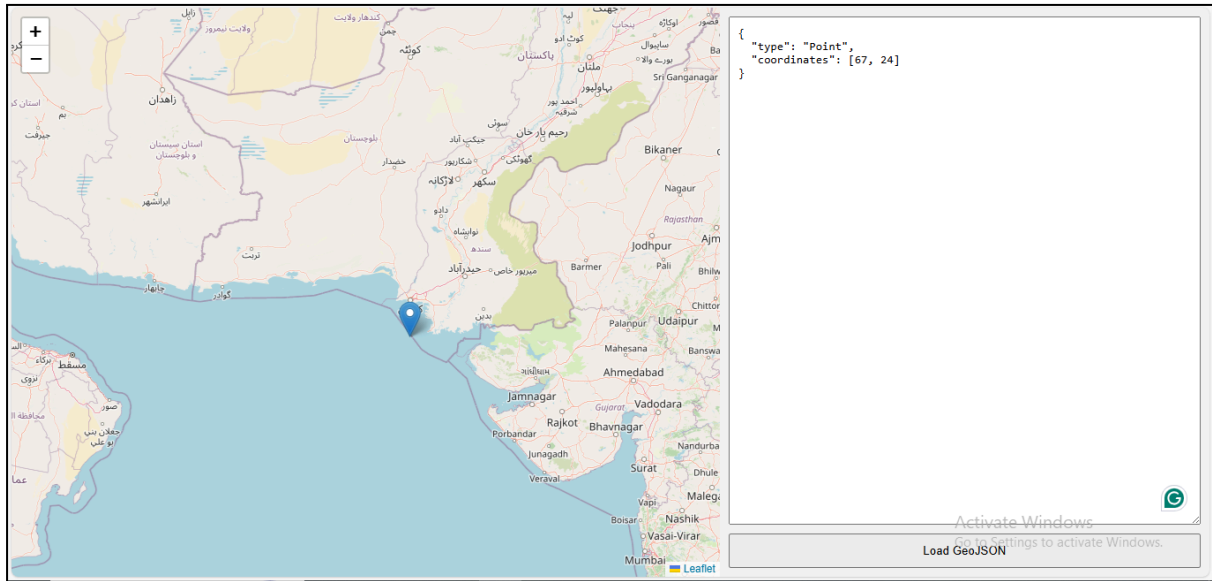


Testing Tasks

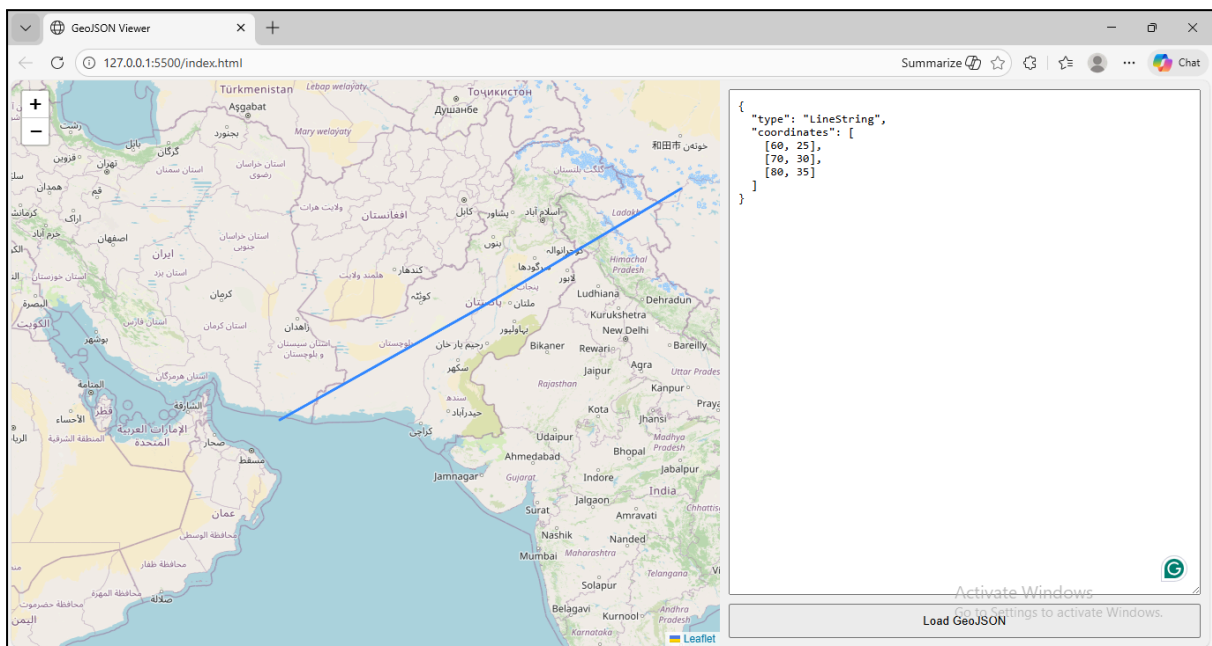
Test 1 – A Point

`{"type": "Point", "coordinates": [67, 24]}`



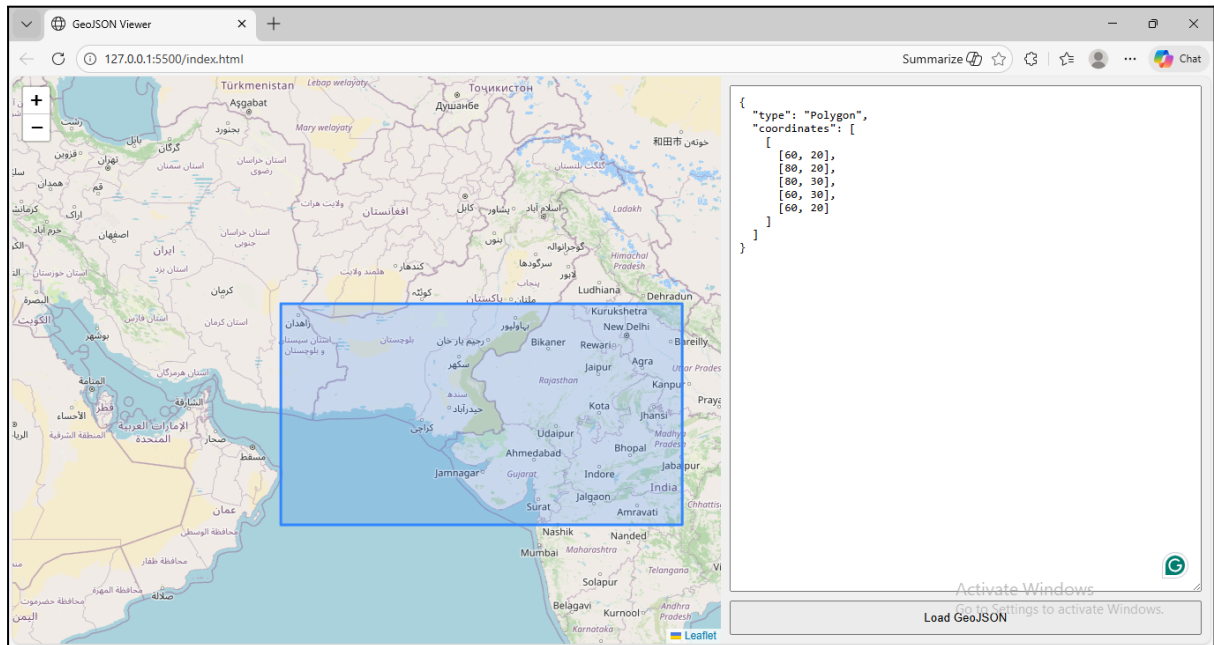
Test 2 – A LineString

`{"type": "LineString", "coordinates": [[60, 25], [70, 30], [80, 35]]}`



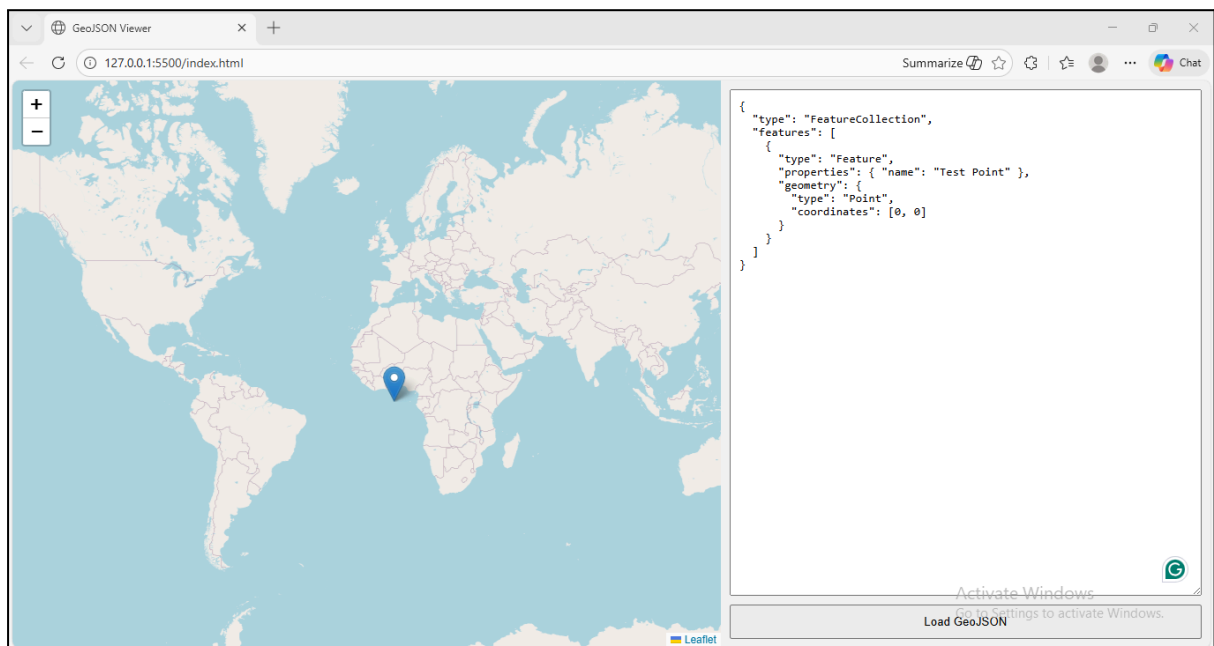
Test 3 – A Polygon

{“type”: “Polygon”, “coordinates”: [[60, 20], [80, 20], [80, 30], [60, 30], [60, 20]]}



Test 4 – A FeatureCollection

{“type”: “FeatureCollection”, “features”: [{“type”: “Feature”, “properties”: {“name”: “Test Point”}, “geometry”: {“type”: “Point”, “coordinates”: [0, 0]} }]}



Questions:

1. What is the purpose of the layer group in this lab?

Ans: It prevents old data from staying on the map when new data is added.

2. Why do we need to use JSON.parse() before passing data into L.geoJSON()?

Ans: Because the text typed in the textarea is a string, not real JSON. The leaflet requires a JavaScript object, not a text string. JSON.parse() converts the string into a JavaScript object, allowing L.geoJSON(parsedData) to correctly read the GeoJSON structure.

3. What error appears if your GeoJSON is invalid?

Ans: "Invalid GeoJSON! Please check your syntax."

4. How would you modify the viewer to let users load a .geojson file?

Ans: You can add a file input element:

```
<input type="file" id="fileInput" accept=".json,.geojson">
```

Then read the file using JavaScript FileReader:

```
var reader = new FileReader();
reader.onload = function(e) {
  let data = JSON.parse(e.target.result);
  layers.clearLayers();
  L.geoJSON(data).addTo(layers);
};
reader.readAsText(file);
```

This allows users to upload a .geojson file directly instead of pasting text.

5. How could you style GeoJSON features with custom colours or markers?

Ans:

```
L.geoJSON(data, {
  style: {
    colour: "red",
```

```
        weight: 3
    }
}).addTo(layers);

L.geoJSON(data, {
    pointToLayer: function(feature, latlng) {
        return L.circleMarker(latlng, {
            radius: 6,
            color: "blue",
            fillColor: "cyan",
            fillOpacity: 0.7
        });
    }
}).addTo(layers);
```