El Mapa - Portfolio Project Number 1

There comes a time when you find yourself lost, you go through a door and forget why you are there, or what you are looking for, you seem to have been neuralyzed. It is for those times that we have developed El Mapa.

El Mapa is a web application that will run seamlessly on both mobile and desktop. On load, it will provide you with a map centred on your own location and (dependant on internet speeds) instant access to a mountain of information such as current position on the planet, weather and even the currency of preference so that the next time you go through your front door, you will know everything about what is going on outside.

Here is how we are going to do it

On load, a jQuery.ajax call will be made to getIsoCountries.php, which will return an object (isoCountries) associating all countries with their ISO codes. On fulfilment of this request, isoCountries will be mapped and its elements used to populate a drop-down list of countries at the top of the screen[1].

At this point a map will be initialized and displayed with the use of library Leaflet.js

The method `window.navigator.geolocation.getCurrentPosition()` will be used to retrieve the user's location and the data returned will be used on a call to api.geonames.org/countryCode which returns the ISO country code for the user's location. In case of failing to determine the device's location, the user will be able to select a country from a drop-down list.

The ISO country code will be used on ajax calls to:

- getBorders.php: returns the country's borders in geoJSON format.
- api.geonames.org/info?country=<country-code>: provides the bounding box coordinates of the given country.

With this data, and the help of Leaflet's `.geoJSON()` method, a polygon will be added to the map outlining the user's country and the map will then be re-centred by passing the bounding box coordinates to `L.flyTo()`.

After this a series of asynchronous calls will be made to different APIs in order to retrieve information about their location. This calls will be made by PHP routines via ajax and include the following APIs:

- api.geonames.org/earthquakesJSON
- api.geonames.org/findNearbyWikipediaJSON
- api.geonames.org/gtopo30JSON
- openexchangerates.org/api

1 See screen layout on annexed files.

- openweathermap.org/data/2.5/forecast/weather
- aviation-edge.com/v2/public/nearby
- newsapi.org/v2/top-headlines
- reverse.geocoder.ls.hereapi.com/6.2/reversegeocode.json
- wft-geo-db.p.rapidapi.com/v1/geo/cities

The data retrieved from the above includes population, area, altitude, currency, exchange rate, facts of interest, weather, news, landmarks and cities and will be located in one out of four locations on the screen:

1. A marker on the user's location will display all location-specific data,
2. a marker on the capital city will display country-specific data,
3. an expandable widget bar at the top of the screen will display other data not related to specific locations on the map, and finally
4. landmarks and cities will have their own markers.

Annex

getIsoCountries.php

On request, the program will open file countryBorders.geo.json, map the `file['features']` array and return an object associating all country names to their ISO codes:

```
$isoCountries = ['country-name' => [
                'iso_a2' => "",
                'iso_a3' => "",
                'iso_n3' => ""
                ]
]
```

getBorders.php

On request the program will open file countryBorders.geo.json, map the `file['features']` array and return the element whose `properties['iso_3']` property matches the one requested.