

PORTFOLIO WEBSITE SPECIFICATION

Project #1: “GAZETTEER”

PREPARED BY

Paddy Evans

IT CAREER SWITCH

EXECUTIVE SUMMARY

This application is envisaged as a “mobile first” website that will operate equally well on desktop computers. It will provide profiling for all countries through the presentation of demographic, climatic, geographical and other data.

The system will make extensive use of third party APIs. Some candidates are suggested below but you are strongly encouraged to seek out more to add value to your solution.

In order to complete this project the student will require access to a PHP enabled web server. This can be achieved either by using localhost with IIS or Apache on a standalone PC or by subscribing to an internet hosting account which will involve a modest cost and the registration of a domain name. The latter is preferable as it will give you the best platform to display your portfolio publicly.

This project will form the first part of your portfolio that will showcase your talents as a software developer and a solution provider to potential employers. Remember that you are not the only one attempting it and so this is your chance to prove that you can supply a better thought out, better looking and easier to use product than anyone else.

Before launching yourself into writing a system to fulfill the system specification, give yourself time to absorb it, research the techniques that you believe are required and decide upon the functionality that you are going to deliver. Once you have done this then you will need to produce a short document for me that outlines this. It should be brief but it should also contain enough detail to convince me that you are on the right lines. Once I am happy that you are, then you can begin the development in earnest.

Good luck and please do not hesitate to contact me should you require help, guidance or clarification.

paddy@itcareerswitch.co.uk

1. Project Overview

This document describes the core functionality of the proposed system along with a suggestion of how the website may be presented. The final design and mode of operation is down to you and you will be expected to exploit many of the techniques that you have learnt in the Code Academy courses.

However, you are not constrained by the contents of the courses and you may decide to employ other frameworks and libraries to achieve the end result. If at any point you wish to talk over your approach or specific design or programming problems then please feel free to contact me and, if necessary, arrange for a telephone call.

2. Minimum skill set

The ability to deploy the following will be required to complete this project:

- HTML / CSS (preferably with a framework)
- JavaScript
- JQuery DOM manipulation / AJAX
- PHP cURL

3. Additional resources

The following third party APIs are recommended (and all of them have restricted limits for free plans) but you may decide to use others that you are already familiar with or have identified as suitable replacements:

- [OpenCage](#)
- [Open Weather](#)
- [Geonames](#) (register at <http://www.geonames.org/login>, activate your account and then login and enable web services)
- [Rest Countries](#)
- [Open Exchange Rates](#)

Additionally, deployment of the open source Leaflet library will be required to display maps and overlays of your data. [Introductory tutorials may be found here](#). Please do not follow the suggestion in the examples of using Mapbox as a tile provider and instead investigate this link for a [demo of free tile providers](#).

Look at the functionality ([L.geoJSON](#)) that allows you to easily add multiple markers and polylines to a **FeatureGroup** instead of directly on to the map and make good use of the

map.fitBounds() function that will help you to optimise the display of features to the available height and width of the map.

Other resources that you should be looking at including to improve the user experience are:

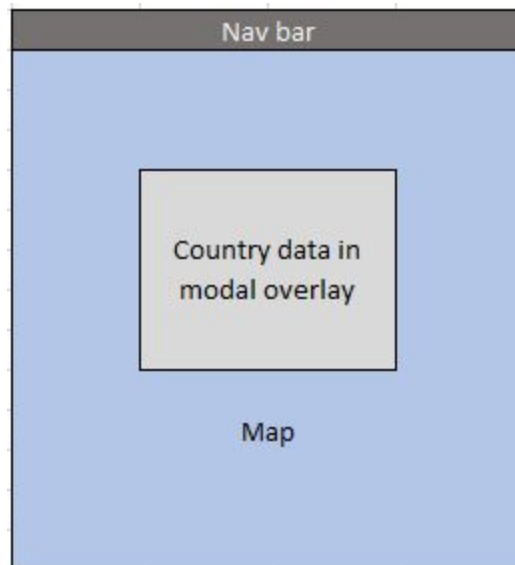
- Leaflet.markercluster
- Leaflet.easyButton
- Leaflet.extraMarker or equivalent (or just use L.Icon to create your own markers)

You should also overlay as much data as you can source data on the map but don't just settle for displaying a marker or two and try to make it as visually appealing as possible.

4. Interface / design

I recommend that you attempt to implement this as a one-page, full screen application that works equally well across all devices.

Using a framework like Bootstrap will ease the task of design and a good approach would be to have a standard nav bar with a select tag for the countries and the map taking up all of the rest of available space on screen.



The following file, [countryBorders.geo.json](#), contains a list of all countries with ISO2 and ISO3 codes as well as polygons describing the border. You can use this in two ways; one,

to create a PHP routine to return the ISO code and name for inclusion in the <select> and, two, to create a PHP routine to parse the JSON object to return a border where the entry matches a given country code.

On loading, the system should be able to determine the current location of the device and display a map highlighting the country that the user is currently in. This may be achieved using the JavaScript **navigator** command.

The latitude and longitude data returned by this can then be used by AJAX functions to make calls to PHP routines that you design to call the above APIs. The data that they return for a country can then be arranged and displayed in the modal dialogs. This will require putting some thought into how this section may best be designed to optimise the display of at the very least the following data:

- Country name / Capital city
- Population
- Currency / current exchange rate
- Current weather / weather forecast
- Wikipedia links

The above are an absolute minimum so please try to identify other APIs to retrieve interesting facts and geodata to overlay on the map to make your application as engaging as possible.

Make the best use of attractive imagery to display flags and currencies and other relevant data. Supply as much information as you can; Geonames has several APIs that will also provide nearby points of interest and links to Wikipedia - even a history of earthquake activity.

Use Chrome DevTools to resize the application to various devices to make sure that your solution works well across all of them.

5. Program flow

This is an overview as to how you could approach the design of this system but the final decision is up to you. Try to imagine how the system could be used and plan the functionality to fit how you envisage the user experience:

- Display loader whilst HTML renders.
- JQuery onload retrieve current location.
- Update map with currently selected country border.

- AJAX call to PHP routine to OpenCage using location data to return core information.
- AJAX calls to PHP routines to other API providers using information from OpenCage.