Location/Address Flow Assignment

Overview:

In this assignment, you are tasked with building a Location/Address flow similar to the one illustrated in the provided images. Your solution should allow users to select and save their delivery location, using Google Maps API to implement location search and geolocation.

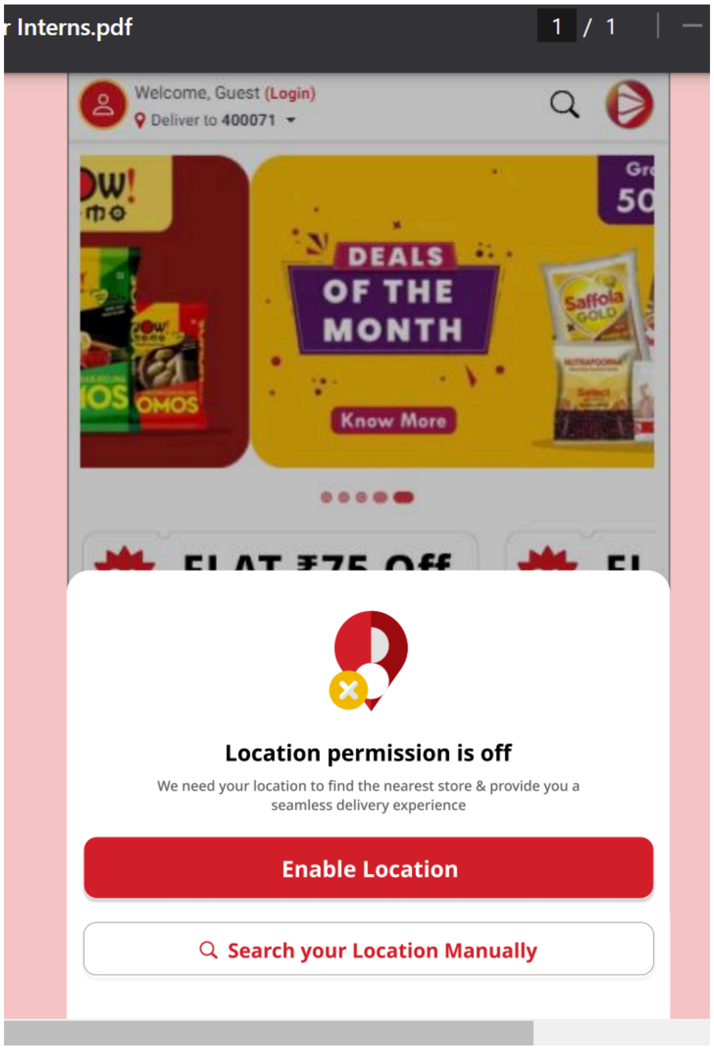
You are expected to showcase your skills in **React** for the frontend and **Node.js** for the backend. You can use Google Maps APIs for map functionalities.

Steps to Follow:

Step 1: Setup & Requirements

- **Frontend**: You must use React for creating the UI.
- **Backend**: Implement backend APIs using Node.js.
- **Map Integration**: Use Google Maps API for location functionalities.
- **State Management**: You are free to use state management tools like Redux or Context API for handling the state.
- **User Authentication**: Optional, but bonus points if implemented using any token-based authentication method.

Step 2: Implementing Location Permission Request

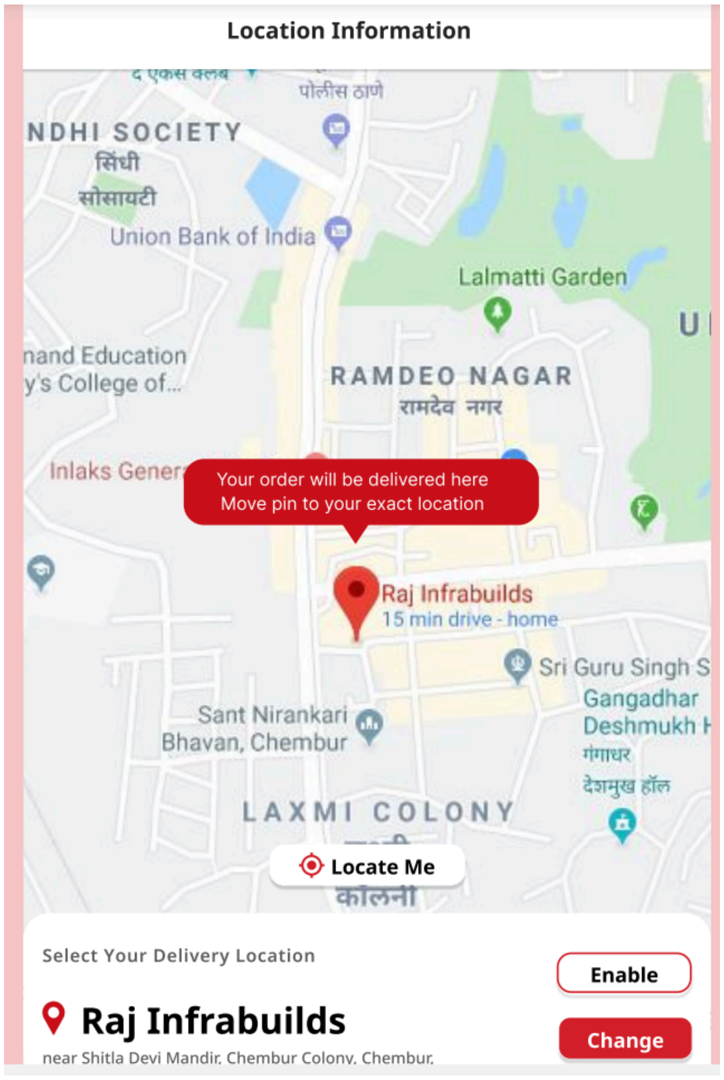


Location permission is off

We need your location to find the nearest store & provide you a seamless delivery experience

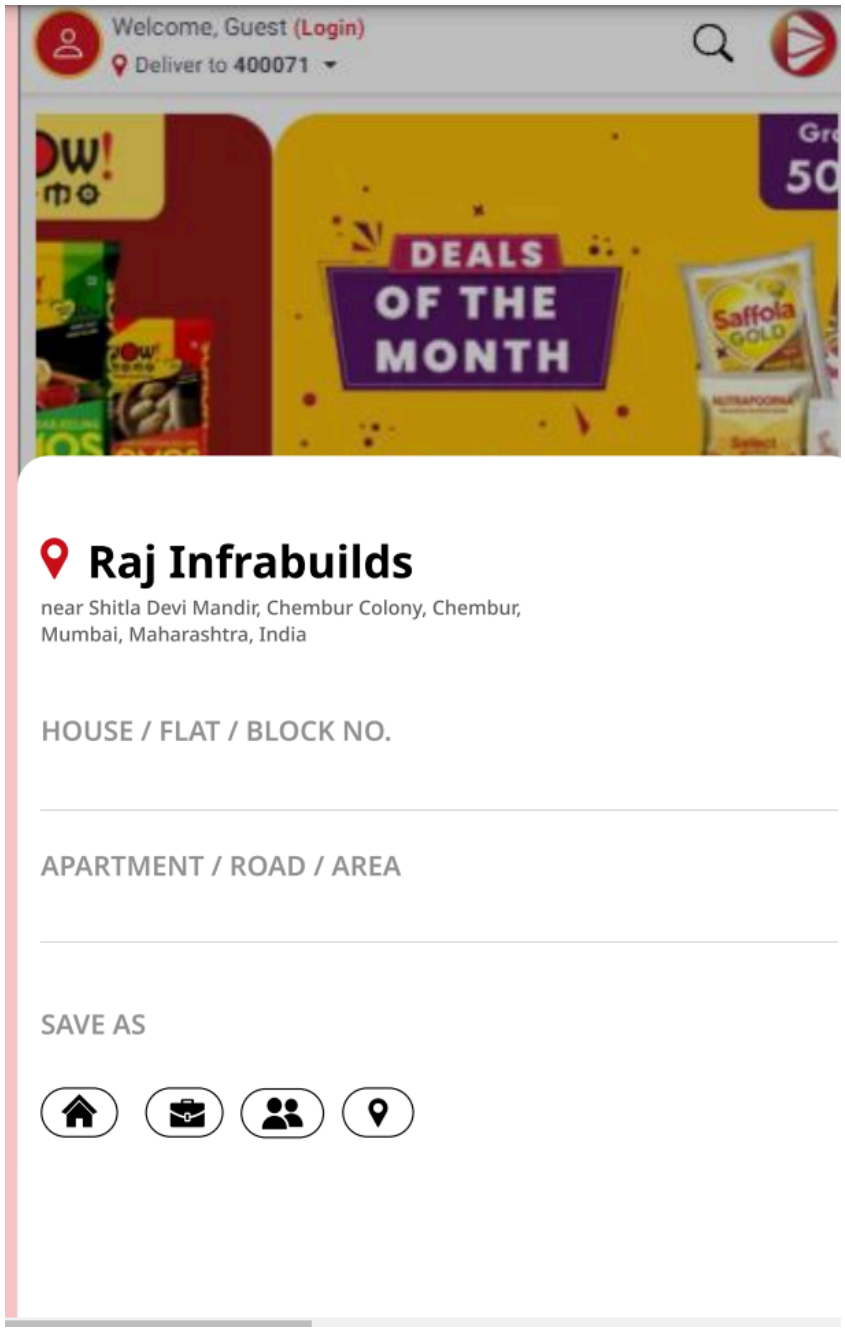
- Create a popup modal that informs the user when the location permission is turned off, as shown in Image 1.
- Add two buttons:
    - **Enable Location**: Clicking this should request the user's location through browser permissions and update the delivery address accordingly.
    - **Search Manually**: This should allow users to manually search for their address using a Google Map location input field.

Step 3: Geolocation & Pin Selection



- Implement the location selection interface (Image 2).
  - Once the user enables their location or manually searches for an address, display the selected address on a map.
  - Allow users to adjust the map pin to fine-tune their location.
  - Add a **Locate Me** button to help users find their current location automatically using the browser's geolocation feature.

Step 4: Delivery Address Form



near Shitla Devi Mandir, Chembur Colony, Chembur, Mumbai, Maharashtra, India

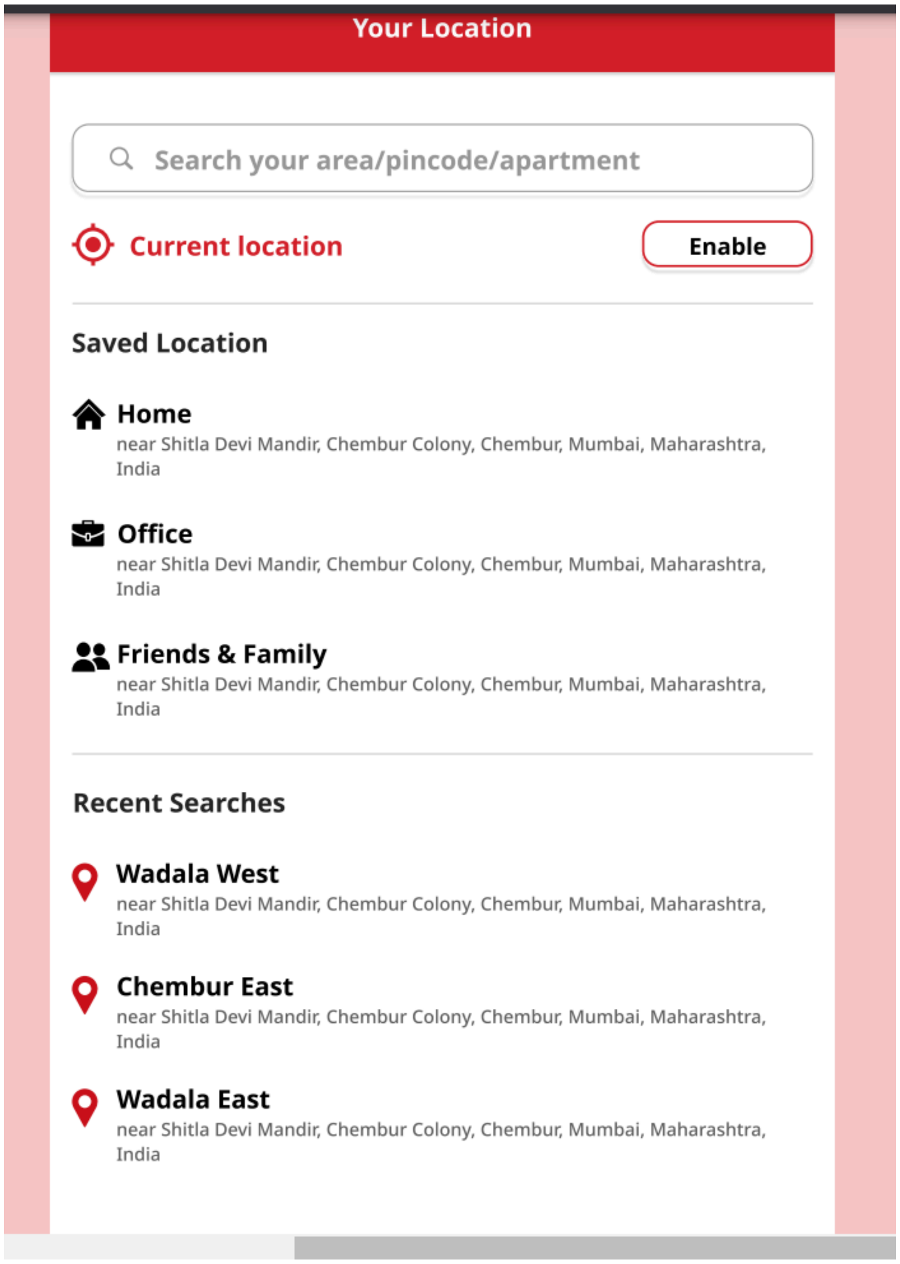
HOUSE / FLAT / BLOCK NO.

APARTMENT / ROAD / AREA

SAVE AS

- In this step, implement an address form (as shown in Image 3) where the user can enter specific details like:
  - House/Flat/Block No.
  - Apartment/Road/Area
- Allow the user to save the address under specific categories like Home, Office, or Friends & Family by selecting one of the icons.

Step 5: Address Management


Your Location

Search your area/pincode/apartment

Current location — Enable

Saved Location

Home
near Shitla Devi Mandir, Chembur Colony, Chembur, Mumbai, Maharashtra, India

Office
near Shitla Devi Mandir, Chembur Colony, Chembur, Mumbai, Maharashtra, India

Friends & Family
near Shitla Devi Mandir, Chembur Colony, Chembur, Mumbai, Maharashtra, India

Recent Searches

Wadala West
near Shitla Devi Mandir, Chembur Colony, Chembur, Mumbai, Maharashtra, India

Chembur East
near Shitla Devi Mandir, Chembur Colony, Chembur, Mumbai, Maharashtra, India

Wadala East
near Shitla Devi Mandir, Chembur Colony, Chembur, Mumbai, Maharashtra, India

- Implement a page where users can manage their saved addresses (Image 4).
  - Show a list of saved addresses (e.g., Home, Office, Friends & Family).
  - Allow users to select an address for delivery or update/delete existing addresses.
  - The user should also be able to search for recent addresses or search for new locations.

Bonus Features (Optional):

- Implement a **Save as Favorite** feature for frequently used locations.
- Add **Address validation** to ensure the accuracy of the entered location data.
- Add a **Map Preview** button for users to see a quick preview of their selected address on the map.

Expectations:

- **Functionality**: Your app should be fully functional, with all the steps mentioned above working as intended.
- **Responsiveness**: Ensure that your application is fully responsive and works well across different screen sizes.
- **Error Handling**: Implement proper error handling for scenarios like location permission denial, network issues, or invalid addresses.

Submission:

- Submit your code via a public GitHub repository.
- Ensure that the repository has a detailed **README** file explaining how to run the project and any additional features you have implemented.

Evaluation Criteria:

- Code quality: Clean, readable, and maintainable code.
- Functionality: Complete and correctly working features.
- UI/UX: User-friendly interface and design.
- Optional Bonus Features: Extra points for any implemented bonus features.