Integrated Project 2 (IP2) ALC: Project Specification August 2018

1 Introduction

Students will work in groups and will undertake the following tasks over one trimester: Analysis, design, implementation, test and evaluation of a web application that accesses a range of online data sources; specific extracts of these data sources are to be represented within the web application, including the use of Google Maps with markers.

It is suggested that students undertake the following tasks in the order presented:

- Read this document
- Run the code examples and examine them in detail
- Re-read this document.

1.1 Technologies

Technologies to be used include HTML, CCS, JavaScript (including JQuery and Google Maps libraries). All aspects of coding are supported by detailed example code with explanatory comments, as well as online tutorial/documentation sources. The web application is to be implemented with client-side code (JavaScript) and makes API calls to access data from external sources; no server-side code is required.

2

2.1 Deliverables

The following deliverables are required:

- A group Report and the web application (70% of total assessment)
- Individual student reflective report (20% of total assessment)
- Presentation (10% of total assessment)

2.2 Structure of the specification document

The remainder of this specification document has the following sections:

- A specification of the required content of the web application
- A brief introduction to the GeoJSON format used by the earthquake data sources
- An introduction to the data sources (earthquake data and weather data). Students will also select another data source of their own choice.
- A set of code examples to introduce the following:
    - Loading data sources
    - Processing the data sources to extract specific data content
    - Using Google Maps
        - Loading a map
        - Placing and labelling markers on the map
        - Creating marker pop-up infoWindow content
        - Executing code when a marker is clicked
        - Creating clusters (groups) of markers when they are close to each other on a map.
    - Dynamic creation of buttons on web pages. The buttons are configured to load specific data sources.

3 Specification

The specification includes an overview and details of the features required.

3.1 Overview

The web application will comprise a set of web pages and a menu to allow the user to access these pages. The styling and layout of the web application is to be defined by the student groups. The web application can be deployed on a web server (including GitHub Pages) or can be run locally (since it does not require any back-end code).

The application must have the following pages:

- An overview page that explains the purpose of the application
- An author page that provides a brief introduction to each of the group members
- A page that introduces GeoJSON
    - This should include a brief overview and links to useful external web pages that explain the GeoJSON format in detail
- An Earthquake mapping page with a set of buttons and a map. The buttons represent different feeds that are to be displayed on the map.
- A Weather page that allows a user to enter a latitude and longitude and to view weather data for that location.
- An additional mapping page that is based upon data selected by the student group (by undertaking some web search for appropriate data sources).

3.2 Detailed Specification

3.2.1 Sample Code

Sample code is discussed in Section 6.

3.2.2 API Keys

In order to run the example code and also run your own code, you will need to register for the following API keys (free):

- Google Maps: https://developers.google.com/maps/documentation/javascript/get-api-key
- APIXU (for Weather data). Register at: https://www.apixu.com/pricing.aspx using the Free plan and its 'Get Started' button.

You will need to use your own API keys in the sample code. Once you have your keys, just do a search in the supplied HTML example source code files for the following keywords:

- YOUR-GOOGLE-MAPS-API-KEY (search for this)
    - In files: IP2_Map1.html, IP2_Map2.html, IP2_Map3.html, IP2_Map4.html, IP2_Map5.html
    - Replace 'YOUR-GOOGLE-MAPS-API-KEY' with your own Google Maps API key
- YOUR- APIXU-KEY (search for this)
    - In files: IP2_Map5.html
    - Replace 'YOUR-APIXU-KEY' with your own apixu API key

3.2.3 Web Page General

Try to make the layout responsive if you have implemented responsive layouts previously. Try to fit content so as to not have the pages scrolling. The menu should be in a regular format/position for each web page.

3.2.4 Overview Page

This should explain the purpose and content of the IP2 Project. It can include a link to a version of the project specification document in a suitable web-readable format (such as PDF).

3.2.5 Author Page

A brief bio (including technical interests) for each student in the group. Include photos if you like (not obligatory).

3.2.6 GeoJSON Overview

Introduce GeoJSON so that a reader who does not already know about this can gain a basic understanding. Use links to a range of external tutorial pages that provide more detailed information about GeoJSON.

3.2.7 Earthquake Mapping Page

To gain the fundamental understanding of the programming methods to be used on this page, students should refer to the example code discussed in section 6 of this report. You will find examples of all aspects that are required for your web application. You need to integrate these to implement the pages as specified below.

The Earthquake Mapping page should comprise the following:

- A set of dynamically created buttons to select all feeds that are available from the following page: https://earthquake.usgs.gov/earthquakes/feed/v1.0/geojson.php
    - See the code example: IP2_Map6.html . You will have to extend the number of buttons to cover all the groups ('Past Hour', 'Past Day', etc) and the individual items for each group ('Significant Earthquakes', 'M4.5+ Earthquakes', etc).
- A Google Map that is freshly populated by clicking any button
    - Must include clustered markers
        - See examples: IP2_Map4.html
    - Must include an infoWindow for each marker.
        - The infoWindow has the content that is included in example: IP2_Map3.html.
    - Must include Weather data. When a marker is clicked two things should happen:
        - The infoWindow is enabled
        - An ajax call fetches weather data, fetches a graphic image of the weather state, populates the web page with 1. the graphic, 2. a text overview of the weather state and 3. any other useful weather data. See example: IP2_Map5.html as a starting point.

3.2.8 Weather Page

This page should allow the user to enter a latitude and longitude and then click a button. The page then displays a current weather graphic image and as much useful data as you can find

within the weather feed. [You need to explore what data is returned from the API call. Use the explorer at: https://www.apixu.com/api-explorer.aspx to see the data that is returned].

Once this is working, add an additional data entry field and additional button; this should allow the user to enter the name of a place, click this button and bring back the current weather.

3.2.9 Additional Mapping Page

This should use any other source of GeoJSON data that you can find. You should look for **point** data rather than **line** data. See the web site: http://geojson.xyz/ for interesting example of data. You should have the following:

- A map with clustered markers
- An infoWindow for each marker, that displays useful information that is part of the data feed.

4 General Advice

- Go through all the code examples, running them and examining each line of code. It is best to work in pairs to do this task, so do this before you leave ALC.
- Run through the examples in order (from IP2_Map1.html onwards).
- When starting to build your own code, add one new feature at a time. When a new feature is working, then save that code as a working version (and don't touch it again!). Use this version as the starter code for the next version. If something goes wrong, you can simply start with the previous working version.
- Save all of your versions on ALL of the following:
    - Online repositories (Cloud providers/GitHub/Bitbucket)
    - Local storage on your laptop
    - USB storage
        - Yes, I said ALL of these. Don't be lazy about this. Something is bound to go wrong, somewhere, at some time. When you tell us that you have lost all your stuff, we will just say: "well, it's good that everything is backed up". If you tell us that you didn't back-up, then we will tell you that you don't get any marks for lost stuff.
        - Back-up frequently. How often? Well, how much work do you want to have to redo if you lose something? Your time is valuable, so save frequently and back up often.
- Split up tasks with your team members ASAP.

4.1 Tools

Use development tools that you familiar and confident with. Test your code using Chrome browser and its Developer Tools. You can easily add breakpoints to your code so that you can stop and examine variables. You can easily breakpoint within the response of an ajax call so that you can see the data returned from an API call.
Use a good quality editing environment for your code. I'm using the free Visual Studio Code environment. Its code highlighting is very useful. You can right-click on a source file in the editor to open the code in your browser. Use GitHub or BitBucket code repositories if you are familiar with these.

5 Data Feeds

5.1 GeoJSON

GeoJSON is a geospatial data interchange format based on JavaScript Object Notation (JSON). It defines several types of JSON objects and the manner in which they are combined to represent data about geographic features, their properties, and their spatial extents. GeoJSON uses a geographic coordinate reference system, World Geodetic System 1984, and units of decimal degrees [1]. The examples that you will use are based on *points* (latitude, longitude). The page at [2] explains GeoJSON at an introductory and comprehensive level.
You can see many examples at: http://geojson.xyz/

You will be accessing GeoJSON data for earthquakes:
U.S Geological Survey. The page at [3] explains the GeoJSON format and also has links to predefined data feeds that you can access. The supplied HTML code examples show exactly how to access this data.

5.2 Weather Data from APIXU

We will source weather data from apixu.com [4]. Register for the free APIXU service. Try the API explorer at: https://www.apixu.com/api-explorer.aspx. This will provide an example of the data returned for a weather query. You are interested in Current Weather. Documents and help are at: https://www.apixu.com/doc/

6 Google Maps Help

Documentation and examples are available for Google Maps at:
https://cloud.google.com/maps-platform/
The following documentation is especially useful:
https://developers.google.com/maps/documentation/javascript/get-api-key
https://developers.google.com/maps/documentation/javascript/adding-a-google-map
https://developers.google.com/maps/documentation/javascript/marker-clustering
https://developers.google.com/maps/documentation/javascript/maptypes

7 Sample Code

Sample code is made available at:
https://drive.google.com/drive/folders/1htM426P26Hobuzg7hmyGPceeCOaszfPj?usp=sharing

File	Overview
IP2_Map1.html	Web page with Google map. Click button to populate map with earthquake markers from a specific USGS data feed. Markers are not labelled. Note that these examples use data sources that end with **.geojson**. Do NOT use sources that end with **.geojsonp**
IP2_Map2.html	Web page with Google map. Click button to populate map with earthquake markers from a specific USGS data feed. Markers are labelled to show the magnitude of the earthquake.
IP2_Map3.html	Web page with Google map. Click button to populate map with earthquake markers from a specific USGS data feed. Markers are labelled to show the magnitude of the earthquake. Each marker has an infoWindow that includes a URL that is part of the GeoJSON data. The infoWindow URL opens in the same browser tab.
IP2_Map4.html	Web page with Google map, earthquake markers and clustering. Click on a cluster to zoom in to see markers (and maybe other clusters).
IP2_Map5.html	Web page with Google map and markers labelled with earthquake magnitude. Accesses weather data when clicking on a Google Maps marker. The weather data is used to populate a weather image and a weather summary.
IP2_Map6.html	Constructing a set of buttons where each one make a different ajax request, each button associated with a specific URL.

8 Examples of Data

8.1 Earthquake Data

The data is shown as an illustration. The data will obviously change over time.
Feed: https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/2.5_hour.geojson


```
{
  "type": "FeatureCollection",
  "metadata": {
    "generated": 1533742458000,
    "url": "https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/2.5_hour.geojson",
    "title": "USGS Magnitude 2.5+ Earthquakes, Past Hour",
    "status": 200,
    "api": "1.5.8",
    "count": 1
  },
  "features": [
    {
      "type": "Feature",
      "properties": {
        "mag": 2.6,
        "place": "6km ENE of Mammoth Lakes, CA",
        "time": 1533741217560,
        "updated": 1533742270324,
        "tz": -480,
        "url": "https://earthquake.usgs.gov/earthquakes/eventpage/nc73066416",
        "detail": "https://earthquake.usgs.gov/earthquakes/feed/v1.0/detail/nc73066416.geojson",
        "felt": 2,
        "cdi": 2.2,
        "mmi": null,
        "alert": null,
        "status": "automatic",
        "tsunami": 0,
        "sig": 104,
        "net": "nc",
        "code": "73066416",
        "ids": ",nc73066416,",
        "sources": ",nc,",
        "types": ",dyfi,focal-mechanism,geoserve,nearby-cities,origin,phase-data,scitech-link,",
        "nst": 30,
        "dmin": 0.003796,
        "rms": 0.05,
        "gap": 53,
        "magType": "md",
        "type": "earthquake",
```

```
      "title": "M 2.6 - 6km ENE of Mammoth Lakes, CA"
    },
    "geometry": {
      "type": "Point",
      "coordinates": [
        -118.9095001,
        37.6525002,
        2.43
      ]
    },
    "id": "nc73066416"
  }
 ]
}
```


Weather Data: OVER/

8.2 Weather Data

The data is shown as an illustration. The data will obviously change over time.
Here is a request that uses latitude and longitude as parameters:
http://api.apixu.com/v1/current.json?key=PUT_YOUR_KEY_HERE&q=-5.5221,151.7805


```
{
  "location": {
    "name": "Buka",
    "region": "East New Britain",
    "country": "Papua New Guinea",
    "lat": -5.52,
    "lon": 151.78,
    "tz_id": "Pacific/Port_Moresby",
    "localtime_epoch": 1533743239,
    "localtime": "2018-08-09 1:47"
  },
  "current": {
    "last_updated_epoch": 1533742339,
    "last_updated": "2018-08-09 01:32",
    "temp_c": 20.1,
    "temp_f": 68.2,
    "is_day": 0,
    "condition": {
      "text": "Partly cloudy",
      "icon": "//cdn.apixu.com/weather/64x64/night/116.png",
      "code": 1003
    },
    "wind_mph": 3.4,
    "wind_kph": 5.4,
    "wind_degree": 214,
    "wind_dir": "SW",
    "pressure_mb": 1012.0,
    "pressure_in": 30.3,
    "precip_mm": 0.0,
    "precip_in": 0.0,
    "humidity": 86,
    "cloud": 9,
    "feelslike_c": 20.1,
    "feelslike_f": 68.2,
    "vis_km": 18.0,
    "vis_miles": 11.0
  }
}
```

9 Bibliography

[1] http://geojson.org/, “GeoJSON,” [Online]. Available: http://geojson.org/.
[2] “More than you ever wanted to know about GeoJSON,” [Online]. Available: https://macwright.org/2015/03/23/geojson-second-bite.html.
[3] “GeoJSON Summary Format,” [Online]. Available: https://earthquake.usgs.gov/earthquakes/feed/v1.0/geojson.php.
[4] “Apixu,” [Online]. Available: https://www.apixu.com/.
[5] http://geojson.org/. [Online]. Available: https://tools.ietf.org/html/rfc7946.