

PPKS – 4. laboratorijska vježba

Napravljena web stranica omogućuje vizualni prikaz putanje satelita na interaktivnoj Leaflet mapi, s mogućnošću filtriranja podataka po minuti i sekundi putem preglednog sučelja. Tehnologije koje su korištene: HTML, CSS, JavaScript, Leaflet.js za interaktivnu kartu i Java HTTP server za posluživanje datoteka.

Program se pokreće komandama:

```
javac Server.java
```

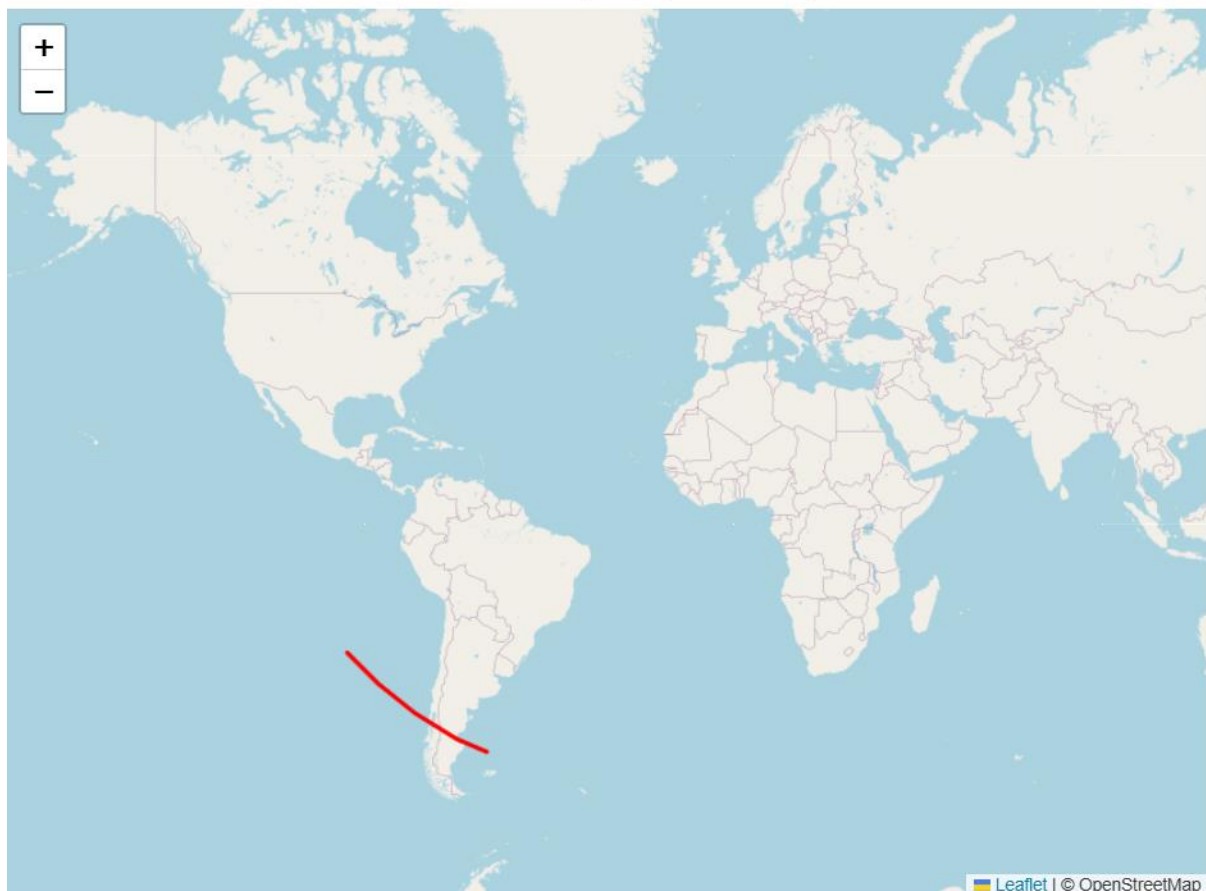
```
java Server
```

klik na <http://localhost:8000>

Putanja Satelita

Odaberi minutu putanje, datum: 2022-06-01 Cijela putanja ▼

Odaberi sekundu putanje: ----- ▼



Ključne funkcionalnosti

Grupiranje i prikaz podataka po minuti. Kada se želi prikazati točan položaj satelita unutar neke minute u vremenu, korisnik u prvom dropdown menu može odabrati neku minutu. Dropdown za minute se automatski puni detektiranim vremenima iz podataka.

```
function groupByMinute() {
  groupedData = {};

  satellitePath.forEach(p => {
    const date = new Date(p.timestamp * 1000);
    const key = date.getUTCFullYear() + '-' +
      String(date.getUTCMonth() + 1).padStart(2, '0') + '-' +
      String(date.getUTCDate()).padStart(2, '0') + ' ' +
      String(date.getUTCHours()).padStart(2, '0') + ':' +
      String(date.getUTCMinutes()).padStart(2, '0');

    if (!groupedData[key]) {
      groupedData[key] = [];
    }
    groupedData[key].push(p);
  });

  populateDropdown();
}
```

Funkcija za pronalazak svih raspoloživih minuta

Putanja Satelita

Odaberi minutu putanje, datum: 2022-06-01

Cijela putanja

Odaberi sekundu putanje: -----

Cijela putanja

07:07

07:08

07:09

07:10

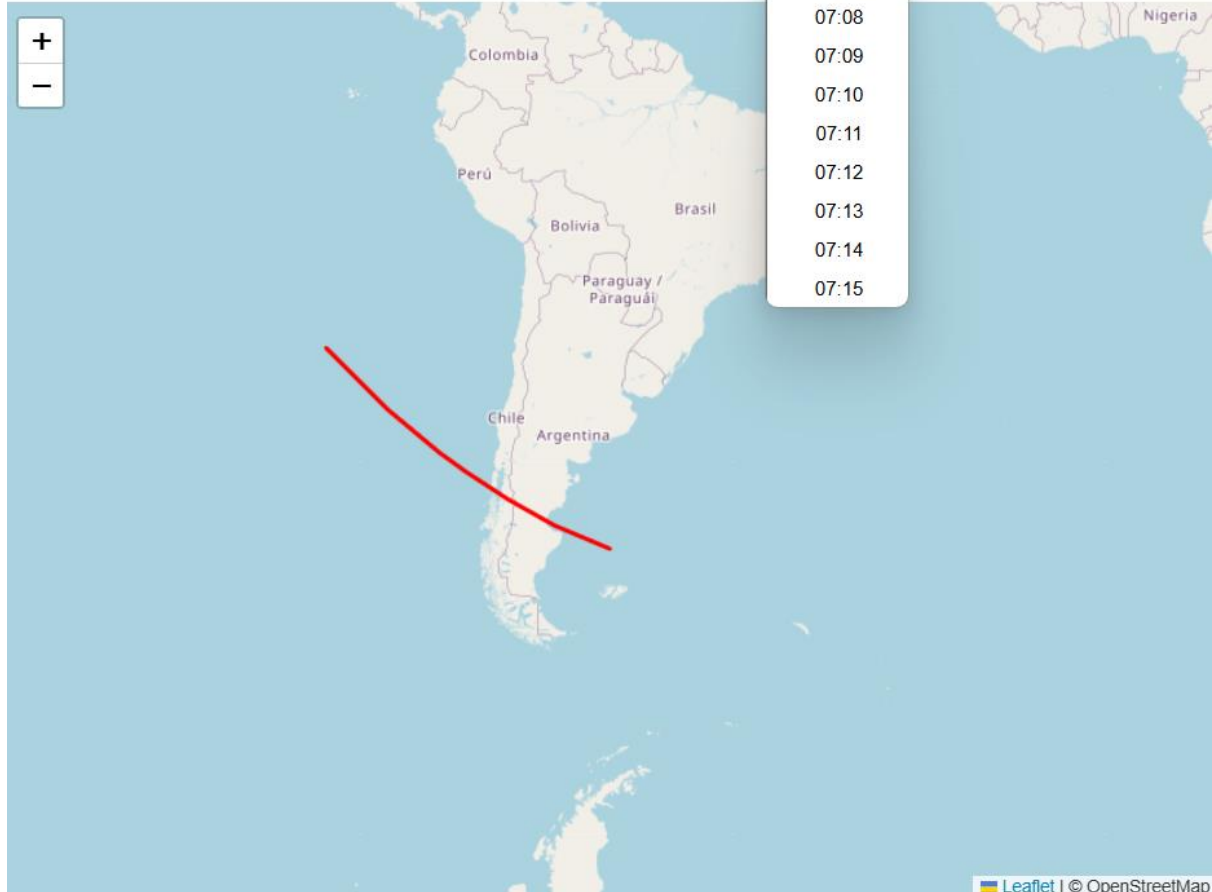
07:11

07:12

07:13

07:14

07:15

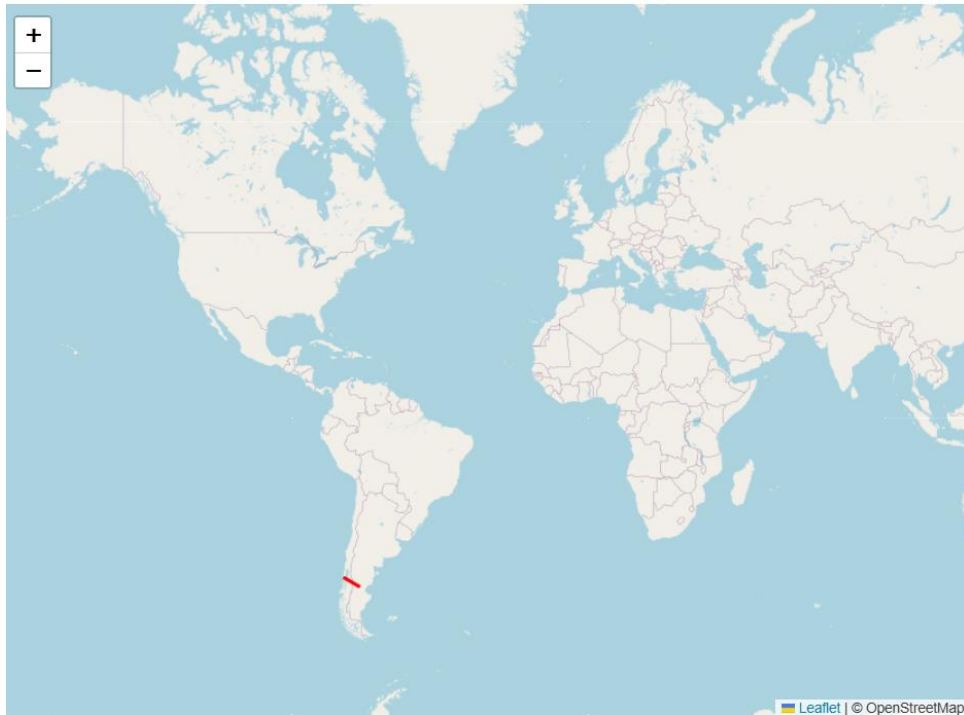


Prikaz odabira točnog vremena u prvom dropdown menu

Putanja Satelita

Odaberi minutu putanje, datum: 2022-06-01

Odaberi sekundu putanje: 07:13



Prikaz putanje nakon odabrane minute

Prikaz sekundi unutar odabrane minute. U funkciji onMinuteSelect(), nakon korisnikovog odabira minute, na sličan se način dobivaju sve dostupne sekunde za odabranu minute.

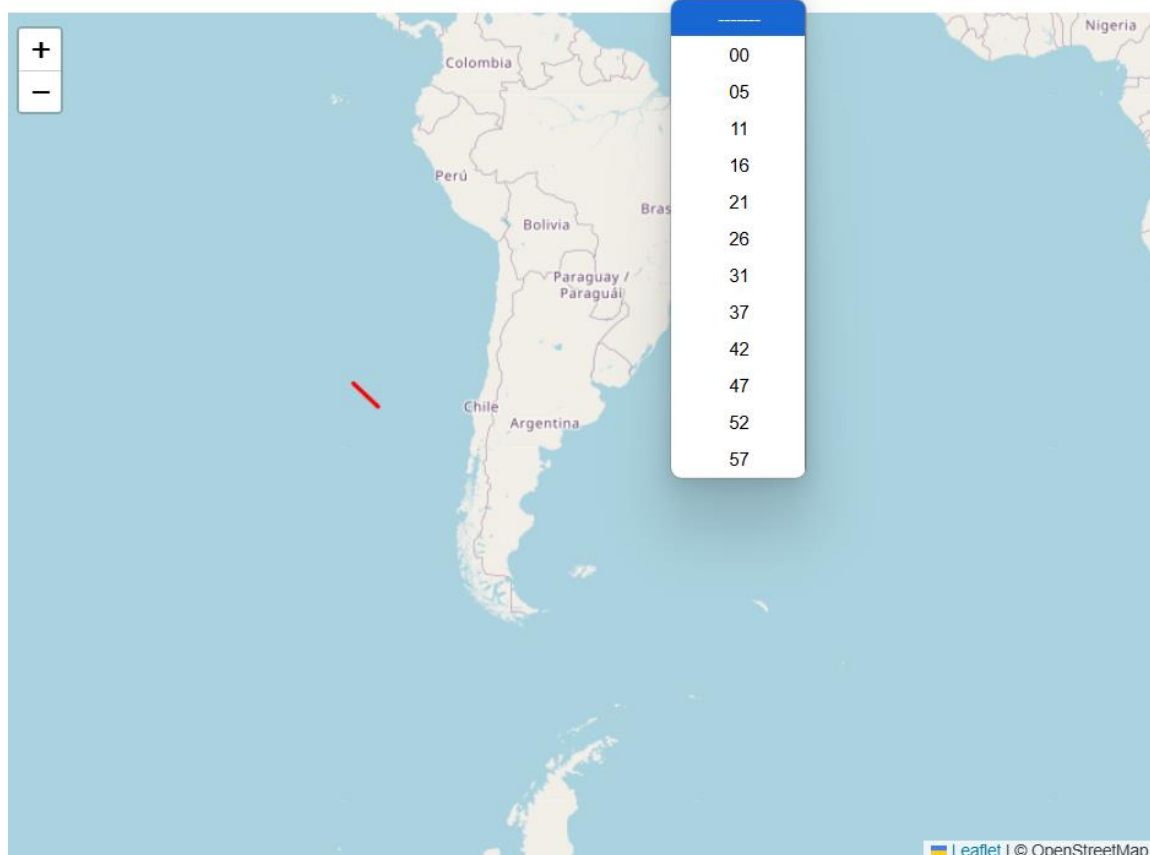
```
points.forEach(p => {  
  const date = new Date(p.timestamp * 1000);  
  const second = String(date.getUTCSeconds()).padStart(2, '0');  
  const label = `${second}`;  
  
  const option = document.createElement('option');  
  option.value = p.timestamp;  
  option.textContent = label;  
  secondSelect.appendChild(option);  
});
```

Pronalazak koordinata za svaku sekundu unutar odabrane minute

Putanja Satelita

Odaberi minutu putanje, datum: 2022-06-01

Odaberi sekundu putanje: 07:09



Prikaz odabira vremena u sekundu

Prikaz točke. Ako korisnik odabere točnu minute i sekundu ispisat će se točka.

```
function onSecondSelect() {
  const secondValue = document.getElementById('secondSelect').value;

  if (marker) { // ukloni prethodne
    map.removeLayer(marker);
    marker = null;
  }

  if (secondValue === "minute") { // nije odabrana sekunda
    onMinuteSelect();
    return;
  }

  const timestamp = parseInt(secondValue, 10);
  const point = satellitePath.find(p => p.timestamp === timestamp);

  if (point) {
    drawPath([]);

    marker = L.circleMarker([point.lat, point.lon], {
      radius: 2,
      color: 'red',
      fillOpacity: 1
    }).addTo(map);

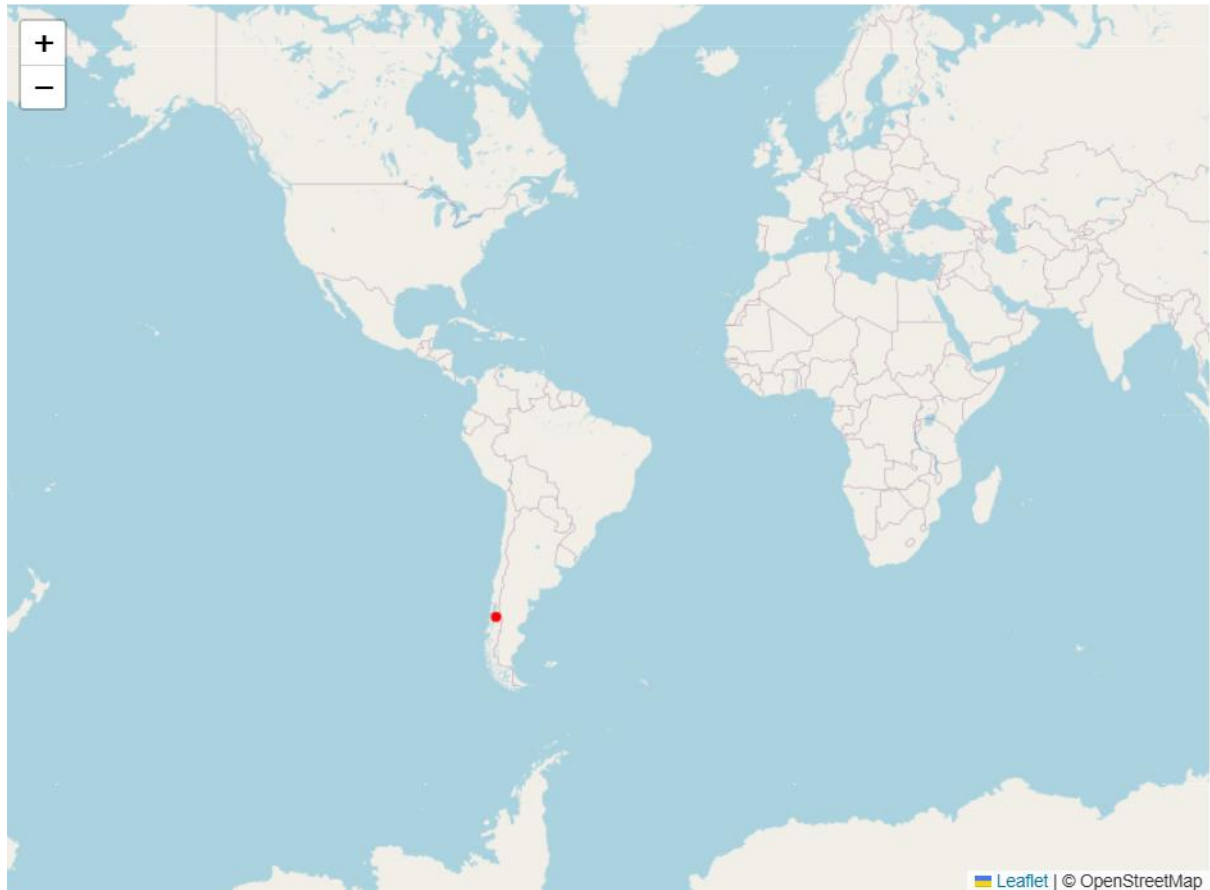
    map.setView([point.lat, point.lon], map.getZoom());
  }
}
```

Funkcija za obradu drugog odabira

Putanja Satelita

Odaberi minutu putanje, datum: 2022-06-01

Odaberi sekundu putanje: 07:13



Prikaz lokacije za točno vrijeme

Implementirana obrada dolaznih HTTP zahtjeva s metodom GET. Metoda handle() obrađuje samo GET zahtjeve jer se ne provodi nikakva obrada ostalih metoda.

```
static class MyHandler implements HttpHandler {
    public void handle(HttpExchange t) throws IOException {
        String uri = t.getRequestURI().getPath();
        if (uri.equals(anObject:"/"))
            uri = "/index.html";
        Path path = Paths.get(first:".", uri);

        if (!Files.exists(path)) {
            String response = "404 Not Found";
            t.sendResponseHeaders(rCode:404, response.length());
            OutputStream os = t.getResponseBody();
            os.write(response.getBytes());
            os.close();
            return;
        }

        byte[] bytes = Files.readAllBytes(path);
        String contentType = "text/html";
        if (uri.endsWith(suffix:".css"))
            contentType = "text/css";
        else if (uri.endsWith(suffix:".js"))
            contentType = "application/javascript";

        // slanje HTTP odgovora
        t.getResponseHeaders().add(key:"Content-Type", contentType);
        t.sendResponseHeaders(rCode:200, bytes.length);
        OutputStream os = t.getResponseBody();
        os.write(bytes);
        os.close();
    }
}
```