

# Full Stack Development with MERN

## Project Documentation

### 1. Introduction

**Project Title:** House Hunt – Your Perfect Rental Home Finder

**Team Members:**

Team Leader : Veluri Laharika Choudary

Team member : Kotha Akshaya

Team member : Devathi Reddy Chandana

Team member : Sharabu Nandu

Team member : Bandaru Yeswanthi

### 2. Project Overview

**Purpose:** To create a streamlined, user-friendly platform that connects potential tenants with property owners, simplifying the process of finding and listing rental homes.

**Features:** User authentication (for tenants and owners), advanced property search with filters (location, price, bedrooms), detailed property listings, user profile management, and a dedicated dashboard for property owners to manage their listings.

### 3. Architecture

**Frontend:** Developed using React for a dynamic, interactive, and responsive user interface to browse listings and manage profiles.

**Backend:** Built with Node.js and the Express framework to handle API requests, user management, and property data logic.

**Database:** MongoDB (connected via MongoDB Atlas) serves as the database to store all application data, including user credentials, property details, and inquiries.

### 4. Setup Instructions

**Prerequisites:** Node.js, npm (Node Package Manager), Git, and a MongoDB Atlas account are required.

### 5. Folder Structure

**frontend:** Contains the complete React frontend source code, including components, pages, and styles.

**root (/):** Contains the backend server implemented with Express, including models, routes, controllers, and middleware.

### 6. Running the Application

**Backend:** Run `npm run dev` from the project's root directory.

**Frontend:** Run `npm start` from inside the `/frontend` directory.

## **7. API Documentation**

The backend exposes REST APIs for essential operations. Key endpoints include:

POST /api/auth/register - User registration (for Tenants and Owners)

POST /api/auth/login - User login

GET /api/properties - Retrieve a list of all rental properties

GET /api/properties/:id - Retrieve a single property by its ID

POST /api/properties - Add a new property listing (Owner only)

## **8. Authentication**

Standard email and password-based login for both Tenants and Property Owners.

The system uses JWT (JSON Web Tokens) for securing API routes and maintaining user sessions for enhanced security.

## **9. User Interface**

Screens provided in the system include:

Login & Register Page

Home Page (Property Listing & Search)

Property Details Page

User Profile/Dashboard

Owner Dashboard (for managing properties)

Add New Property Page

## **10. Testing**

Backend: API endpoints can be tested using tools like Postman for validation and debugging.

Frontend: The user interface and functionality can be tested manually or by using the browser's built-in Chrome DevTools.

## **11. Screenshots or Demo**

Screenshots and demo videos can be provided separately in the project folder to showcase the application's functionality.