Flood Relief Navigator Documentation

Objective

To create an application that provides users with information about nearby emergency services and flood zones using geolocation. The application is further made live on the web and built into mobile applications.

Documented Algorithm with Flowchart

Steps in the Algorithm

1. **Initialize Project**:
    - Design the HTML, CSS, and JavaScript files.
    - Set up the map using `Leaflet.js` and configure basic interactivity.
2. **Geolocation Detection**:
    - Use the `navigator.geolocation` API to detect the user's location.
    - Handle geolocation errors by falling back to a default location.
3. **Fetch Nearby Services**:
    - Query the Overpass API to retrieve data for nearby emergency services.
    - Parse the API response and classify services into categories: medical, rescue, shelter, etc.
4. **Display Data on Map**:
    - Add markers for emergency services.
    - Create polygons to visualize flood zones and their risk levels.
    - Enable marker popups with details such as address, distance, and risk level.
5. **User Interaction**:
    - Add buttons for filtering services by type (e.g., hospitals, rescue, shelters).
    - Implement the "Locate Me" functionality to recenter the map on the user's location.
6. **Enhance Accessibility**:
    - Upload the project to GitHub for version control and sharing.

- Deploy the project to Netlify, making it publicly accessible via a live URL.
- Convert the live website into mobile applications (Android and iOS) using Median.

Flowchart

The flowchart below illustrates the process from initialization to the final application deployment:

```
START/Initialize
   |
   v
 Geolocation Detection
   - Detect user's location using geolocation API.
   - Handle errors by using a default location.
   |
   v
 Fetch Nearby Services
   - Query Overpass API for emergency services data.
   - Parse response and categorize services.
   |
   v
 Display Data on Map
   - Use Leaflet.js to add markers and polygons.
   - Include popups with service details and distances.
   |
   v
 Live Website using Netlify
   - Upload the project to GitHub.
   - Deploy the website to Netlify for public access.
   |
   v
 Final Application
   - Use Median to convert the website into Android and iOS apps.
   - Test and optimize the mobile applications.
   |
   v
END
```

Pseudo-code

```
START

FUNCTION main():
    LOAD Leaflet libraries
    INITIALIZE map and variables
    GET user location
    IF location detected:
```

```
        SHOW user location on map
        FETCH nearby services
        DISPLAY services on map and list
    ELSE:
        SHOW default location on map
    ENDIF
    LOAD flood zones
    SETUP UI interactions

FUNCTION getUserLocation():
    IF geolocation supported:
        OBTAIN latitude, longitude
        RETURN location
    ELSE:
        RETURN default location
    ENDIF

FUNCTION fetchNearbyServices(lat, lon, radius):
    BUILD Overpass API query for amenities
    SEND query to API
    RECEIVE response
    PARSE response into services list
    RETURN services list

FUNCTION createServiceMarker(service):
    CLASSIFY service type
    CREATE marker with icon color based on type
    BIND popup with service details
    RETURN marker object

FUNCTION addFloodZones():
    FOR each flood zone:
        CREATE polygon with risk-level color
        BIND popup with zone details
        ADD polygon to map
    ENDFOR

FUNCTION filterMarkers(filterType):
    FOR each marker:
        IF marker matches filterType:
            SHOW marker on map
        ELSE:
            HIDE marker
        ENDIF
    ENDFOR
    IF filterType == "floodzones":
        SHOW flood zones
    ELSE:
        HIDE flood zones
    ENDIF

FUNCTION populateServicesList():
    CLEAR current list
    FOR each service:
        ADD service details to list
        BIND event to center map on marker
    ENDFOR
```

```
END
```

Data Structures

Flood Zones (Static Data)

The Flood Zones are inherently dynamic in nature with time but due to the lack of live feed data, we have chosen the Flood zone to be static, this also inherently reduces the load on the app and in the server side the flood zone polygons can be changed when required reducing the load of analyzing flood zones in user end.

```
const floodZones = [
    {
        name: "Hiranandani Gardens Basin",
        risk: "LOW",
        coordinates: [[lat1, lon1], [lat2, lon2], ...]
    },
    ...
];
```

Service Markers (Dynamic Data)

The Service Markers are dynamic as they are rendered based on their proximity to the user.

```
let serviceMarkers = [
    {
        marker: L.marker(...),
        type: "medical", // "rescue", "shelter", etc.
        details: "Service details",
        source: "OpenGovernmentData"
    },
    ...
];
```

User Location

Detect user's location using geolocation API, if location is turned off, we use default location to continue with the process of loading the application

```
  navigator.geolocation.getCurrentPosition(
    (position) => {
      userLocation = { lat: position.coords.latitude, lng: position.
          coords.longitude };
      }
let userLocation = { lat: 19.1334, lng: 72.9133 }; // Default location
```

Filter State

```
let currentFilter = "all"; // Options: "all", "medical", "rescue", "
    shelter", "floodzones"
```

Application Deployment

1. **Upload to GitHub**:
   - Create a GitHub repository and push the project files for version control and collaboration.
2. **Host on Netlify**:
   - Connect the GitHub repository to Netlify.
   - Deploy the site, making it live and accessible through a public URL.
3. **Build Mobile Applications**:
   - Use Median to convert the live Netlify site into mobile apps for both Android and iOS.
   - Customize app settings (e.g., app name, icon) and generate the APK and IPA files.

Sample Query Passed to Overpass API

```
[out:json][timeout:25];
(
  node["amenity"="hospital"](around:5000,19.1334,72.9133);
  node["amenity"="fire_station"](around:5000,19.1334,72.9133);
  node["amenity"="police"](around:5000,19.1334,72.9133);
  node["amenity"="shelter"](around:5000,19.1334,72.9133);
);
out body;
```

Sample Data Received from API


```
{
  "version": 0.6,
  "generator": "Overpass API 0.7.62.4 2390de5a",
  "osm3s": {
    "timestamp_osm_base": "2024-11-23T14:28:52Z",
    "copyright": "The data included in this document is from www.
       openstreetmap.org. The data is made available under ODbL."
  },
  "elements": [
    {
      "type": "node",
      "id": 1855812192,
      "lat": 19.1246531,
      "lon": 72.9172844,
```

```
      "tags": {
        "amenity": "hospital",
        "emergency": "no",
        "name": "Powai Dental Clinic"
      }
    },
    {
      "type": "node",
      "id": 1855812194,
      "lat": 19.1240880,
      "lon": 72.9164511,
      "tags": {
        "addr:district": "Mumbai Suburban",
        "addr:full": "1-A, BSNL Colony, Powai, Mumbai",
        "addr:postcode": "400076",
        "addr:state": "Maharashtra",
        "amenity": "hospital",
        "name": "Powai Hospital",
        "source": "Local Knowledge"
      }
    }
    ...
  ]
}
```


Flood Relief Navigator: Website and Mobile App (Snapshots)

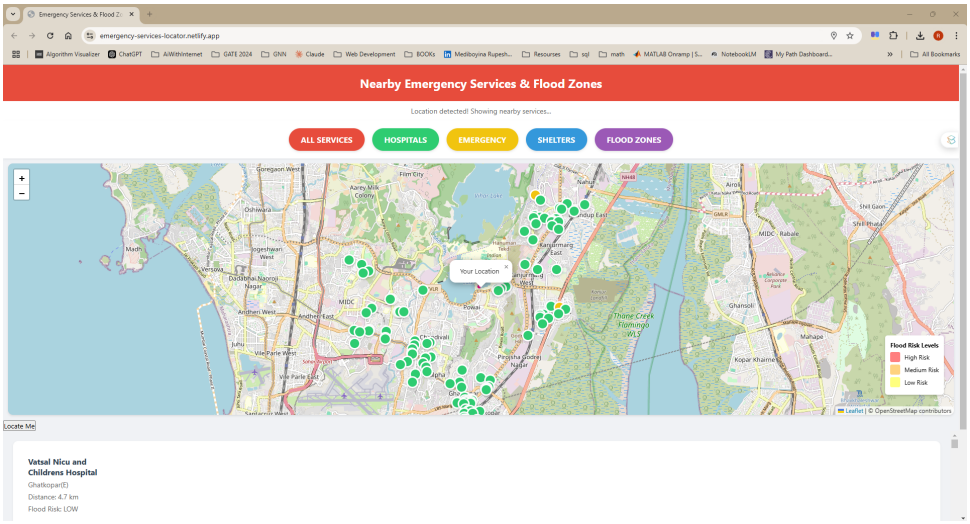

Figure 1: **Flood Relief Navigator Website:** The fully functional website hosted on Netlify. The website features an interactive map displaying nearby emergency services and flood zones.

The above image showcases the website hosted on Netlify. Visit it at https://emergency-services-locator.netlify.app/`https://emergency-services-locator.netlify.app/` to explore an interactive map displaying nearby emergency services and flood zones.

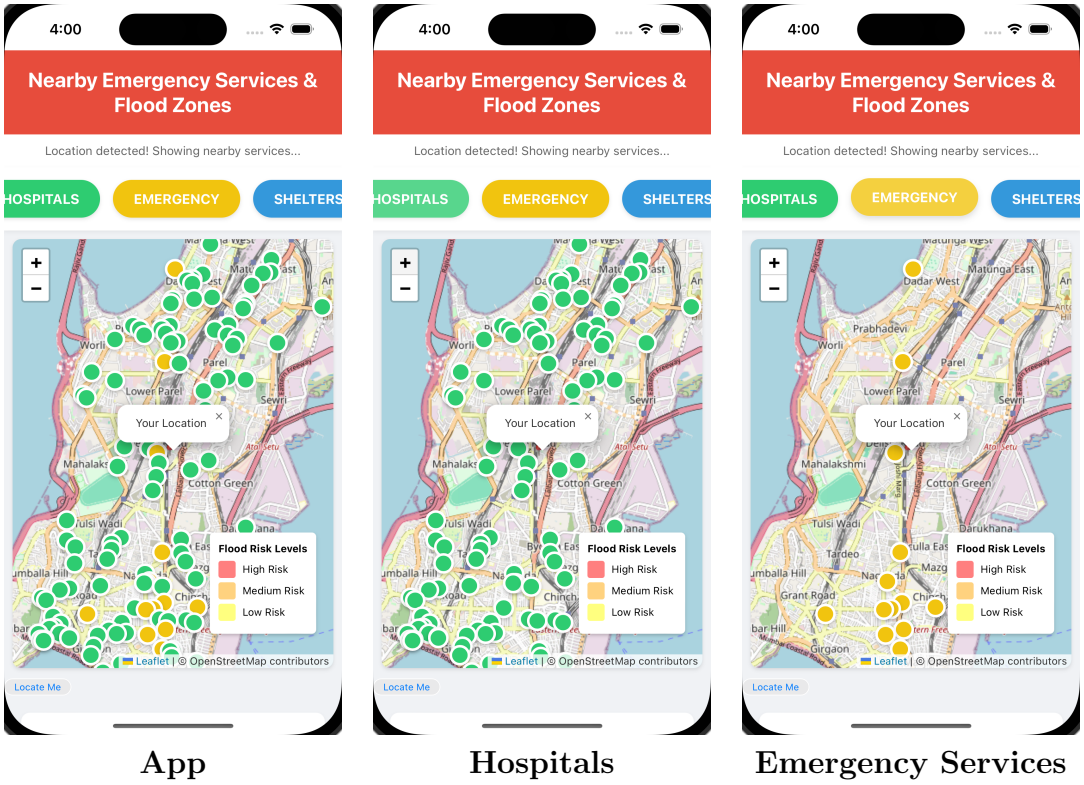

Figure 2: Features of the Flood Relief Navigator: All Services, hospitals, emergency services.

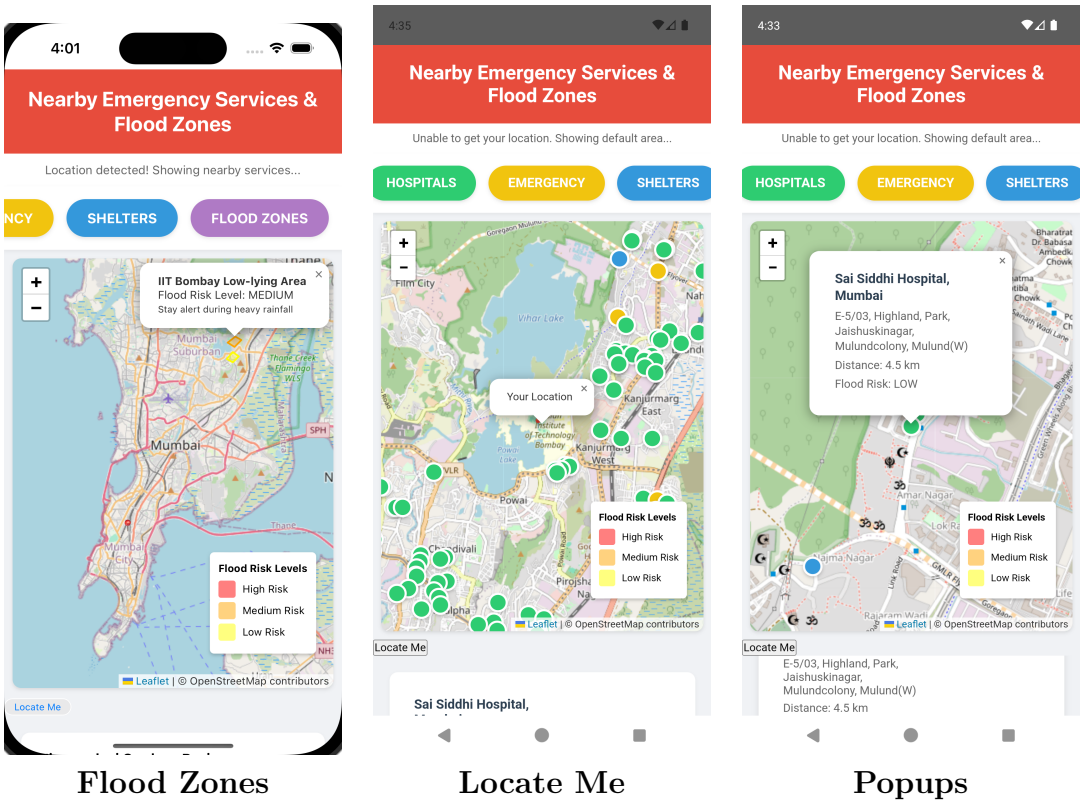

Figure 3: Features of the Flood Relief Navigator: flood zones, "Locate Me," and popups.

In the above Figure showcases the features on both present in the Android & IOS Applications.

Below is the QR code or the link https://median.co/share/erayepapk`https://median.co/share/eraye`
Use it to download the Flood Relief Navigator app for iOS or Android devices:

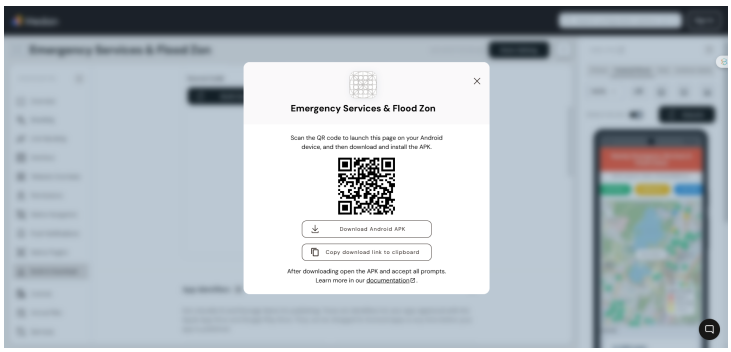

Figure 4: **Download the App:** Scan this QR code to download the Flood Relief Navigator app.