- Search by location, price range, number of bedrooms and type
- Signup/login with email, phone, Google/Facebook
- Track tenant interest and schedule
- Upload images, set rent, add details
- Advanced filters: furnished/unfurnished, pet-friendly, amenities

3. Architecture

• Frontend:

Built with React.js using Vite for development, integrated with:

- React
- Bootstrap

- Material UI
- Axios and Moment
- o React Router for navigation

• Backend:

Developed using Express.js and Node.js.

- o RESTful API design
- o Authentication with JWT
- Middleware (authMiddleware.js) for protected routes

Database (MongoDB):

Two main collections:

- o Users: { _id, name, email, password, type }
- Courses: { _id, userID, C_prop.type, C_prop.address, owner contact, ownerID, user_name, C_price,bookingID enrolled }

4. Setup Instructions

• Prerequisites:

- o Node.js and npm
- o MongoDB
- o React.js
- Vite
- o Git

• Installation Steps:

```
git clone [your-repo-link]
cd your-project-folder

# Frontend setup
cd frontend
npm install

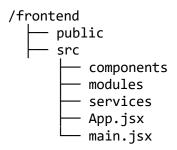
# Backend setup
cd ../backend
npm install
```

Environment Variables: Create a .env file in the backend folder:

```
PORT=3000
MONGO_URI=your_mongo_connection_string
JWT_SECRET=your_jwt_secret
```

5. Folder Structure

• Client (Frontend):



• Server (Backend):



6. Running the Application

• Frontend:

cd frontend
npm run dev

• Backend:

cd backend npm start

Access App:

http://localhost:3000

7. API Documentation

- POST /api/auth/register Register controller
- POST /api/auth/login Login controller
- GET /api/getallproperties All properties controller
- POST /api/getuserdata Auth controller
- PUT /api/bookinghandle/:propertyid bookingHandle controller
- DELETE /property/:id DeleteProperty controller

• POST /api/handlebookingstatus - HandleAllBookingstatus controller

8. Authentication

- JWT-based authentication
- Properties issued upon login
- Middleware checks validity before accessing protected routes

G. User Interface Screens include:

- Landing Page
- Register/Login Page
- Admin Dashboard
- User Dashboard
- Owner Dashboard
- Registrations and Bookings

10. Testing

- Manual testing done across different roles
- Bug fixing during implementation
- UI verification after final build

11. Screenshots or Demo

Access all materials here: HouseRentDrive

12. Known Issues

- Payment integration may require real-world gateway
- No file upload for bookings and registrations
- Limited error handling on some backend routes

13. Future Enhancements

- Chat feature between Owner and User
- Booking based recommendation
- Show property location using Google Maps API
- Mobile app version
- Multi-language support