

Full Stack Development with MERN

Frontend Development Report

Date	12-July-2024
Team ID	SWTID1720069738
Project Name	House Hunt
Maximum Marks	10 Marks

Project Title: House Hunt

Date: 12-07-2024.

Prepared by: L.Chaitanya, G.Bhargav

Objective

The objective of this report is to document the frontend development progress and key aspects of the user interface implementation for the “House Hunt” project.

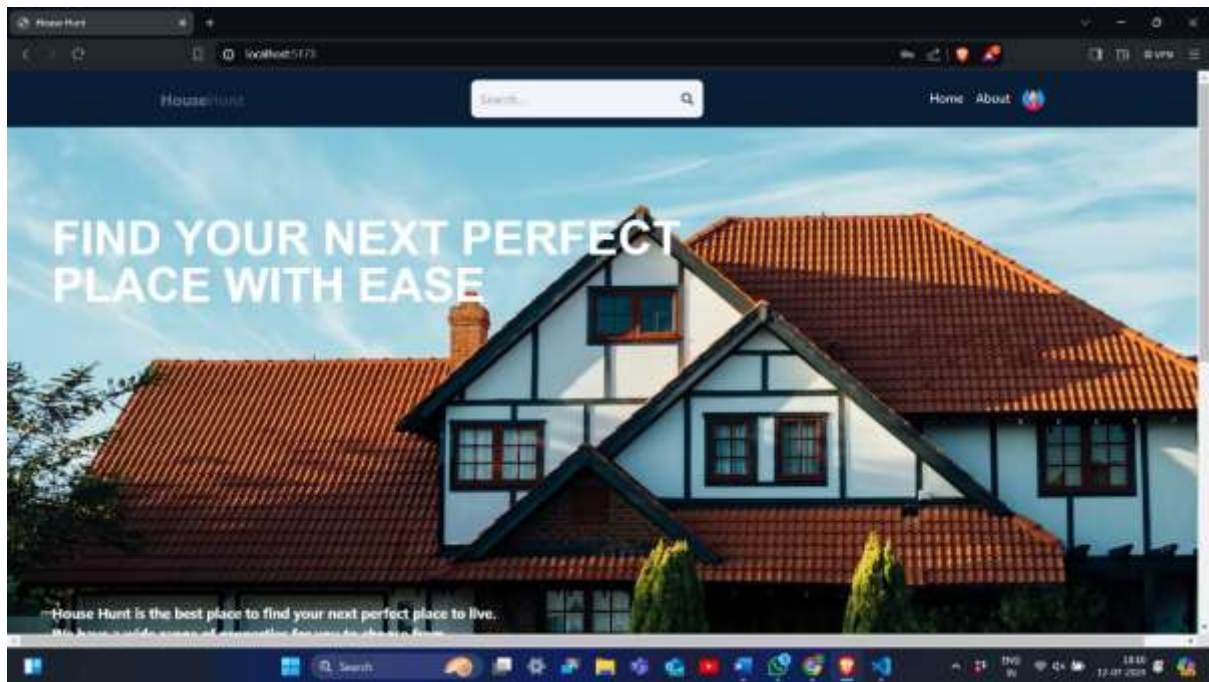
Technologies Used

- **Frontend Framework:** React.js
- **State Management:** Redux
- **UI Framework/Libraries:** Tailwind CSS, Swiper, React-Icons
- **Routing Library:** React Router
- **Build Tool:** Vite
- **Authentication Library:** Firebase

Project Structure

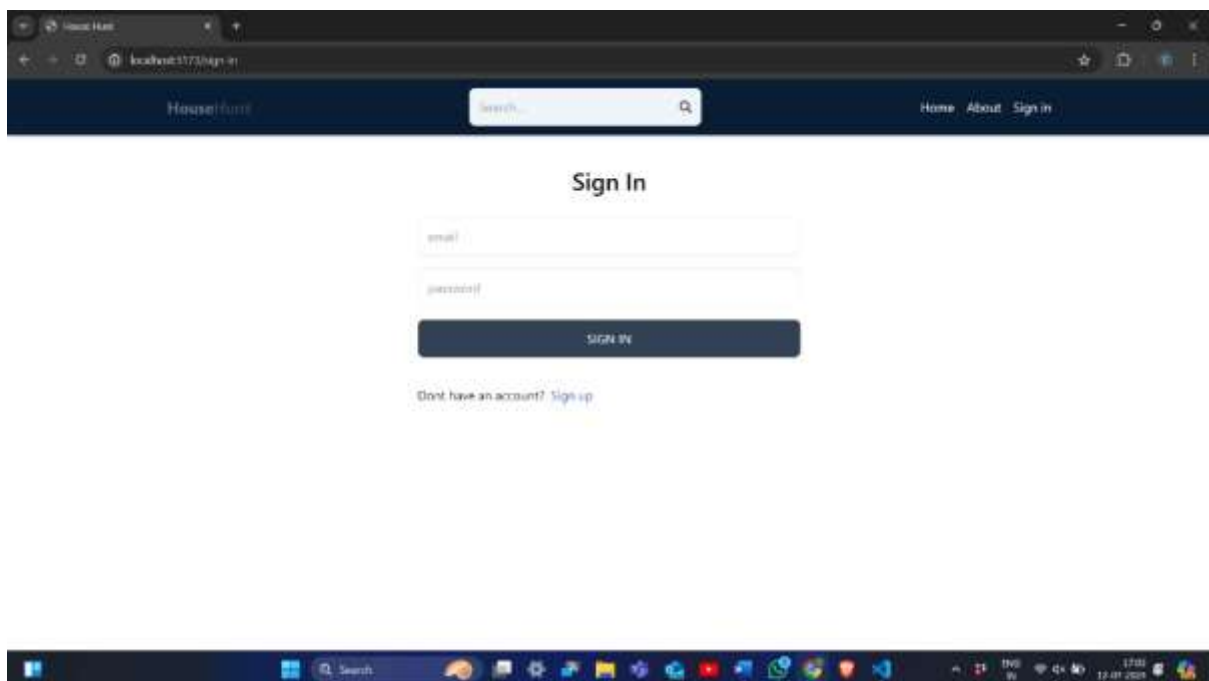
Home Page:

House-Hunt make it easy to find your next perfect home. Whether you're looking to rent or buy, our platform offers a wide range of properties to suit your needs. Start browsing now and find your dream place with ease.



Sign-in page:

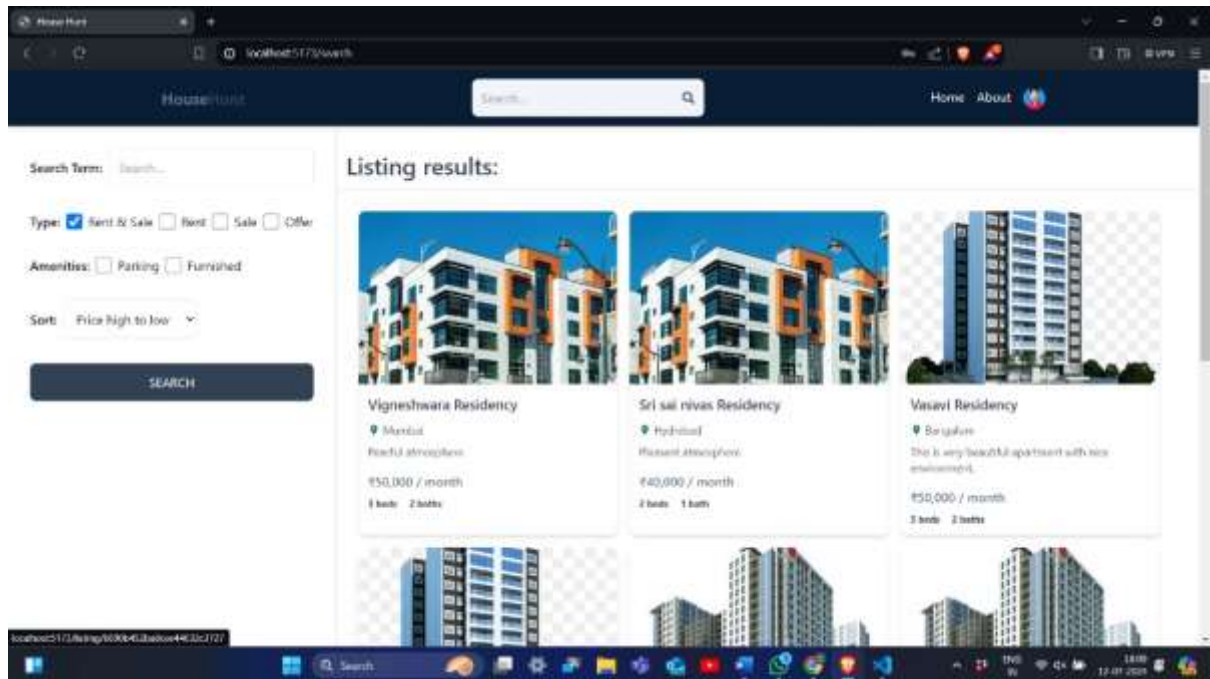
Sign in to your House-Hunt account to unlock exclusive listings and personalized recommendations. Your information is secure with us, and our site is designed to be user-friendly. New here? Sign up today and start discovering the best property deals around.



Search Results Page:

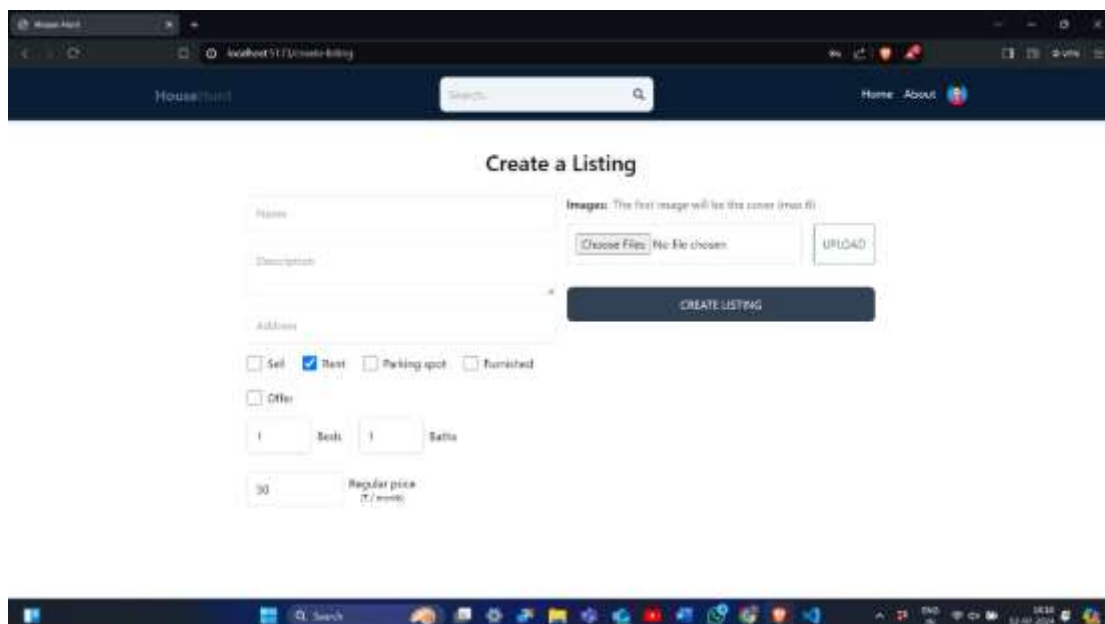
- **Description:** We can check out the detailed property listings on House-Hunt using our advanced search feature. You can filter results by type, amenities, and price to

find exactly what you're looking for. Each listing gives you all the important details like location, price, and more. Start your search now and find the perfect home effortlessly.



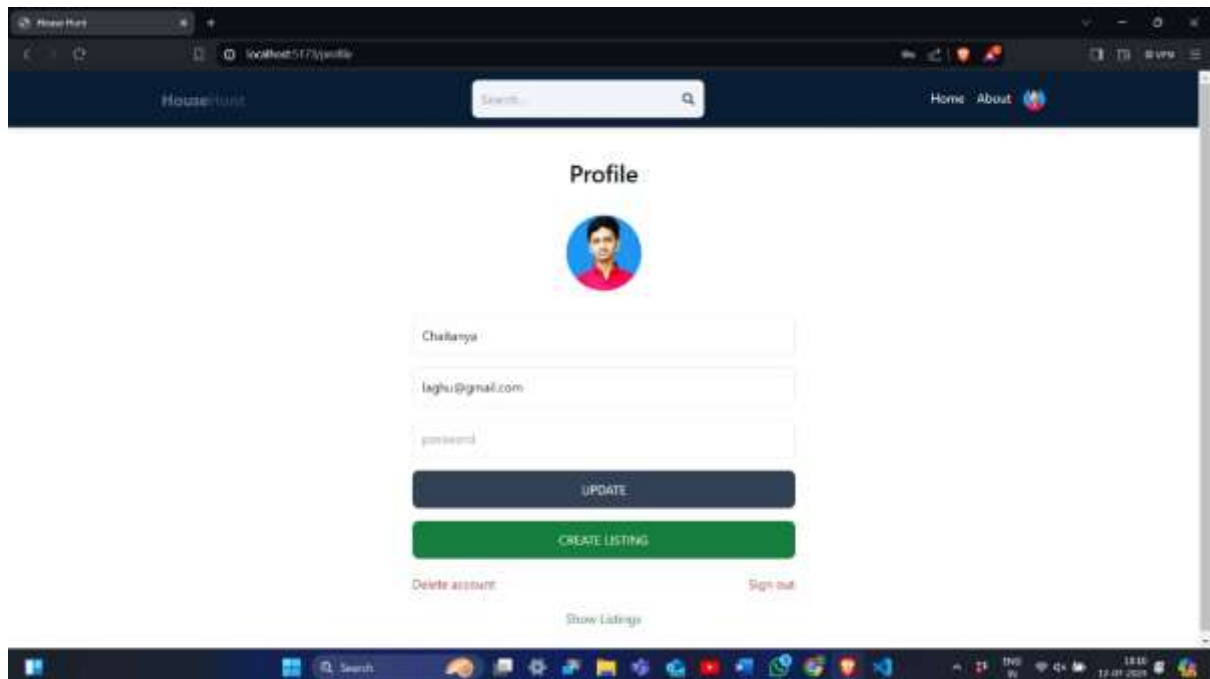
Create Listings:

In the create listings, the user can add the property details like name, description, address and we can check-in the necessary facilities available there. And also user can add the property images minimum of 6. And after entering details they can select Create listings button.



Profile:

In the profile section, the users can edit their profile and also they can add their profile image .And also they can edit their listings ,which they have uploaded their property previously. And also user can Sign-out from the account.



Key Components

1. App.js

- Manages routing and main application layout.

2. components

- **list**
 - Displays the list of properties.
- **map**
 - Integrates maps for property location visualization using React-Leaflet.
- **navbar**
 - Provides the main navigation bar for the application.
- **pin**
 - Represents a pin on the map for property locations.
- **searchBar**
 - Allows users to search for properties based on criteria.
- **slider**
 - Displays property images in a slider format.
- **uploadWidget**
 - Handles property image uploads.

3. **pages**

- Includes different pages for the web application.

4. **context**

- Provides context and state management using Context API and Redux.

5. **routes**

- Defines routes and layout components for the application.

6. **lib**

- Contains utility functions and loaders for data fetching.

7. **apiRequest.js**

- Configures axios for API requests with base URL and credentials.

Routing

Routing is managed using React Router. Here are the main routes:

/routes:

- **layout**
 - Defines the main layout and structure for nested routes.
- **homePage**
 - Displays the homepage with featured listings, recent offers, and search functionality.
- **listPage**
 - Displays the list of properties with filtering options.
- **login**
 - Provides the login page for user authentication.
- **newPostPage**
 - Allows users to add new property listings.
- **profilePage**
 - Displays the user's profile with their listed properties.
- **profileUpdatePage**
 - Enables users to update their profile information.
- **register**
 - Provides the registration page for new users.
- **singlePage**
 - Shows detailed information about a single property.

State Management (If Applicable)

State management is achieved using Redux ToolKit

Integration with Backend

The frontend communicates with the backend APIs hosted on `http://localhost:3000/api`. Key endpoints include:

- **POST /api/auth/signup**
 - Handles user registration.
- **POST /api/auth/signin**
 - Handles user authentication.
- **POST /api/auth/google**
 - Handles authentication via Google.
- **GET /api/auth/signout**
 - Logs the user out by clearing authentication cookies.
- **POST /api/listing/create**
 - Creates a new property listing.
- **DELETE /api/listing/delete/**
 - Deletes a property listing.
- **POST /api/listing/update/**
 - Updates a property listing.
- **GET /api/listing/get/**
 - Retrieves a single property listing by ID.
- **GET /api/listing/get**
 - Retrieves multiple property listings with filtering options.
- **POST /api/user/update/**
 - Updates user profile information.
- **DELETE /api/user/delete/**
 - Deletes a user account.
- **GET /api/user/listings/**
 - Retrieves all property listings created by a specific user.
- **GET /api/user/**
 - Retrieves a user's profile information.

User Interface (UI) Design

- The UI design follows a modern, responsive approach.
- Implemented using Tailwind CSS for styling and includes interactive maps with Swiper.
- Rich text editing capabilities are provided by Swiper.

Third-Party Integration

Third-party integrations in the "House Hunt" project contribute to its functionality and user experience:

- **Firebase**
 - Provides authentication services and real-time database capabilities.

- **React-Leaflet**
 - Integrates interactive maps for property location visualization.
- **Swiper**
 - Enhances the UI with image sliders and carousel functionalities.
- **Axios**
 - Handles HTTP requests to communicate with the backend API.
- **Tailwind CSS**
 - Utilizes utility-first CSS framework for responsive and customizable styling.