

INSTRUCTIONS:

Goal of the Project:

In Class 178, we have learned how we can add interactive maps using Mapbox JS GL library in our web applications.

We have also learned to add controls using GeolocateControl, to get the current location of the user and MapboxDirections control, to get the directions to travel from the source to till the destination.

In this project, you will practice the concepts learned in the class.

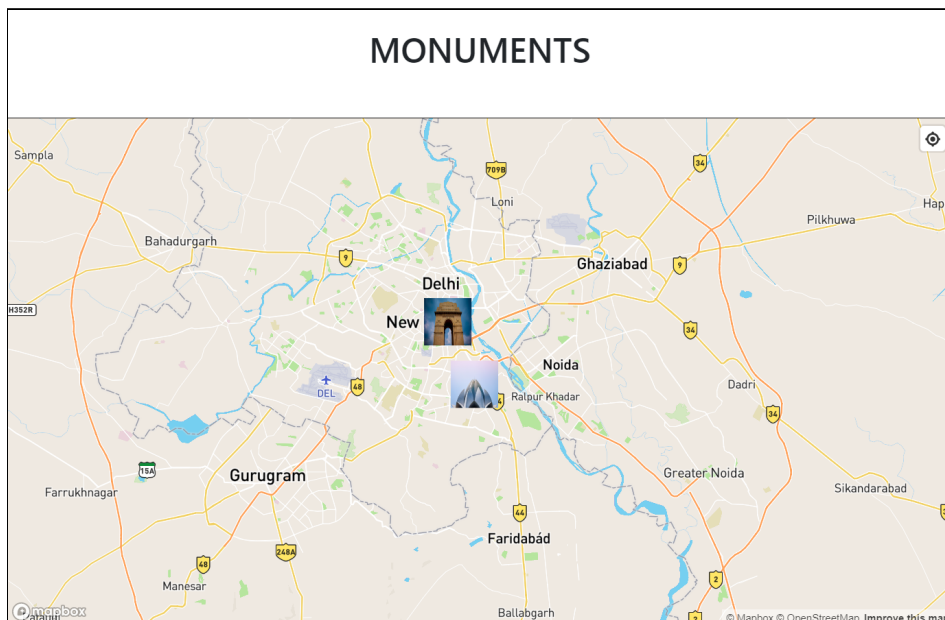
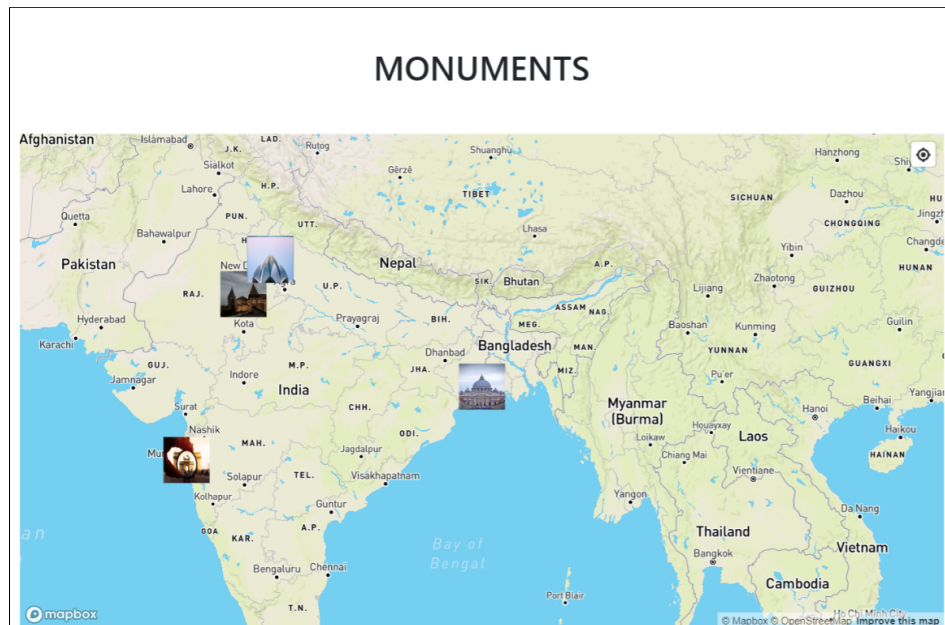
Story:

Tourism is important for the country's growth. It helps to connect people from all over the world who travel to places and get to know the world better.

People who like to travel would want to know what places they can visit when they travel to a particular state/country. The Governor of the state hired you, a software engineering expert and gave you the responsibility to provide this information for all the tourists. He has asked you to provide this information online of all monuments (or other places) that tourists can visit in your state.

We have already learned how we can add maps in any web application using Mapbox JS GL library. We can do so much more with maps!

We can add any HTML elements on the maps using [Markers](#). In this project, you have to add images of all the monuments on the map that a tourist can visit in your state.



***This is just for your reference. We expect you to apply your own creativity to the project.**

Getting Started:

1. Open it in VS Code Editor.
2. Include the following in <head>:

- **jQuery**

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"
integrity="sha256-/xUj+3OJU5yExlq6GSYGSgHk7tPXikynS7ogEvDej/m4="
crossorigin="anonymous"></script>
```

- **Bootstrap**

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0ho
P+VmmDGMN5t9UJ0Z" crossorigin="anonymous">
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.bundle.mi
n.js"
integrity="sha384-LtrjvnR4Twt/qOuYxE721u19sVFLVSA4hf/rRt6PrZTmiPltdZcl
7q7PXQBYTKyf"
crossorigin="anonymous"></script>
```

- **Mapbox API (.css and .js)**

```
<link href="https://api.mapbox.com/mapbox-gl-js/v2.1.1/mapbox-gl.css"
rel="stylesheet">
<script
src="https://api.mapbox.com/mapbox-gl-js/v2.1.1/mapbox-gl.js"></script>
```

- **Mapbox Geocoder CDN**

```
<scriptsrc="https://api.mapbox.com/mapbox-gl-js/plugins/mapbox-gl-geocoder/
v4.7.0/mapbox-gl-geocoder.min.js"></script>
<link rel="stylesheet"
href="https://api.mapbox.com/mapbox-gl-js/plugins/mapbox-gl-geocoder/v4.7.0
/mapbox-gl-geocoder.css" type="text/css">
```

3. Include the following in <body> **below the map** render container <div>:

- **Local CSS File**

```
<script src="./main.css"></script>
```

- **Local JavaScript File**

```
<script src="./main.js"></script>
```

Specific Tasks to complete the Project:

1. Add the HTML elements (with ids) for the following elements (in **main.html**):
 - Heading
 - Map
 - Monuments images
2. In **main.js** file:
 - Take the **mapboxgl** global variable.
 - Assign [Mapbox User Account](#) access token to it.
 - Create a new Map() object and set the following options:
 - container
 - style
 - center
 - zoom
 - Add a **MapboxGeocoder** control: To **search places in the map**(Refer to the Mapbox Geocoder's Documentation [here](#).)
 - To add the image [Marker](#) on the map:
 - Select the image element:

```
var img1= document.querySelector("#id")
```

- Use Marker() object to set the image element at a particular longitude and latitude and add it to the maps(Refer Hints):

```
var marker1=new mapboxgl.Marker({  
    element: img1  
})  
.setLngLat([xx.xxxxx , xx.xxxxx])  
.addTo(mapVariable)
```

Important Note 1: The order is **[longitude, latitude]** in the **.setLngLat()** method. The first value is longitude.

[How to find location coordinates using Google maps?](#)

Important Note 2: The order of the location coordinates in Google Maps is **[latitude, longitude]**. It is different from standard map applications, which have reverse order as **[longitude, latitude]**.

Submitting the Project:

1. Upload your completed project to your own GitHub account.
2. Create a new repository named **Project 178**.
3. **Upload** your project code to this GitHub repository.
4. Submit the published link of the project in the Student Dashboard.

Hints:

1. HTML elements (in **main.html**):

```
<!-- Body HTML -->
<div class="container">
  <div class="row">
    <div class="col-sm-12 col-md-12 col-lg-12 text-center p-5">
      <h1>MONUMENTS</h1>
    </div>
    <div class="col-sm-12 col-md-12 col-lg-12 text-center">
      <div id="map"></div>
    </div>
  </div>
  <div>
    
    
    
    
    
  </div>
</div>

<!-- CSS and JS -->
<link rel="stylesheet" href="./main.css" />
<script src="./main.js" type="text/javascript"></script>
```

2. Style HTML elements(in **main.css**).

```
body {  
    margin: 0;  
    padding: 0;  
    overflow-y: scroll;  
}  
  
#map {  
    height: 75vh;  
}  
  
.container {  
    margin-top: 1em;  
    border: 3px solid rgb(73, 88, 221);  
}  
  
.image-marker {  
    height: 55px;  
    width: 55px;  
}
```

3. Add Map(in **main.js**).

```
let latitude = 22.7868542, longitude = 88.3643296;  
  
// Initializing Mapbox  
mapboxgl.accessToken = '  
  
var map = new mapboxgl.Map({  
    container: 'map',  
    style: 'mapbox://styles/mapbox/streets-v11',  
    center: [longitude, latitude],  
    zoom: 4  
});
```

4. Add Marker(in **main.js**):

```
var img1 = document.querySelector("#amber")

// Create a Amber Palace, Jaipur Marker and add it to the map.
var marker1 = new mapboxgl.Marker({
    element:img1
})
.setLngLat([75.85133, 26.98547])
.addTo(map);
```

5. Add Mapbox Geocoder(in **main.js**)

```
map.addControl(
    new MapboxGeocoder({
        accessToken: mapboxgl.accessToken,
        mapboxgl: mapboxgl
    })
);
```

6. Refer to the **Mapbox Geocoder's Documentation** [here](#).

REMEMBER...

Try your best, that's more important than being correct.

After submitting your project your teacher will send you feedback on your work.

xxx

xxx

xxx

xxx

xxx