Project Title: Interactive Map Web Application

Name:- Rahul Kumar

Date:-27/07/2024

Institution/Organization :- Unified mentor , Student of Galgotias University

Table of Contents

Introduction

Project Overview: The Interactive Map Web Application is designed to provide a dynamic and engaging way for users to explore geographical information. Utilizing cutting-edge web technologies like Leaflet.js, ESRI Leaflet, and various mapping APIs, the application offers a range of features, including real-time geolocation, custom markers, search functionality, and multiple map layers. This project aims to create a user-friendly interface that allows for easy navigation and interaction with the map, making geographical data accessible and informative.

Objectives: The primary goals of the Interactive Map Web Application are as follows:

1. **User Interaction:** Enable users to interact with the map through marker placement, popups, and custom icons.
2. **Real-time Geolocation:** Provide real-time geolocation features to show the user's current location with high accuracy.
3. **Search Functionality:** Integrate a robust search feature to help users find specific locations on the map quickly and efficiently.
4. **Custom Layers:** Offer multiple base and overlay map layers, including OpenStreetMap, Google Maps, and custom GeoJSON data layers.
5. **Responsive Design:** Ensure the map is fully responsive and accessible on various devices, including desktops, tablets, and mobile phones.
6. **Educational and Informative:** Present geographical and cultural information through popups and markers, enhancing user knowledge about different locations.

Scope: The scope of this project includes:

- **Map Integration:** Integrating Leaflet.js and ESRI Leaflet libraries to create the interactive map.
- **Marker and Popup Functionality:** Adding and customizing markers, including popups with text, images, and videos.
- **Real-time Geolocation:** Implementing features to track and display the user's current location.
- **Layer Management:** Providing various map layers and controls to switch between them.
- **Search Feature:** Integrating a search bar to locate places on the map.
- **Responsive Design:** Ensuring the application works seamlessly across different screen sizes and devices.

However, the project has some limitations:

- **Data Dependency:** The accuracy and detail of the map depend on the data provided by external APIs and services.
- **Performance:** Performance may vary depending on the device and internet connection, particularly when handling large datasets or complex layers.

- **Browser Compatibility:** While efforts are made to ensure cross-browser compatibility, some advanced features may not work on older browser versions.

Background

Map Technologies:

Leaflet.js: Leaflet.js is an open-source JavaScript library for mobile-friendly interactive maps. It provides a lightweight, simple, and efficient means to create and manage map interfaces on the web. Leaflet's modularity allows for the inclusion of various plugins that can extend its functionality, making it highly customizable to suit different project needs.

Key Features:

- **Lightweight and Efficient:** Leaflet is designed to work efficiently across all major desktop and mobile platforms, focusing on simplicity, performance, and usability.
- **Customizability:** Supports various map layers, custom markers, popups, and interactive elements.
- **Plugins:** A wide array of plugins are available to add functionalities such as geocoding, drawing, and more.

ESRI Leaflet: ESRI Leaflet is a lightweight set of tools for working with ArcGIS services in Leaflet. It allows developers to integrate ESRI's robust geospatial capabilities with the simplicity of Leaflet, enabling the use of ArcGIS Online, ArcGIS Server, and ArcGIS Enterprise services within a Leaflet application.

Key Features:

- **ArcGIS Services Integration:** Seamlessly integrates with ESRI's powerful mapping and geospatial analysis services.
- **Geocoding:** Provides tools for address and place searching.
- **Feature Layers:** Supports dynamic map services, feature services, and image services.

Geolocation and Mapping APIs:

Navigator.geolocation API: The Navigator.geolocation API is a web standard API provided by modern browsers to access the geographical location of the device. It allows for real-time tracking of a user's location with methods to obtain the current position and watch for changes in position.

Key Features:

- **Real-time Location Tracking:** Provides real-time latitude and longitude coordinates of the user's device.
- **Accuracy:** Offers accuracy information, ensuring precise location tracking.
- **Cross-Browser Compatibility:** Supported by most modern web browsers.

OpenStreetMap (OSM): OpenStreetMap (OSM) is a collaborative project to create a free editable map of the world. It allows developers to use and contribute to its comprehensive and up-to-date geographical data.

Key Features:

- **Free and Open:** All data is free to use and edit, encouraging community participation.
- **Detailed Mapping:** Includes a wide range of geographical features and points of interest.
- **Tile Services:** Provides map tiles that can be easily integrated into web applications.

Google Maps API: Google Maps API is a set of web services and libraries provided by Google to integrate maps, geocoding, and geolocation services into web applications. It offers a rich set of features and a high degree of customization.

Key Features:

- **Rich Data:** Access to detailed and up-to-date geographical data and satellite imagery.
- **Customization:** Allows extensive customization of map styles, markers, and controls.
- **Geocoding and Geolocation:** Provides powerful tools for address lookups and location tracking.

Stamen Maps: Stamen Maps is a project by Stamen Design that provides custom map tiles, such as Watercolor, Toner, and Terrain maps. These tiles offer unique and artistic visual styles that can enhance the aesthetic appeal of a web map.

Key Features:

- **Artistic Styles:** Provides visually distinctive map styles.
- **Tile Services:** Easily integrates with Leaflet for custom map layers.

Design and Planning

Wireframes: Wireframes are basic, low-fidelity sketches that outline the structure and layout of the interactive map application. They focus on the placement of key elements such as the map container, controls, and informational popups without detailed design elements.

- **Home Page Wireframe:** Shows the basic layout of the map, including the header, map container, and footer. It outlines where the map and various controls (such as zoom buttons and layer switchers) will be placed.
- **Popup Wireframe:** Details the layout of popups that appear when a user clicks on a marker. This includes space for text, images, and possibly video content.

Mockups: Mockups are higher-fidelity visual representations of the final design, including colors, typography, and images. They provide a more realistic view of how the finished application will look.

- **Home Page Mockup:** Demonstrates the completed design of the map interface, including custom markers, base layer options, and the search bar.
- **Popup Mockup:** Displays the design of the popups with styled text, images, and multimedia elements.

Feature List:

1. **Interactive Map:**
    - **Map Display:** A responsive map that adjusts to the size of the browser window.
    - **Zoom Controls:** Buttons to zoom in and out.
    - **Layer Control:** Options to switch between different map layers (OSM, Watercolor, Dark, etc.).
2. **Markers:**
    - **Custom Markers:** Use custom icons to represent specific locations on the map.
    - **Popups:** Markers display popups with information, images, and videos when clicked.
    - **Draggable Markers:** Allow users to drag markers to new locations.
3. **Real-time Geolocation:**
    - **Location Tracking:** Display the user's current location on the map.
    - **Accuracy Circle:** Show a circle around the location indicating the accuracy of the geolocation.
4. **Search Functionality:**
    - **Search Bar:** A text input field for users to search for locations.
    - **Search Results:** Display search results as markers on the map.
5. **Map Layers:**

    - **Base Layers:** Options to switch between different base maps (e.g., OpenStreetMap, Google Streets, Satellite).
    - **Overlay Layers:** Additional data layers such as GeoJSON data, WMS layers, and custom layers.
6. **Information Display:**
    - **Coordinate Display:** Show latitude and longitude coordinates of the mouse position on the map.
    - **Popup Content:** Display relevant information about locations and points of interest.

User Interface Design:

1. **Layout:**
    - **Header:** Includes the title of the map and any additional controls or navigation links.
    - **Map Container:** The central area where the map is displayed, which should be responsive and fill the available space.
    - **Footer:** May include credits, attribution, or additional links.
2. **Design Considerations:**
    - **Responsiveness:** Ensure the map and all controls adjust gracefully to different screen sizes (desktop, tablet, mobile).
    - **Usability:** Design controls and interactions to be intuitive and easy to use. Ensure that map layers, search functions, and popups are accessible.
    - **Accessibility:** Consider accessibility features such as keyboard navigation and screen reader support to ensure the map is usable by all individuals.
    - **Visual Design:** Use a consistent color scheme, typography, and iconography to create a cohesive visual experience. Ensure that markers, popups, and controls are easily distinguishable.
3. **Aesthetic Elements:**
    - **Icons and Markers:** Choose icons that are easily recognizable and visually appealing.
    - **Colors and Themes:** Select colors that enhance readability and contrast well with the map's base layer.
    - **Popups and Information Windows:** Design these elements to be visually appealing and informative, incorporating images and multimedia effectively.

Map Layers and Data

Base Layers:

The base layers provide the foundational map tiles over which other data and overlays are displayed. Here is a description of the different map layers used in the project:

1. **OpenStreetMap (OSM):**
    - URL: `https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png`

- Description: OSM provides a detailed and free map created by contributors around the world. It serves as the default layer for the map.
- Attribution: `&copy; OpenStreetMap contributors`

2. **Watercolor Map:**
   - URL: `https://stamen-tiles-{s}.a.ssl.fastly.net/watercolor/{z}/{x}/{y}.{ext}`
   - Description: The watercolor map tiles by Stamen Design provide a unique, artistic view of the map, giving a watercolor painting effect.
   - Attribution: `Map tiles by Stamen Design, CC BY 3.0 &mdash; Map data &copy; OpenStreetMap contributors`
   - Properties: `subdomains: 'abcd', minZoom: 1, maxZoom: 16, ext: 'jpg'`
3. **Dark Map:**
   - URL: `https://{s}.basemaps.cartocdn.com/dark_all/{z}/{x}/{y}{r}.png`
   - Description: The dark map tiles by CARTO offer a dark-themed basemap suitable for highlighting overlays with lighter colors.
   - Attribution: `&copy; OpenStreetMap contributors &copy; CARTO`
   - Properties: `subdomains: 'abcd', maxZoom: 19`
4. **Google Streets:**
   - URL: `http://{s}.google.com/vt/lyrs=m&x={x}&y={y}&z={z}`
   - Description: This layer provides Google's standard street map view, familiar to many users.
   - Properties: `maxZoom: 20, subdomains: ['mt0', 'mt1', 'mt2', 'mt3']`
5. **Hybrid:**
   - URL: `http://{s}.google.com/vt/lyrs=s,h&x={x}&y={y}&z={z}`
   - Description: Google's hybrid layer combines satellite imagery with street names and major landmarks.
   - Properties: `maxZoom: 20, subdomains: ['mt0', 'mt1', 'mt2', 'mt3']`
6. **Satellite View:**
   - URL: `http://{s}.google.com/vt/lyrs=s&x={x}&y={y}&z={z}`
   - Description: This layer provides high-resolution satellite imagery from Google.
   - Properties: `maxZoom: 20, subdomains: ['mt0', 'mt1', 'mt2', 'mt3']`
7. **Terrain View:**
   - URL: `http://{s}.google.com/vt/lyrs=p&x={x}&y={y}&z={z}`
   - Description: Google's terrain view layer shows physical features such as terrain and elevation.
   - Properties: `maxZoom: 20, subdomains: ['mt0', 'mt1', 'mt2', 'mt3']`

User Experience

Usability Testing:

Usability testing was conducted to gather feedback and identify potential areas for improvement. The process included the following steps:

1. **Test Plan:**
    - Objectives: Assess the overall usability of the interactive map, identify any issues with responsiveness and accessibility, and gather user feedback.
    - Participants: A diverse group of users, including individuals with different levels of map usage experience and those with disabilities.
2. **Test Scenarios:**
    - Participants were asked to perform a series of tasks, such as locating specific markers, using the search functionality, switching between map layers, and navigating the map using keyboard shortcuts.
3. **Feedback Collection:**
    - Feedback was collected through direct observation, user interviews, and online surveys. Key findings included:
        - Positive feedback on the map's responsiveness and ease of use on mobile devices.
        - Suggestions for improving keyboard navigation and making the search bar more prominent.
        - Requests for additional features, such as the ability to save custom map views.
4. **Improvements Implemented:**
    - Based on the feedback, several improvements were made:
        - Enhanced keyboard navigation by adding more focusable elements.
        - Increased the size and visibility of the search bar.
        - Added a feature to save and load custom map views for future sessions.

Challenges and Solutions

Technical Issues:

During the development of the interactive map project, several technical challenges were encountered:

1. **Map Performance:**
    - **Issue:** As more markers and layers were added, the map's performance began to degrade, especially on mobile devices.
    - **Impact:** Slow loading times and laggy interactions affected the user experience.

2. **Cross-Browser Compatibility:**
    - **Issue:** The map and its features did not behave consistently across different browsers (e.g., Chrome, Firefox, Safari).
    - **Impact:** Users experienced various issues such as incorrect display of map layers and non-functional geolocation features.
3. **Geolocation Accuracy:**
    - **Issue:** The accuracy of the geolocation feature varied significantly depending on the device and network conditions.
    - **Impact:** Users received inaccurate location data, making the geolocation feature unreliable.
4. **Layer Control Complexity:**
    - **Issue:** Managing multiple layers and overlays became complex, leading to difficulties in adding new layers and ensuring they worked seamlessly together.
    - **Impact:** Potential for conflicts and bugs, which could confuse users.
5. **Search Functionality:**
    - **Issue:** Integrating the search functionality with the map's various layers and ensuring accurate results was challenging.
    - **Impact:** Users struggled to find specific locations or markers, reducing the effectiveness of the search feature.

Solutions Implemented:

To address these technical issues, the following solutions were implemented:

1. **Optimizing Map Performance:**
    - **Solution:** Utilized clustering techniques to group nearby markers, reducing the number of individual elements rendered on the map.
    - **Implementation:**

```
var markers = L.markerClusterGroup();
markers.addLayer(L.marker([28.7041, 77.1025])); // Example
marker
map.addLayer(markers);
```

    - **Result:** Improved map performance, especially on mobile devices, by reducing rendering load.
2. **Ensuring Cross-Browser Compatibility:**
    - **Solution:** Conducted extensive testing on multiple browsers and used polyfills to ensure compatibility.
    - **Implementation:**

```
<script
src="https://cdn.polyfill.io/v2/polyfill.min.js"></script>
```

    - **Result:** Consistent behavior of map features across different browsers, enhancing the overall user experience.

3. **Improving Geolocation Accuracy:**
    - **Solution:** Combined geolocation data from multiple sources (GPS, Wi-Fi, cell towers) to improve accuracy and implemented a fallback mechanism.
    - **Implementation:**

```
 navigator.geolocation.getCurrentPosition(
    successCallback,
    errorCallback,
    { enableHighAccuracy: true }
);
```

    - **Result:** Increased accuracy and reliability of the geolocation feature, providing more precise user locations.
4. **Simplifying Layer Control:**
    - **Solution:** Refactored the layer management code to use a modular approach, making it easier to add, remove, and manage layers.
    - **Implementation:**

```
var baseMaps = {
    "OSM": osm,
    "Water color map": watercolor,

};

var overlayMaps = {
    "Markers": markers,
    };

L.control.layers(baseMaps, overlayMaps, { collapsed: false
}).addTo(map);
```

    - **Result:** Simplified layer control, reducing complexity and potential for conflicts, making it easier to manage multiple layers.
5. **Enhancing Search Functionality:**
    - **Solution:** Improved the search algorithm to handle various queries and integrated it more tightly with the map layers.
    - **Implementation:**

```
var searchControl = new
L.esri.Controls.Geosearch().addTo(map);
searchControl.on('results', function(data){
    results.clearLayers();
    for (var i = data.results.length - 1; i >= 0; i--) {
        results.addLayer(L.marker(data.results[i].latlng));
    }
});
```

    - **Result:** More accurate and reliable search results, making it easier for users to find specific locations and markers on the map.

Future Enhancements

Planned Features:

As the interactive map project evolves, several potential improvements and new features have been identified to enhance functionality and user experience:

1. **Heatmaps:**
    - **Description:** Integrate heatmaps to visualize density data, such as population density, traffic patterns, or user activity.
    - **Implementation:** Use the Leaflet.heat plugin to add heatmaps based on specific data points.

```
var heat = L.heatLayer([
    [50.5, 30.5, 0.2], // lat, lng, intensity
    [50.6, 30.4, 0.5],
    [50.7, 30.3, 0.8]
], { radius: 25 }).addTo(map);
```

2. **Routing and Directions:**
    - **Description:** Add routing and directions capabilities to help users find the best path between two points on the map.
    - **Implementation:** Integrate the Leaflet Routing Machine plugin for turn-by-turn directions.

```
L.Routing.control({
    waypoints: [
        L.latLng(28.7041, 77.1025),
        L.latLng(27.173891, 78.042068)
    ]
}).addTo(map);
```

3. **Custom User Data:**
    - **Description:** Allow users to upload and display their own data on the map, such as CSV files with coordinates.
    - **Implementation:** Create a file upload interface and parse the uploaded data to add markers or shapes.

```
<input type="file" id="fileInput" accept=".csv" />
<script>

document.getElementById('fileInput').addEventListener('chan
ge', handleFileUpload);
    function handleFileUpload(event) {
        var file = event.target.files[0];

    }
</script>
```

4. **Enhanced Analytics:**
    - **Description:** Provide analytics tools to analyze user interactions with the map, such as heatmaps of clicks or time spent on specific areas.
    - **Implementation:** Use JavaScript to track user interactions and visualize the data.

```
map.on('click', function(e) {
});
```

5. **User Accounts and Personalization:**
    - **Description:** Implement user accounts to save personalized map settings and data layers.
    - **Implementation:** Use authentication libraries and a backend service to manage user accounts and preferences.

```
auth.signInWithEmailAndPassword(email,
password).then((user) => {
    // Save and load user preferences
});
```

Technological Upgrades:

To ensure the project remains current and leverages the latest advancements, several technological upgrades are planned:

1. **Library Updates:**
    - **Leaflet:** Regularly update the Leaflet library to benefit from performance improvements, new features, and bug fixes.
    - **ESRI Leaflet:** Upgrade to the latest version to ensure compatibility and access new functionalities.
2. **Performance Optimization:**
    - **Web Workers:** Utilize web workers for offloading heavy computations, such as data processing for large datasets.

```
var worker = new Worker('worker.js');
worker.postMessage({ data: largeDataSet });
```

3. **Improved Geolocation:**
    - **Advanced Positioning APIs:** Integrate advanced positioning APIs to enhance geolocation accuracy, especially in urban environments.

```
// Example of using a more advanced geolocation API
advancedGeolocation.getCurrentPosition(successCallback,
errorCallback);
```

4. **Augmented Reality Integration:**
    - **AR Features:** Explore the integration of augmented reality (AR) features to provide an immersive mapping experience.

```
// Example of initializing an AR session
navigator.xr.requestSession('immersive-ar').then((session)
=> {
    // Add AR features to the map
});
```

5. **Progressive Web App (PWA):**
    - **PWA Conversion:** Convert the project into a Progressive Web App to provide offline access and improved performance on mobile devices.

```
// Example of registering a service worker
if ('serviceWorker' in navigator) {
    navigator.serviceWorker.register('/service-
worker.js').then((registration) => {
        console.log('ServiceWorker registration successful
with scope: ', registration.scope);
    });
}
```

Conclusion

The interactive map project successfully achieved its primary goals and objectives, providing a robust and engaging platform for users to explore various geographic locations and features. Throughout the development process, the project incorporated multiple advanced map technologies, including Leaflet and ESRI Leaflet, to create a dynamic and user-friendly map interface. Key features such as customizable markers, real-time geolocation, search functionality, and various map layers were effectively implemented, offering a comprehensive and interactive user experience.

Significant milestones and learnings from this project include:

- **Map Initialization and Layer Management:** The project successfully demonstrated how to initialize a map with various base layers and overlays, ensuring seamless integration and control.
- **Geolocation and User Interaction:** Implementing real-time geolocation and user interaction features highlighted the importance of accuracy and performance in mapping applications.
- **Search Functionality and Data Integration:** The integration of search capabilities and custom data overlays showcased the potential for creating highly interactive and informative map interfaces.
- **Responsive Design:** Ensuring that the map adjusted effectively to different screen sizes and devices was a critical aspect of the project's success.

Impact:

The interactive map provides significant benefits to users by enhancing their ability to visualize and interact with geographic data. Key impacts include:

- **Enhanced Navigation and Exploration:** Users can easily navigate and explore various locations, accessing detailed information and visuals for each point of interest. The addition of markers with custom icons and popups enhances the exploration experience.
- **Real-Time Geolocation:** The real-time geolocation feature allows users to view their current location on the map, providing a valuable tool for navigation and situational awareness.
- **Custom Data Visualization:** The ability to display custom data layers and GeoJSON overlays enables users to visualize complex datasets in a spatial context, facilitating better analysis and decision-making.
- **Search and Discovery:** The integrated search functionality allows users to quickly find specific locations and features, improving the efficiency of map-based tasks.
- **Educational and Informative Tool:** The map serves as an educational resource, providing detailed information and visuals for various geographic locations and landmarks.

In conclusion, the interactive map project has successfully delivered a powerful tool for geographic visualization and interaction. The project's achievements and learnings provide a solid foundation for future enhancements and technological upgrades, ensuring continued improvement and relevance in the evolving field of web-based mapping applications.

References

Libraries and Tools:

1. **Leaflet:**
   - **Description:** A popular open-source JavaScript library for mobile-friendly interactive maps.
   - **Link:** Leaflet
2. **ESRI Leaflet:**
   - **Description:** A lightweight set of tools for working with ArcGIS services in Leaflet.
   - **Link:** ESRI Leaflet
3. **Stamen Tile Layers:**
   - **Description:** Provides a variety of beautiful and engaging tile layers for maps.
   - **Link:** Stamen Design
4. **CartoDB Basemaps:**
   - **Description:** Offers various basemap styles, including dark mode.
   - **Link:** CartoDB Basemaps
5. **Google Maps Layers:**
   - **Description:** Integrates Google Maps layers such as streets, satellite, hybrid, and terrain.

- **Link:** Google Maps

Documentation:

1. **Leaflet Documentation:**
   - **Description:** Comprehensive guide to using Leaflet for creating interactive maps.
   - **Link:** Leaflet Documentation
2. **ESRI Leaflet Documentation:**
   - **Description:** Official documentation for using ESRI services with Leaflet.
   - **Link:** ESRI Leaflet Documentation
3. **Stamen Tile Layers Documentation:**
   - **Description:** Information on using Stamen's beautiful tile layers.
   - **Link:** Stamen Tile Layers
4. **CartoDB Basemaps Documentation:**
   - **Description:** Details on using CartoDB's various basemap styles.
   - **Link:** CartoDB Basemaps
5. **Google Maps Documentation:**
   - **Description:** Guides and references for integrating Google Maps into web applications.
   - **Link:** Google Maps Documentation

Implementation

Code :-

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Interactive Map</title>

    <!--leaflets css-->
    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css"
integrity="sha256-p4NxAoJBhIIN+hmNHrzRCf9tD/miZyoHS5obTRR9BMY="
crossorigin="" />
    <link rel="stylesheet" type="text/css" href="https://cdn-
geoweb.s3.amazonaws.com/esri-leaflet-geocoder/0.0.1-beta.5/esri-leaflet-geocoder.css">
    <link rel="stylesheet" href="styles.css">

</head>
<body>
    <h1><b>Our World Map</b></h1>
    <div id="map">
```

```
        <div class="leaflet-control coordinate"></div>
    </div>
</body>
</html>
<!--leaflet.js-->
<script src="https://unpkg.com/leaflet@1.9.4/dist/leaflet.js" integrity="sha256-
20nQCchB9co0qIjJZRGuk2/Z9VM+kNiyxNV1lvTlZBo=" crossorigin=""></script>
<script src="https://cdn-geoweb.s3.amazonaws.com/esri-leaflet/0.0.1-beta.5/esri-
leaflet.js"></script>
<script src="https://cdn-geoweb.s3.amazonaws.com/esri-leaflet-geocoder/0.0.1-
beta.5/esri-leaflet-geocoder.js"></script>
<script src="./point.js"></script>
<script src="./line.js"></script>
<script src="./locator.js"></script>
<script src="./polygon.js"></script>


<script>

// Detect the screen size
var isMobile = window.innerWidth <= 768;

//Map initialization
var map = L.map('map').setView([28.7041, 77.1025], isMobile ? 8 :  10);

    //osm layer
var osm = L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
    attribution: '&copy; <a
href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'
});
map.addLayer(osm)

// water color
var watercolor = L.tileLayer('https://stamen-tiles-
{s}.a.ssl.fastly.net/watercolor/{z}/{x}/{y}.{ext}', {
    attribution: 'Map tiles by <a href="http://stamen.com">Stamen Design</a>, <a
href="http://creativecommons.org/licenses/by/3.0">CC BY 3.0</a> &mdash; Map data
&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a>
contributors',
    subdomains: 'abcd',
    minZoom: 1,
    maxZoom: 16,
    ext: 'jpg'
});
// watercolor.addTo(map)
```

```
// dark map
var dark = L.tileLayer('https://{s}.basemaps.cartocdn.com/dark_all/{z}/{x}/{y}{r}.png',
{
    attribution: '&copy; <a
href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors
&copy; <a href="https://carto.com/attributions">CARTO</a>',
    subdomains: 'abcd',
    maxZoom: 19
});
// dark.addTo(map)

//Google Streets layer
googleStreets = L.tileLayer('http://{s}.google.com/vt/lyrs=m&x={x}&y={y}&z={z}',{
    maxZoom: 20,
    subdomains:['mt0','mt1','mt2','mt3']
});
//googleStreets.addTo(map);

if(!navigator.geolocation) {
        console.log("Your browser doesn't support geolocation feature!")
    } else {
        setInterval(() => {
            navigator.geolocation.getCurrentPosition(getPosition)
        }, 5000);
    }

    var marker, circle;

    function getPosition(position){
        // console.log(position)
        var lat = position.coords.latitude
        var long = position.coords.longitude
        var accuracy = position.coords.accuracy

        if(marker) {
            map.removeLayer(marker)
        }

        if(circle) {
            map.removeLayer(circle)
        }

        marker = L.marker([lat, long])
        circle = L.circle([lat, long], {radius: accuracy})
```

```
        var featureGroup = L.featureGroup([marker, circle]).addTo(map)

        map.fitBounds(featureGroup.getBounds())

        console.log("Your coordinate is: Lat: "+ lat +" Long: "+ long+ " Accuracy: "+
accuracy)
    }

//Hybrid

Hybrid = L.tileLayer('http://{s}.google.com/vt/lyrs=s,h&x={x}&y={y}&z={z}',{
    maxZoom: 20,
    subdomains:['mt0','mt1','mt2','mt3']
});
//Hybrid.addTo(map);

//satellite

googleSat = L.tileLayer('http://{s}.google.com/vt/lyrs=s&x={x}&y={y}&z={z}',{
    maxZoom: 20,
    subdomains:['mt0','mt1','mt2','mt3']
});
//googleSat.addTo(map);

//Terrain

googleTerrain = L.tileLayer('http://{s}.google.com/vt/lyrs=p&x={x}&y={y}&z={z}',{
    maxZoom: 20,
    subdomains:['mt0','mt1','mt2','mt3']
});
//googleTerrain.addTo(map);

//ADD WMS
var wms = L.tileLayer.wms("http://localhost:8080/geoserver/wms", {
        layers: 'geoapp:admin',
        format: 'image/png',
        transparent: true,
        attribution: "wms test"
    });

//icon
var myIcon = L.icon({
    iconUrl: 'img/red_marker.png',
    iconSize: isMobile ? [20, 20] : [40, 40],
```

```
});
//marker
var singleMarker = L.marker([28.7041, 77.1025], { icon: myIcon });
var popup = singleMarker.bindPopup(
    '<h1>Delhi</h1>' +
    '<p><b>The location of Delhi is 28.7 degrees N and 77.1 degrees E. Delhi sits on the latitude of 28 degrees lying in the Northern Hemisphere - hence 28.7 degrees N. The longitude of Delhi is at 77 degrees East which means that the city is 77 degrees to the east of the Prime Meridian.</b></p>' +
    '<img src="img/indiagate.jpg" alt="India Gate" class="popup-image" />' +
    singleMarker.getLatLng()
).openPopup();


var secondMarker = L.marker([27.173891, 78.042068], { icon: myIcon });
var popup = secondMarker.bindPopup(
    '<h1>Agra</h1>' +
    '<p><b>One of the world\'s most known and famous tourist attractions, the Taj Mahal (ताज महल) is a historical palace built by the Mughal emperor Shah Jahan for his third wife Mumtaz Mahal in 1632.</b></p>' +
    '<img src="img/Taj_Mahal.png" alt="Taj Mahal" class="popup-image" />' +
    secondMarker.getLatLng()
 ).openPopup();

 var thirdMarker = L.marker([26.924089, 75.825833], { icon: myIcon });
var popup = thirdMarker.bindPopup(
    '<h1>Rajasthan</h1>' +
    '<p><b>The Hawa Mahal is a palace in the city of Jaipur, Rajasthan, India. Built from red and pink sandstone, it is on the edge of the City Palace, Jaipur, and extends to the Zenana, or women\'s chambers. </b></p>' +
    '<img src="img/hawa mahal.jpeg" alt="Hawa Mahal" class="popup-image" />' +
    thirdMarker.getLatLng()
 ).openPopup();


 var forthMarker = L.marker([25.122265, 85.456177], { icon: myIcon });
var popup = forthMarker.bindPopup(
    '<h1>Bihar</h1>' +
    '<p><b>Nālandā University is a central research university located in the ancient city of Rajgir in the state of Bihar, India. </b></p>' +
    '<img src="img/nalanda.jpeg" alt="Nalanda Mahavihar" class="popup-image" />' +
    forthMarker.getLatLng()
 ).openPopup();
```

```
 var fifthmarker = L.marker([26.347883, 86.071861], { icon: myIcon, draggable: true });
       var popupContent =
          '<h1>Bihar</h1>' +
          '<p><b>Madhubani railway station is a railway station in Madhubani district,
Bihar. Its code is MBI. It serves Madhubani city. The station consists of 3 platforms.
Near Hanuman Prem Mandir</b></p>' +
          '<img src="img/madhubani.jpeg" alt="Madhubani Station" class="popup-image"
/>'+
          '<video controls class="popup-video">' +
          '<source src="img/video.mp4" type="video/mp4">' +
          'Your browser does not support the video tag.' +
          '</video>';
       fifthmarker.bindPopup(popupContent).openPopup();

// Control 2: This add a scale to the map
L.control.scale().addTo(map);

// Control 3: This add a Search bar
    var searchControl = new L.esri.Controls.Geosearch().addTo(map);

    var results = new L.LayerGroup().addTo(map);

      searchControl.on('results', function(data){
        results.clearLayers();
        for (var i = data.results.length - 1; i >= 0; i--) {
          results.addLayer(L.marker(data.results[i].latlng));
        }
      });

console.log(singleMarker.toGeoJSON())

//geojson
var pointData = L.geoJSON(pointJson)
var locatorData = L.geoJSON(locatorJson)
    var lineData = L.geoJSON(lineJson)
    var polygonData = L.geoJSON(polygonJson, {
        onEachFeature: function (feature, layer) {
            layer.bindPopup(`<b>Name: </b>` + feature.properties.name)
        },
        style: {
            fillColor: 'blue',
            fillOpacity: 1,
            color: '#c0c0c0',
        }
    });
```

```
//layer controller

var baseMaps = {
    "OSM": osm,
    "Water color map": watercolor,
    'Dark': dark,
    "Google Streets": googleStreets,
    "Hybrid": Hybrid,
    "Satellite View": googleSat,
    "Terain View": googleTerrain

};

var overlayMaps = {
    "First Marker": singleMarker,
    "Second Marker": secondMarker,
    "Third Marker": thirdMarker,
    "Fourth Marker": forthMarker,
    "Fifth Marker": fifthmarker,
    "Point Data": pointData,
    "Locator Data": locatorData,
    "Line Data": lineData,
    "Polygon Data": polygonData,
    "WMS": wms
};
  // map.removeLayer(singleMarker)

  L.control.layers(baseMaps, overlayMaps, { collapsed: false }).addTo(map);

  //EVENT
  map.on('mouseover', function () {
        console.log('your mouse is over the map')
    })

    map.on('mousemove', function (e) {
        document.getElementsByClassName('coordinate')[0].innerHTML = 'lat: ' +
e.latlng.lat + 'lng: ' + e.latlng.lng;
        console.log('lat: ' + e.latlng.lat, 'lng: ' + e.latlng.lng)
    })


</script>
```

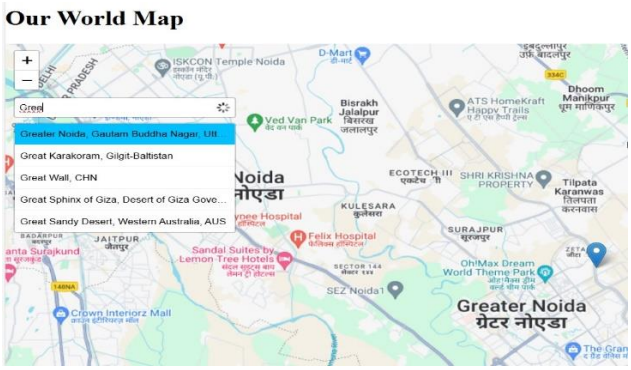
CSS Styling:-

```
html, body
        {
            height: 100%;
            margin: 0;
            padding: 0;
        }
        #map {
            width: 100%;
            height: 90vh;
        }


@media (max-width: 768px) {
    #map {
        height: 80vh; /* Adjust the height for tab */
    }
}
@media (max-width: 480px) {
    #map {
        height: 60vh; /* Adjust the height for smaller phones */
    }
}

        .coordinate {
            position: absolute;
            bottom: 10px;
            right: 50%;
        }

        .leaflet-popup-content-wrapper {
            background-color: yellow;
            color: black;
            border: 1px solid brown;
            border-radius: 0px;
        }
        .popup-image {
        width: 200px;
        height: auto;
     }
```

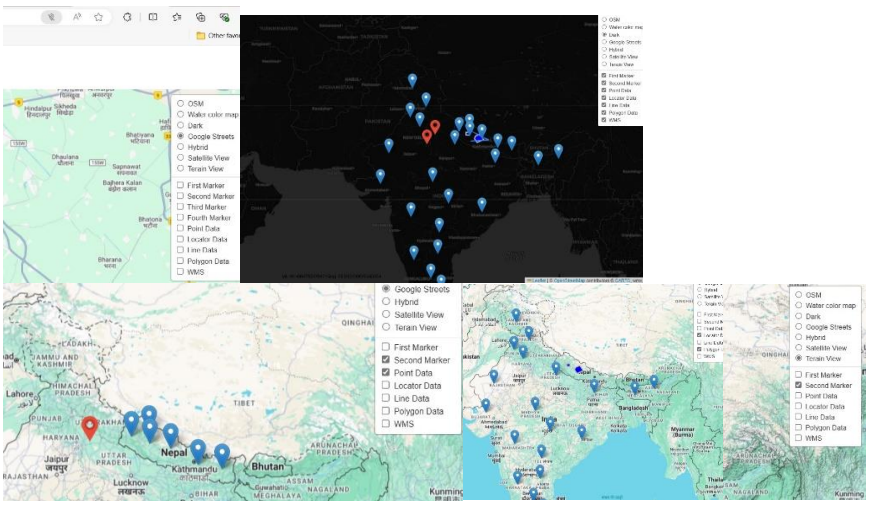
```
    .popup-video {
    width: 100%;
    height: auto;
}
```

Additional Images:

1. **Wireframes and Mockups:**
    - Include images or screenshots of initial wireframes and mockups used during the design phase.



2. **Map Screenshots:**
    - Include screenshots of the interactive map showcasing different features such as markers, layers, and popups.



3. **Responsive Design:**
    - Screenshots of the map on different devices (desktop, tablet, mobile) to illustrate responsiveness.