AIRBNB PROJECT

#navbar\_ (import--

container

logo

search

usermenu)

usermenu \_ (import--

avatar

menuitems

)

#LAYOUT-(import--

navbar

ClientOnly

ToasterProvider

RegisterModal

)

#providers folder include

ToasterProvider.tsx

#hooks folder include useRegisterModal

#components folder include input.tsx

#models folder include (Modal.tsx , RegisterModal.tsx)

#and rest files or folder are--

-Avatar.tsx

-Button.tsx

-ClientOnly.tsx

-Heading.tsx

Navbar(

-Logo.tsx

-MenuItems.tsx

-Navbar.tsx

-Search.tsx

-UserMenu.tsx

)

### ****Main App Layout and Structure****

#### layout.tsx

* The **main layout component** of your app.
* Wraps all pages with **global structure**, including things like the Navbar, RegisterModal, and ToasterProvider.
* Helps in initializing modals and toast notifications globally so they work across all pages.

#### page.tsx

* Your **homepage** or root route (/) file.
* Uses ClientOnly to ensure content only renders on the client-side.
* Includes layout wrappers like Container and shows placeholder content like "Hello Airbnb" or potentially a hero component in the future.

### 🔷 ****Modal and Form Logic****

#### RegisterModal.tsx

* A **registration modal** that allows users to sign up.
* Uses the Modal component as its UI container.
* Uses useRegisterModal (a Zustand store) to control whether the modal is open or closed.
* Uses Input.tsx for form fields and react-hook-form for form handling.

#### Modal.tsx

* The **generic modal component** used by RegisterModal.
* Handles modal transitions, close logic, and renders dynamic content passed as props.
* Has support for **primary and secondary actions**, body and footer content.

#### useRegisterModal.ts

* A **Zustand store** used to control the open/close state of the registration modal globally.
* Used by RegisterModal and initialized in layout.tsx.

### 🔷 ****UI Components****

#### Button.tsx

* A **reusable button component** used inside modals and forms.
* Supports variations like outlined buttons, disabled states, and custom actions.

#### Input.tsx

* A **reusable input component** for form fields.
* Supports features like label animation, validation error styling, and optional currency formatting (formatPrice).

#### Heading.tsx

* A simple **heading component** with a title and optional subtitle.
* Used in modals like the registration modal to show section headers.

### 🔷 ****Client-Side Helpers****

#### ClientOnly.tsx

* Ensures its children only render on the **client-side** (not during server-side rendering).
* Useful for components like modals or toasts that rely on client-only hooks or behavior.

#### Container.tsx

* A responsive **layout wrapper** to control max width and padding.
* Used to give a consistent layout across pages like page.tsx.

### 🔷 ****Toast Notifications****

#### ToasterProvider.tsx

* Provides the react-hot-toast toaster globally.
* Enables notifications across the app (e.g., show error/success toast in forms).

App Structure

├── layout.tsx

│ ├── RegisterModal

│ │ ├── Modal

│ │ │ ├── Button

│ │ │ └── Heading

│ │ └── Input

│ ├── Navbar (if added)

│ └── ToasterProvider

│

├── page.tsx

│ └── ClientOnly

│ └── Container

│

├── useRegisterModal (global state for RegisterModal)

**✅ What Happens When RegisterModal Opens?**

1. User clicks a button (possibly from Navbar or other UI).
2. useRegisterModal.onOpen() is called.
3. RegisterModal detects isOpen === true and renders the Modal.
4. The Modal displays:
   * Header from Heading
   * Input fields from Input
   * Buttons from Button
5. On submit, react-hook-form validates the data.
6. If valid, you can show a toast via react-hot-toast using ToasterProvider.