**HOUSEHUNT**

**HouseHunt MERN - Find Your Perfect Rental Home**

HouseHunt MERN is a full-stack web application designed to help users find rental homes. Users can browse listings, filter them based on various criteria, register accounts, log in, post their own rental ads, and manage their ads. This project is built using the MERN stack (MongoDB, Express.js, React, Node.js).

**Table of Contents**

* [Features](https://github.com/vamsi-424/HOUSEHUNT?tab=readme-ov-file#features)
* [Tech Stack](https://github.com/vamsi-424/HOUSEHUNT?tab=readme-ov-file#tech-stack)
* [Project Structure](https://github.com/vamsi-424/HOUSEHUNT?tab=readme-ov-file#project-structure)
* [Prerequisites](https://github.com/vamsi-424/HOUSEHUNT?tab=readme-ov-file#prerequisites)
* [Getting Started](https://github.com/vamsi-424/HOUSEHUNT?tab=readme-ov-file#getting-started)
  + [1. Backend Setup](https://github.com/vamsi-424/HOUSEHUNT?tab=readme-ov-file#2-backend-setup)
  + [2. Frontend Setup](https://github.com/vamsi-424/HOUSEHUNT?tab=readme-ov-file#3-frontend-setup)
  + [3. Running the Application](https://github.com/vamsi-424/HOUSEHUNT?tab=readme-ov-file#4-running-the-application)
* [Seeding Initial Data](https://github.com/vamsi-424/HOUSEHUNT?tab=readme-ov-file#seeding-initial-data)

**Features**

* **User Authentication:** Secure user registration and login using JWT (JSON Web Tokens).
* **Browse Listings:** View all available rental listings.
* **Filter & Search:**
  + Filter listings by location (text search).
  + Filter by price range (min/max).
  + Filter by number of bedrooms (BHK).
  + Filter by property type (Apartment, House, etc.).
  + Filter by furnished status.
  + Filter by who listed the ad (Owner, Dealer, Builder).
* **Create Listings:** Authenticated users can post new rental advertisements with details like title, location, price, property specifics, images (URL-based), and description.
* **Manage Listings:** Users can edit and delete their own rental listings.
* **View Poster Information:** See which user posted a particular ad.
* **Responsive Design:** Basic responsiveness for usability across different device sizes.

**Tech Stack**

**Backend:**

* **Node.js:** JavaScript runtime environment.
* **Express.js:** Web application framework for Node.js.
* **MongoDB:** NoSQL database for storing user and listing data.
* **Mongoose:** ODM (Object Data Modeling) library for MongoDB and Node.js.
* **JSON Web Tokens (JWT):** For user authentication.
* **bcryptjs:** For hashing user passwords.
* **dotenv:** For managing environment variables.
* **cors:** For enabling Cross-Origin Resource Sharing.

**Frontend:**

* **React:** JavaScript library for building user interfaces.
* **React Hooks:** For state management and side effects in functional components.
* **Axios:** Promise-based HTTP client for making API requests.
* **CSS:** For styling the application (global App.css and index.css).
* **Create React App:** For scaffolding the React project.

**Development Tools:**

* **Nodemon:** For automatic server restarts during backend development.
* **Postman (or similar):** For testing backend API endpoints.
* **Git & GitHub:** For version control and code hosting.

**Project Structure# HouseHunt MERN - Find Your Perfect Rental Home**

HouseHunt MERN is a full-stack web application designed to help users find rental homes. Users can browse listings, filter them based on various criteria, register accounts, log in, post their own rental ads, and manage their ads. This project is built using the MERN stack (MongoDB, Express.js, React, Node.js).

**Table of Contents**

* [Features](https://github.com/vamsi-424/HOUSEHUNT?tab=readme-ov-file#features)
* [Tech Stack](https://github.com/vamsi-424/HOUSEHUNT?tab=readme-ov-file#tech-stack)
* [Project Structure](https://github.com/vamsi-424/HOUSEHUNT?tab=readme-ov-file#project-structure)
* [Prerequisites](https://github.com/vamsi-424/HOUSEHUNT?tab=readme-ov-file#prerequisites)
* [Getting Started](https://github.com/vamsi-424/HOUSEHUNT?tab=readme-ov-file#getting-started)
  + [1. Backend Setup](https://github.com/vamsi-424/HOUSEHUNT?tab=readme-ov-file#2-backend-setup)
  + [2. Frontend Setup](https://github.com/vamsi-424/HOUSEHUNT?tab=readme-ov-file#3-frontend-setup)
  + [3. Running the Application](https://github.com/vamsi-424/HOUSEHUNT?tab=readme-ov-file#4-running-the-application)
* [Seeding Initial Data](https://github.com/vamsi-424/HOUSEHUNT?tab=readme-ov-file#seeding-initial-data)

**Features**

* **User Authentication:** Secure user registration and login using JWT (JSON Web Tokens).
* **Browse Listings:** View all available rental listings.
* **Filter & Search:**
  + Filter listings by location (text search).
  + Filter by price range (min/max).
  + Filter by number of bedrooms (BHK).
  + Filter by property type (Apartment, House, etc.).
  + Filter by furnished status.
  + Filter by who listed the ad (Owner, Dealer, Builder).
* **Create Listings:** Authenticated users can post new rental advertisements with details like title, location, price, property specifics, images (URL-based), and description.
* **Manage Listings:** Users can edit and delete their own rental listings.
* **View Poster Information:** See which user posted a particular ad.
* **Responsive Design:** Basic responsiveness for usability across different device sizes.

**Tech Stack**

**Backend:**

* **Node.js:** JavaScript runtime environment.
* **Express.js:** Web application framework for Node.js.
* **MongoDB:** NoSQL database for storing user and listing data.
* **Mongoose:** ODM (Object Data Modeling) library for MongoDB and Node.js.
* **JSON Web Tokens (JWT):** For user authentication.
* **bcryptjs:** For hashing user passwords.
* **dotenv:** For managing environment variables.
* **cors:** For enabling Cross-Origin Resource Sharing.

**Frontend:**

* **React:** JavaScript library for building user interfaces.
* **React Hooks:** For state management and side effects in functional components.
* **Axios:** Promise-based HTTP client for making API requests.
* **CSS:** For styling the application (global App.css and index.css).
* **Create React App:** For scaffolding the React project.

**Development Tools:**

* **Nodemon:** For automatic server restarts during backend development.
* **Postman (or similar):** For testing backend API endpoints.
* **Git & GitHub:** For version control and code hosting.

**Project Structure**

house-hunt-mern/ ├── backend/ │ ├── config/ │ │ └── db.js │ ├── controllers/ │ │ ├── userController.js │ │ └── listingController.js │ ├── middleware/ │ │ ├── authMiddleware.js │ │ └── errorMiddleware.js │ ├── models/ │ │ ├── UserModel.js │ │ └── ListingModel.js │ ├── routes/ │ │ ├── userRoutes.js │ │ └── listingRoutes.js │ ├── utils/ # (Optional, if you have seeder here) │ │ └── seeder.js │ ├── .env │ ├── package.json │ └── server.js └── frontend/ ├── public/ ├── src/ │ ├── App.css │ ├── App.js │ ├── index.css │ ├── index.js │ ├── components/ │ │ ├── Auth/ │ │ ├── Filters/ │ │ ├── Layout/ │ │ └── Listings/ │ ├── services/ │ │ ├── authService.js │ │ └── listingService.js ├── package.json └── ...

**Prerequisites**

* Node.js (v14.x or later recommended)
* npm (Node Package Manager) or yarn
* MongoDB (local instance or a cloud service like MongoDB Atlas)
* Git

**1. Backend Setup**

Navigate to the backend directory and install dependencies: cd backend npm install

Create a .env file in the backend directory with the following content (replace placeholder values): PORT=5001 MONGO\_URI=mongodb://localhost:27017/househuntDB # Or your MongoDB Atlas connection string JWT\_SECRET=yourSuperStrongAndSecretJWTKeyPleaseChangeThis

**2. Frontend Setup**

Navigate to the frontend directory and install dependencies: cd ../frontend # If you are in the backend directory or cd frontend (if you are in the root house-hunt-mern directory) npm install

The frontend uses a proxy to communicate with the backend during development. This is already configured in frontend/package.json: "proxy": "[http://localhost:5001](http://localhost:5001/)"

**3. Running the Application**

You'll need two separate terminal windows: Terminal 1: Start the Backend Server cd backend npm run server

The backend server should start on [http://localhost:5001](http://localhost:5001/) (or the port specified in your .env). Terminal 2: Start the Frontend React App cd frontend npm start

The React development server will start, and your application should open in your browser, typically at [http://localhost:3000](http://localhost:3000/).

**Seeding Initial Data**

To populate the database with initial sample users and listings: Ensure your MongoDB server is running. Navigate to the backend directory in your terminal. Run the seeder script: npm run seed

This will clear existing users and listings and then insert new sample data. Passwords for sample users will be automatically hashed.