Google Maps Project

jan.schulz@devugees.org

Task 1

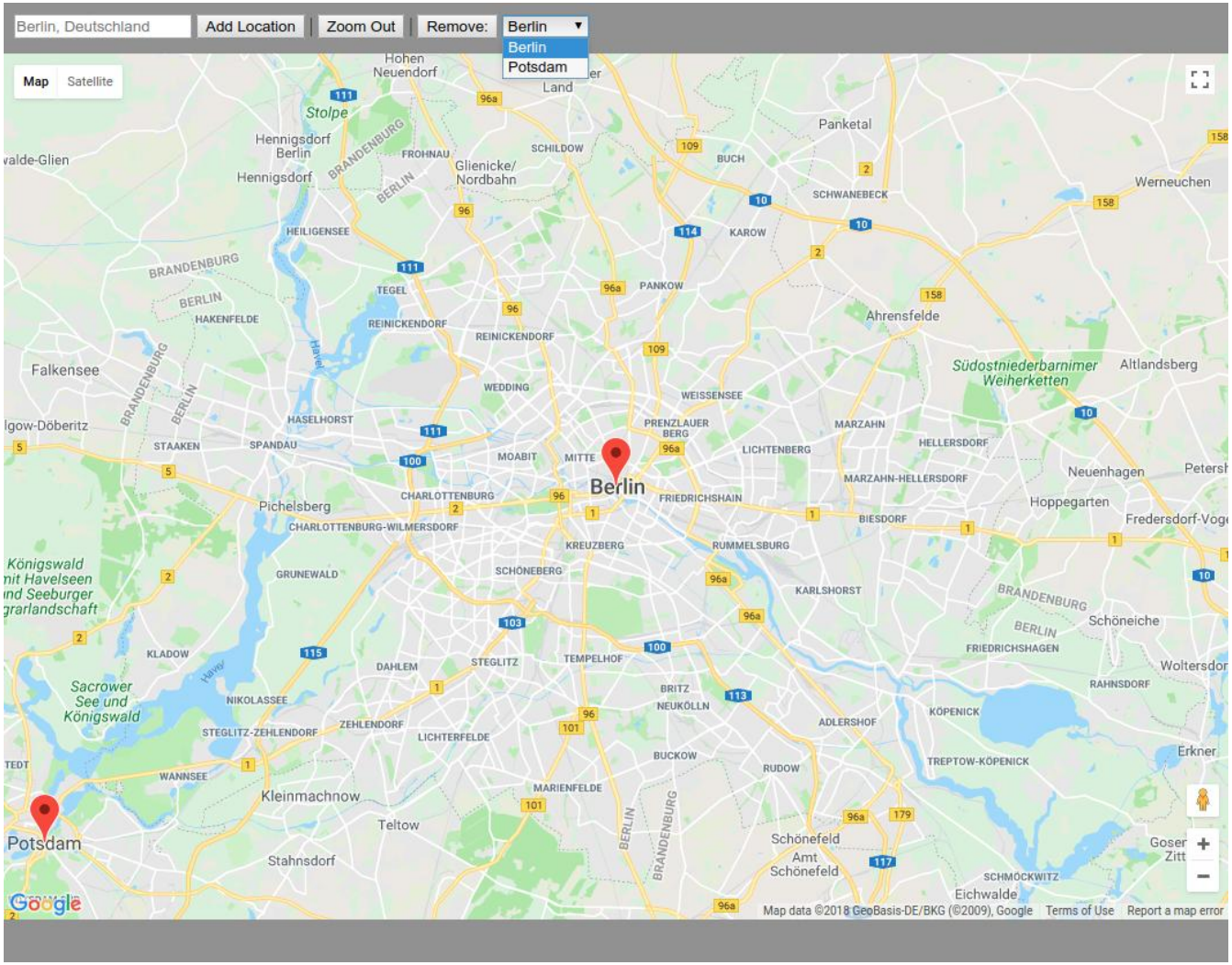
1 – Pull the latest devugees8 repository

2 – Get an API Key from https://cloud.google.com

3 – Go to jquery/gmproject

4 – Analyze gm_1 – gm_3.html – Try to understand it based on the comments and my explanations. Insert your API Key to test it.

5 – Create the following layout in index.html (next slide)

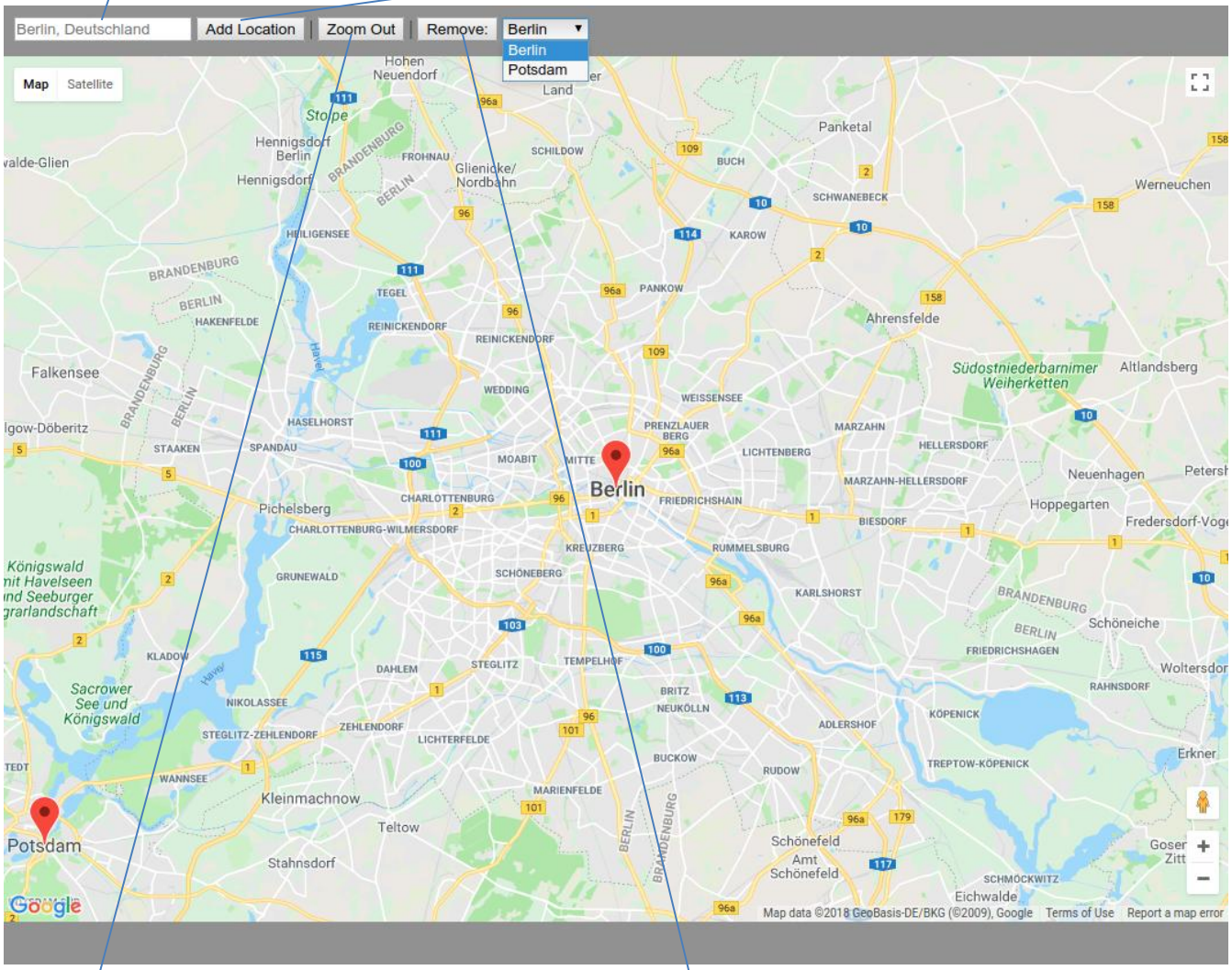

Berlin, Deutschland
Add Location
Zoom Out
Remove:
Berlin
Berlin
Potsdam
Map
Satellite
Berlin
Potsdam
Hohen Neuendorf
Stolpe
Hennigsdorf
Berlin
Hennigsdorf
Glienicke/ Nordbahn
Panketal
Werneuchen
Falkensee
Ahrensfelde
Südostniederbarnimer Weiherketten
Altlandsberg
Pichelsberg
Hoppegarten
Neuenhagen
Königswald mit Havelseen und Seeburger Agrarlandschaft
Sacrower See und Königswald
Schöneiche
Woltersdorf
Erkner
Kleinmachnow
Teltow
Stahnsdorf
Schönefeld
Amt Schönefeld
Eichwalde
Google
Map data ©2018 GeoBasis-DE/BKG (©2009), Google
Terms of Use
Report a map error

Task 2

6 – Add the following features using Jquery. Before you start, think how you can organize your marker points.

The user can enter a geo code here

Adds the geo code as new marker to the map



Zooms out to zoom level 3

Removes the selected location

Task 3

Extend your app to use a backend with the JQuery AJAX functionality.

1 - Hence, go into the jquery/gmprojectex folder and there install the required node modules using:

```
$ npm install express body-parser randomstring path
```

Then, start your server by:

```
$ node server.js
```

(you can exit your server by CTRL+C)

Task 3

2 - The files index.html, main.js and style.css are almost empty. Replace these files by your code or integrate your code in them. At the end, it matters that your frontend code works in the new environment.

3 - When the app loads, perform a GET request to /location and retrieve the locations as JSON.

4 - When the user creates a new location, perform a POST request to /location and send a body like { lat: 12.000, lng: 13.000, title: 'Jamaica' }. You will receive an answer like { error:0, location: { id: ‘8rc39dk’, title: ‘Jamaica’, lat: ‘123’, lng: ‘456’ }}. The id field is important to later properly delete a location on the backend.

5 - When the user removes a location, perform a DELETE request to /location like /location/h48ca, whereas "h48ca" is the location id. If a location was deleted, you receive { errorid: 0 }.