

Full Stack Development with MERN

Frontend Development Report

| | |
|---------------|---|
| Date | 26-06-2025 |
| Team ID | LTVIP2025TMID59418 |
| Project Name | House Hunt : Finding your perfect rental home |
| Maximum Marks | 10 Marks |

Project Title: House Hunt : Finding your perfect rental home

Date: 26-06-2025

Prepared by: Sureboina Durga Bhavani
Sanivarapu Jyothi
Rekha Sri Boyina
Savu V V R Prasad

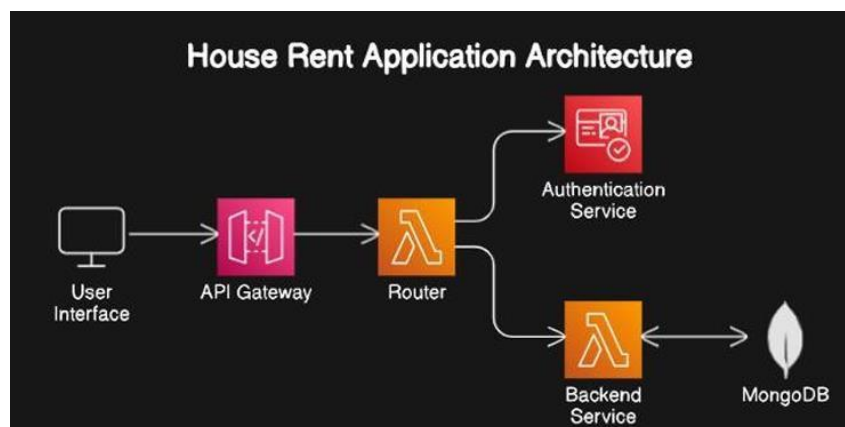
Objective

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

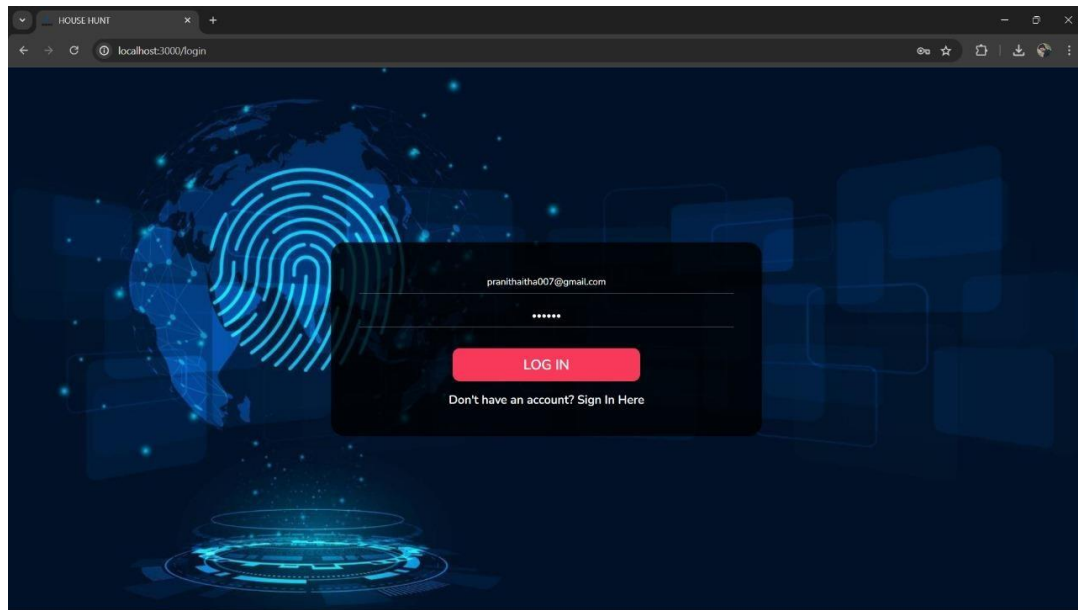
Technologies Used

- **Frontend Framework:** React.js, Html and CSS
- **UI Framework/Libraries:** Bootstrap and Html
- **API Libraries:** Axios

Project Structure

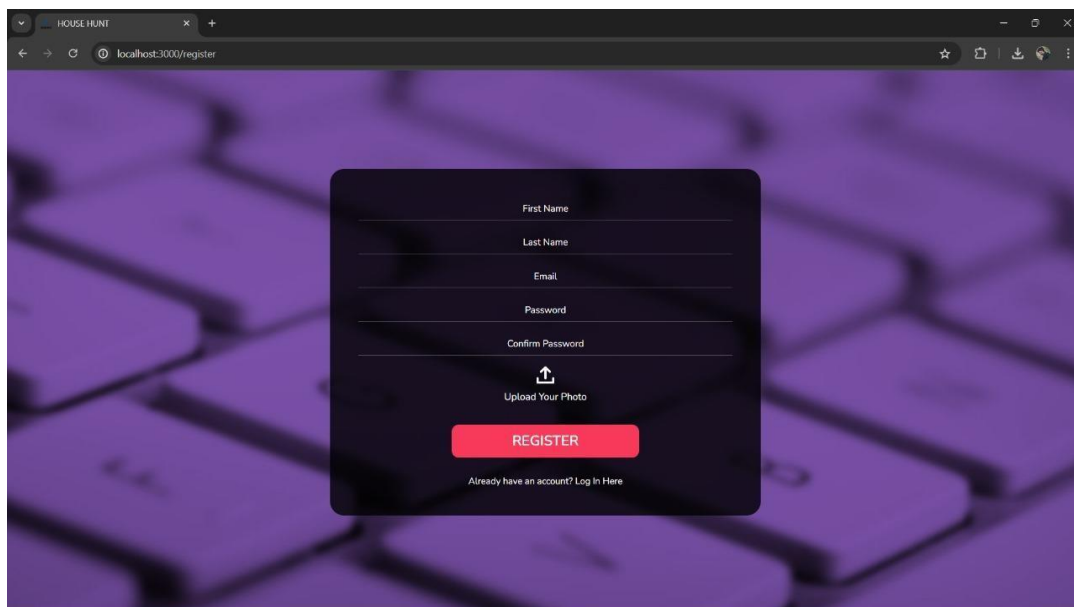


Login Page:



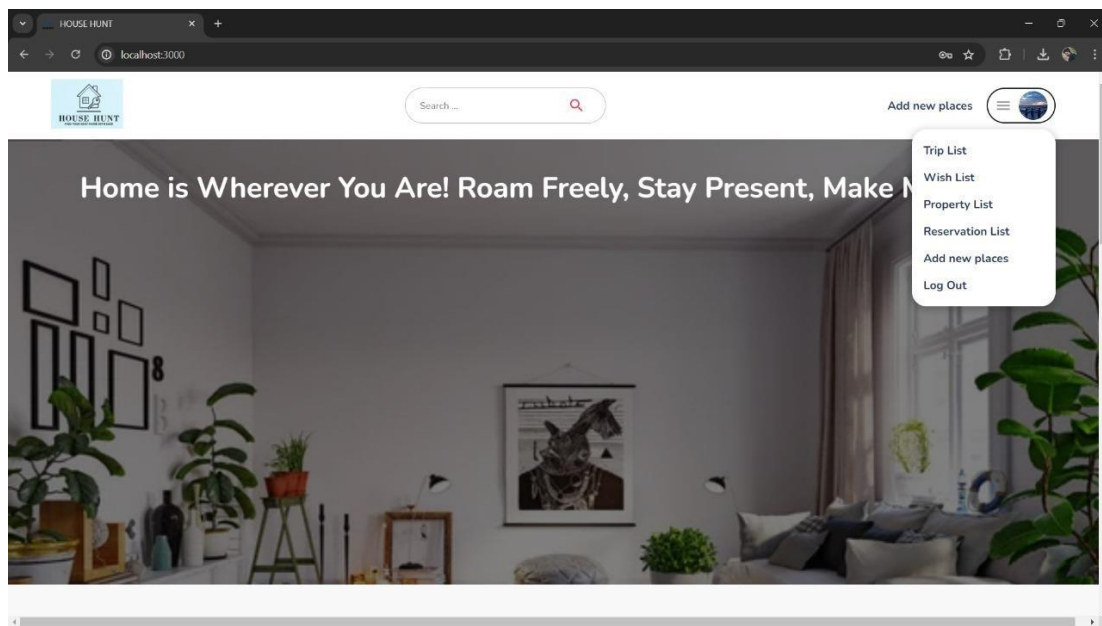
This is the Login page where users can login to their existing accounts by entering their email-id and password. This page also provides users with an option that helps new users create an account

Register:



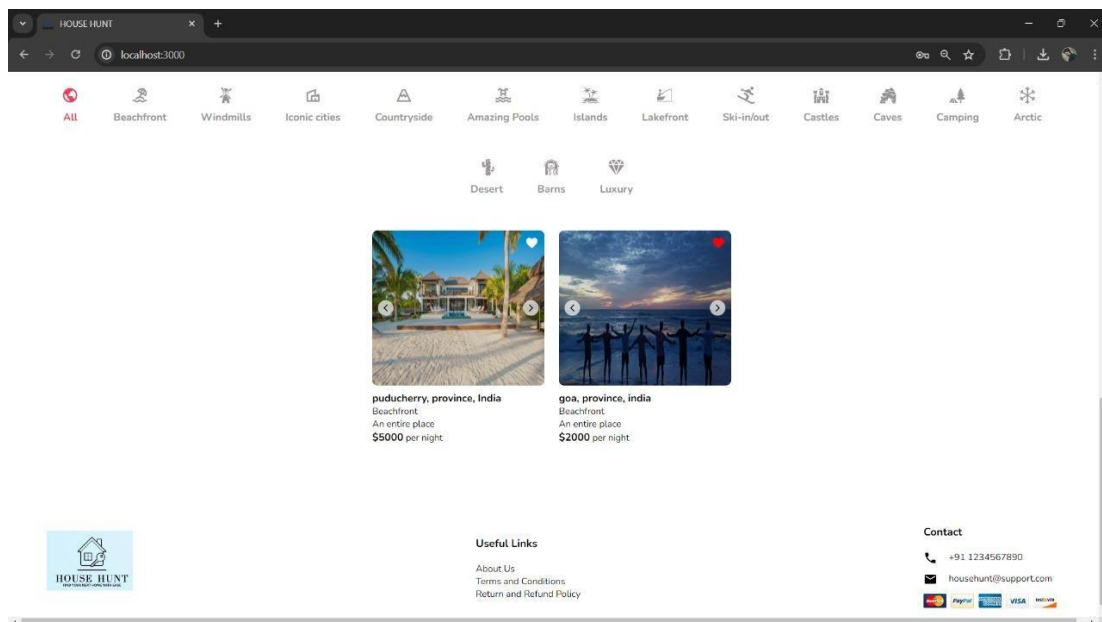
This is the Register page that allows new users create new account on entering their details like name, email, password and a photo of them

Home Page:



This is the main page of House Hunt, where you can easily search for rental homes. Together with user-friendly search features and tailored recommendations, you may find carefully chosen home listings here that are tailored to your preferences. Our platform offers cutting-edge technologies and committed assistance to make your rental experience easier.

Listings:



Reach a large audience of possible tenants by creating and managing house listings on House Hunt with ease. Make sure your rental home shines out to potential tenants by showcasing it with thorough descriptions, images, and features.

Key Components

1. App.js

- **App.js** Importing various components that make up the application, such as header, footer, home page, listing page, etc.
- Responsible for routing and main application layout.

2. Components

- Each Component focuses on a specific part of the UI, allowing us to develop and test them independently.
- Every Component typically has its own CSS file (e.g., Header.css, Footer.css) to encapsulate its styles, ensuring modular and maintainable code.

3. Pages

Includes different pages for Web App like

- Homepage
- Register
- Login
- Search
- Reservation
- Trip list
- Wishlist

Routing

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

- **User** - Manage user accounts, view their booking history, and update their personal information.
- **Listings** - Displays a list of all available properties that match the user's search criteria.
- **Authentication** - Authenticates user based on their username and password
- **Booking** – Responsible for managing bookings

Integration with Backend

The frontend communicates with the backend APIs hosted on [backend URL]. Key endpoints include:

- **GET /api/data** - Retrieves data for display.
- **POST /api/user/login** - Handles user authentication.
- **PUT/api/user/update**- Updates, adds user data in the database
- **DELETE/api/user/delete**- Delete user account and clear search history

User Interface (UI) Design

- The UI design follows

User Personas: Defining the target audience and their needs. Tailoring the design to meet these requirements.

Media Queries: Using CSS media queries to adjust layouts, font sizes, and other elements based on screen size.

Reusable Components: Creating a library of reusable UI components (e.g., buttons, forms, modals) to ensure consistency across the app.

Clear Navigation: Using a clear and simple navigation structure. Including a header with a navigation bar and a footer with additional links.

Color and Contrast: Using color and contrast to highlight important elements and guide users' attention. Ensure accessibility by meeting WCAG contrast standards.

- Implemented using

Bootstrap: Bootstrap is a popular front-end framework that provides pre-designed UI components and a responsive grid system. It's particularly useful for quickly creating a responsive and aesthetically pleasing design.

Header and Footer: Using Header and Footer components for margins and styling

Third-Party Integrations:

- axios for requesting content via HTTP.
- Dompurify, which cleans HTML inputs.
- Socket.io-client facilitates instantaneous communication.
- For displaying relative timestamps, use timeago.js.