Prepared By : Athulya

Date: 03-03-2025

Scope Document

Project Overview:

The project involves developing a service locator map module as a separate domain (e.g., abc.com) for a platform, using the MERN stack (MySQL, Express.js, React, Node.js)**.**This feature will allow users to search for various services (e.g., hospitals, pharmacies, healthcare providers, and non-medical services) based on ZIP code and distance radius.

The system will integrate Google Maps API for location-based search and display, along with third-party APIs for fetching real-time data from a CMS site. Users will also have the ability to save resources for future reference.

Workflow:

1. The user visits the site (e.g., abcmap.com).
2. The user enters a ZIP code and selects a service category.
3. The user applies filter options (e.g., specialty, service type).
4. The system fetches data from the third-party CMS API based on the search criteria.
5. Search results displayed on an interactive Google Map.
6. Hovering over locations shows a pop-up with details.
7. The user can interact with the map, view service details, and save locations for future reference.

Reference CMS Site:

https://data.cms.gov/resources/medicare-durable-medical-equipment-devices-supplies-by-geography-and-service-data-dictionary-2022(Used as the third-party API for retrieving search results).

Additionally, we have an account to access the Google Maps API for map-related functionalities.

Technology Stack:

- **Frontend:** React (PWA-enabled)
- **Backend:** Node.js with Express

- **Database:** MySQL
- **APIs:**
    - CMS API for healthcare data
    - Google Maps API for location services

Core Features:

1. **Search Services by ZIP Code**
    - Users can enter a ZIP code to find nearby services.
    - Default search radius: 30 miles (adjustable).
2. **Service Categories:**
    - Service 1
    - Service 2
3. **Filtering Options:**
    - Specialty
    - Service type
4. **Map Integration:**
    - Search results displayed on an interactive Google Map.
    - Hovering over locations shows a pop-up with details.
5. **User Accounts & Saved Searches:**
    - Users can save preferred service locations by creating an account.
    - Prioritized display for partner-affiliated services.
6. **Performance & Scalability Considerations:**
    - Implement caching for frequent searches.
    - Optimize database queries for fast retrieval.
    - Load balancing and API rate limiting.

Candidate Screening Questions: Fill your answer below each question

1. Have you worked with **MERN stack** development?
2. Are you familiar with **Google Maps API** integration?
3. Have you previously implemented **third-party API integrations**?
4. Do you have experience with **location-based search and filtering**?
5. Have you worked with **Progressive Web Apps (PWAs)** before?
6. How would you optimize a system that retrieves large amounts of location-based data?
7. Please provide an estimate for this work based on your experience.