# JavaScript for Front-End Exam Preparation – Continental Drift

You have been employed by a tourist company to help them realize their goal of a global country database application. Your task is to write the Front-End part of an application which extracts data from a Firebase database, and renders it into HTML.

There are a few simple tasks your employer would like you to do, to test your skills. Those little tasks will ultimately combine into the functionality you need for the application itself.

Create a script.js file which you will use for the problems.

## Rendering Sample Data – 30 pts

Write **3 JavaScript functions** which receive an **associative array** (**object**), and **render** the **data** from it.

The **object** looks like this:

continents: {

continent1: { some object },

continent2: { some object },

. . .

}

The continents are **different objects** which look like this:

continent: {

name: continentName,

countries: { some collection... }

}

* Continent Name – a simple string
* Continent Countries – an associative array with objects

The countries collection **MAY** or **MAY NOT** exist in the continent object. You should check that in order to assure maximum correctness

The countries collection looks like this:

countries: {

country1Name: {

name: countryName,

capital: countryCapital,

officialLanguage: countryOfficialLanguage,

population: countryPopulation,

area: countryArea,

politicalStatus: countryPoliticalStatus,

{monarch/president}: countryLeader,

officialCurrency: countryOfficialCurrency

},

country2Name: {

...

}

The country objects in the collection will **ALWAYS** have **all** of **their properties**. The country object looks like this:

* Country Name – a simple string
* Country Capital – a simple string
* Country Official Language – a simple string
* Country Population – a simple integer number
* Country Area – a simple floating-point number
* Country Political Status – a simple string
  + Value will either be “Republic” or “Monarchy”
* Country Leader – a simple string
  + If the Country Political Status is “Republic”, the country object will hold a “president” property.
  + If the Country Political Status is “Monarchy”, the country object will hold a “monarch” property.
* Country Official Currency – a simple string

Now that you are familiar with all the data, you should render it. The rendering is not complicated, just print it on the console with console.log()...

You will have to write **3 functions** which produce **3 different outputs** and receive different parameters.

### renderAllContinents() function

Write a JavaScript function - renderAllContinents() which accepts the whole continents object, and renders **ONLY** the NAMES of all continents inside.

The output format is:

{continent1name}

{continent2name}

. . .

### renderSingleContinent() function

Write a JavaScript function - renderSingleContinent() which accepts a continent object, and renders the **name** of the continent and **ONLY** the **NAMES** of the countries inside.

The output format is:

{continentName}

Countries

$$${country1Name}

$$${country2Name}

. . .

### renderSingleCountry() function

Write a JavaScript function - renderSingleCountry() which accepts a country object, and renders all of its properties.

**NOTE**: If the country’s politicalStatus is “Republic” a “President” should be rendered, if it is a “Monarchy”, a “Monarch” should be rendered.

The output format is:

|  |  |
| --- | --- |
| **Republic** | **Monarchy** |
| {countryName}  Capital: {countryCapital}  Official Language: {countryOfficialLanguage}  Population: {countryPopulation}  Area: {countryArea} km2  Political Status: {countryPoliticalStatus}  President: {countryPresident}  Official Currency: {countryOfficialCurrency} | {countryName}  Capital: {countryCapital}  Official Language: {countryOfficialLanguage}  Population: {countryPopulation}  Area: {countryArea} km2  Political Status: {countryPoliticalStatus}  Monarch: {countryMonarch}  Official Currency: {countryOfficialCurrency} |

See the examples below for more information.

### Example

|  |
| --- |
| script.js |
| **let *continents*** = {  **Europe**: {  **name**: **"Europe"**,  **countries**: {  **Bulgaria**: {  **name**: **"Bulgaria"**,  **capital**: **"Sofia"**,  **officialLanguage**: **"Bulgarian"**,  **population**: 7000000,  **area**: 111000,  **politicalStatus**: **"Republic"**,  **president**: **"Rumen Radev"**,  **officialCurrency**: **"LEV(BGN)"** },  **Vatican**: {  **name**: **"Vatican"**,  **capital**: **"Vatican City"**,  **officialLanguage**: **"Italian"**,  **population**: 1000,  **area**: 0.44,  **politicalStatus**: **"Monarchy"**,  **monarch**: **"Francis"**,  **officialCurrency**: **"Euro(EUR)"** }  }  },  **Asia**: {  **name**: **"Asia"**,  **countries**: {  **Russia**: {  **name**: **"Russia"**,  **capital**: **"Moscow"**,  **officialLanguage**: **"Russian"**,  **population**: 144463451,  **area**: 17075200,  **politicalStatus**: **"Republic"**,  **president**: **"Vladimir Putin"**,  **officialCurrency**: **"Russian ruble(RUB)"** },  **China**: {  **name**: **"China"**,  **capital**: **"Beijing"**,  **officialLanguage**: **"Chinese"**,  **population**: 1403500365,  **area**: 9596961,  **politicalStatus**: **"Republic"**,  **president**: **"Xi Jinping"**,  **officialCurrency**: **"Renminbi(CNY)"** }  }  } };  *renderAllContinents*(***continents***); *renderSingleContinent*(***continents***[**'Europe'**]); *renderSingleCountry*(***continents***[**'Europe'**][**'countries'**][**'Bulgaria'**]); |

### Output

|  |  |  |
| --- | --- | --- |
| renderAllContinents() | renderSingleContinent() | renderSingleCountry() |
| Europe  Asia | Europe  Countries:  $$$Bulgaria  $$$Vatican | Bulgaria  Capital: Sofia  Official Language: Bulgarian  Population: 7000000  Area: 111000 km2  Political Status: Republic  President: Rumen Radev  Official Currency: LEV(BGN) |

|  |
| --- |
| script.js |
| **function** *renderAllContinents*(continents) {  *//****TODO: Implement me ...*** }  **function** *renderSingleContinent*(continent) {  *//****TODO: Implement me ...*** }  **function** *renderSingleCountry*(country) {  *//****TODO: Implement me ...*** } |

## Rendering Data in HTML – 30 pts

Now that your employer knows that you are good in JavaScript, he wants to see some DOM manipulation skills. You will be given an HTML which will work as a template for you. You will receive a continent object, like the one in the previous task.

In order for the HTML to look tidy and well formed, CSS needs to be applied, which suggests some specific element hierarchy and classes. Check the template and make sure your function renders the object correctly. You can see input and output below.

Create your functions for this task inside the script.js file, and implement them so that they render the data inside the render-data.html file.

You already have some functions which render some data, you can probably use at least 50% of the code you used in the previous problem. Check in your solution from the previous problem, what can be **REUSED** in this problem.

### Example

|  |
| --- |
| script.js |
| **let *continents*** = {  **Europe**: {  **name**: **"Europe"**,  **countries**: {  **Bulgaria**: {  **name**: **"Bulgaria"**,  **capital**: **"Sofia"**,  **officialLanguage**: **"Bulgarian"**,  **population**: 7000000,  **area**: 111000,  **politicalStatus**: **"Republic"**,  **president**: **"Rumen Radev"**,  **officialCurrency**: **"LEV(BGN)"** }  }  } };  *renderDataInHTML*(***continents***); |

### Input

|  |
| --- |
| render-data.html |
| <!DOCTYPE **html**> <**html lang="en"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>Continental Rift</**title**>  <**link href="reset.css" rel="stylesheet" type="text/css"**>  <**link href="show-style.css" rel="stylesheet" type="text/css"**>  <**script src="https://code.jquery.com/jquery-3.2.1.js"**></**script**> </**head**> <**body**> <**div class="container"**>  <**div class="title"**>  <**h1**>Continental Drift</**h1**>  </**div**>  <**div class="content"**>  <**div class="continents"**>  </**div**>  <**div class="continent-data"**>  </**div**>  <**div class="continent-country"**>  </**div**>  </**div**> </**div**> <**script src="script.js"**></**script**> </**body**> </**html**> |

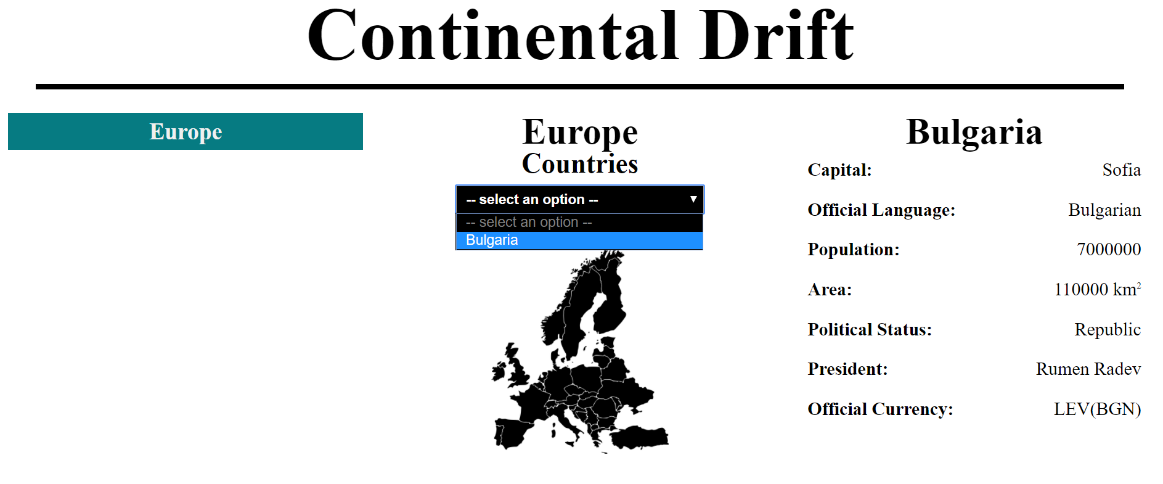
### Output

|  |
| --- |
| render-data.html |
| <!DOCTYPE **html**> <**html lang="en"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>Continental Rift</**title**>  <**link href="reset.css" rel="stylesheet" type="text/css"**>  <**link href="show-style.css" rel="stylesheet" type="text/css"**>  <**script src="https://code.jquery.com/jquery-3.2.1.js"**></**script**> </**head**> <**body**> <**div class="container"**>  <**div class="title"**>  <**h1**>Continental Drift</**h1**>  </**div**>  <**div class="content"**>  <**div class="continents"**>  <**div class="continent"**>  <**h5 class="continent-title"**>Europe</**h5**>  </**div**>  </**div**>  <**div class="continent-data"**>  <**h2 class="continent-title"**>Europe</**h2**>  <**h3 class="countries-title"**>Countries</**h3**>  <**div class="countries"**>  <**select class="dropdown-select"**>  <**option disabled selected value**> -- select an option -- </**option**>  <**option value="Bulgaria"**>Bulgaria</**option**>  </**select**>  </**div**>  <**div class="continent-image"**>  <**img src="images/europe.png"**>  </**div**>  </**div**>  <**div class="continent-country"**>  <**div class="country-title"**>  <**h2**>Bulgaria</**h2**>  </**div**>  <**div class="country-data"**>  <**div class="country-capital"**><**strong**>Capital:</**strong**> <**div**>Sofia</**div**></**div**>  <**div class="country-official-language"**><**strong**>Official Language:</**strong**><**div**>Bulgarian</**div**></**div**>  <**div class="country-population"**><**strong**>Population:</**strong**> <**div**>7000000</**div**></**div**>  <**div class="country-area"**><**strong**>Area:</**strong**> <**div**>110000 km<**sup**>2</**sup**></**div**></**div**>  <**div class="country-political-status"**><**strong**>Political Status:</**strong**><**div**>Republic</**div**></**div**>  <**div class="country-president"**><**strong**>President:</**strong**><**div**>Rumen Radev</**div**></**div**>  <**div class="country-official-currency"**><**strong**>Official Currency:</**strong**><**div**>LEV(BGN)</**div**></**div**>  </**div**>  </**div**>  </**div**> </**div**> <**script src="script.js"**></**script**> </**body**> </**html**> |

In the resources you have been given a folder **images**. You need to load a simple **image div** with the **image** that holds the **name** of the **continent**. See the **HTML** output. It should be enough to help you do it.

**Attach** the name of the continent and the dropdown menu select element to the .continent-data div element, and **attach** the country **data** to the .continent-country div element.

If you do everything correctly, you should see this:



**JQuery** may be used in this problem. It is **ADVISED** that you use **JQuery**, but **native DOM manipulation** is also allowed.

|  |
| --- |
| script.js |
| **function** *renderDataInHTML*(continents) {  *//****TODO: Implement me ...*** } |

## Capture Click event and Dropdown menu Choice – 30 pts

Your employer has acknowledged your JavaScript skills. But now he wants you to do some tasks of handling click events and dropdown menu choice.

Render all continents (**ONLY** the **CONTINENT NAMES**) by **REUSING** the code from the **previous task**, into events.html.

### Click Event

You will be given a **styling** that hides the .continent-data and .continent-country divs.

Your task is to **attach** a click event to each .continent div, which shows the .continent-data and .continent-country divs upon **click**, and **RENDERS** the **DATA** from the **chosen continent**.

**HINT**: Use the functionality from renderDataInHTML() from the **previous task**.

**NOTE**: If you **click** **another** .continent div you should **render** the **NEW CONTINENT DATA**.

**NOTE**: If you **click the same** .continent div, which is **CURRENTLY being shown**, you should **HIDE** the .continent-data and .continent-country divs.

A Continents-Click.GIF has been added to the resources, which displays this functionality.

### Example

|  |
| --- |
| script.js |
| **let *continents*** = {  **Europe**: {  **name**: **"Europe"**,  **countries**: {  **Bulgaria**: {  **name**: **"Bulgaria"**,  **capital**: **"Sofia"**,  **officialLanguage**: **"Bulgarian"**,  **population**: 7000000,  **area**: 111000,  **politicalStatus**: **"Republic"**,  **president**: **"Rumen Radev"**,  **officialCurrency**: **"LEV(BGN)"** },  **Vatican**: {  **name**: **"Vatican"**,  **capital**: **"Vatican City"**,  **officialLanguage**: **"Italian"**,  **population**: 1000,  **area**: 0.44,  **politicalStatus**: **"Monarchy"**,  **monarch**: **"Francis"**,  **officialCurrency**: **"Euro(EUR)"** }  }  },  **Asia**: {  **name**: **"Asia"**,  **countries**: {  **Russia**: {  **name**: **"Russia"**,  **capital**: **"Moscow"**,  **officialLanguage**: **"Russian"**,  **population**: 144463451,  **area**: 17075200,  **politicalStatus**: **"Republic"**,  **president**: **"Vladimir Putin"**,  **officialCurrency**: **"Russian ruble(RUB)"** },  **China**: {  **name**: **"China"**,  **capital**: **"Beijing"**,  **officialLanguage**: **"Chinese"**,  **population**: 1403500365,  **area**: 9596961,  **politicalStatus**: **"Republic"**,  **president**: **"Xi Jinping"**,  **officialCurrency**: **"Renminbi(CNY)"** }  }  } };  *attachEvents*(continents); //**pass the continents object** |

|  |
| --- |
| events.html |
| <!DOCTYPE **html**> <**html lang="en"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>Continental Rift</**title**>  <**link href="reset.css" rel="stylesheet" type="text/css"**>  <**link href="style.css" rel="stylesheet" type="text/css"**>  <**script src="https://code.jquery.com/jquery-3.2.1.js"**></**script**> </**head**> <**body**> <**div class="container"**>  <**div class="title"**>  <**h1**>Continental Drift</**h1**>  </**div**>  <**div class="content"**>  <**div class="continents"**>  </**div**>  <**div class="continent-data"**>  </**div**>  <**div class="continent-country"**>  </**div**>  </**div**> </**div**> <**script src="script.js"**></**script**> </**body**> </**html**> |

You should see this:



As you see nothing is still rendered in the .continent-country div. We’ll get to that soon. As you see the continent is rendered with its **name**, **image** and **countries**, like in the **previous task**.

### Dropdown Menu Choice Event

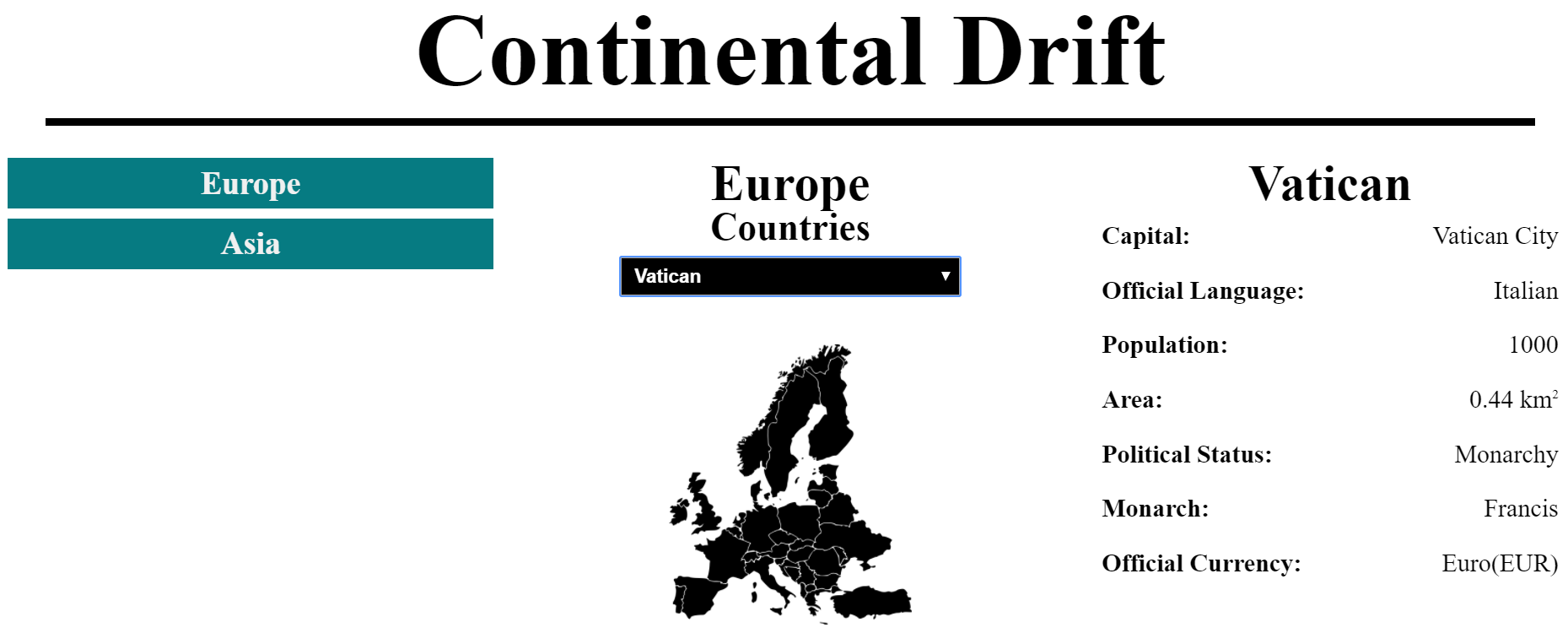
Attach an event, which upon **choice** of option from the dropdown-menu select, **RENDERS** the **DATA** from the **country** of choice.

**NOTE:** If a **new country** is chosen, you should **render** the **NEW COUNTRY DATA**.

**NOTE**: If you **CHOOSE** a **NEW CONTINENT**, the .country-data div should be **EMPTIED**.

**HINT**: Use the functionality from renderDataInHTML() from the **previous task**.

You should see this:



A Countries-Dropdown.GIF has been added to the resources, which displays this functionality.

|  |
| --- |
| script.js |
| **function** *attachEvents*(continents) {  *//****TODO: Implement me...*** } |

## Obtaining Data with AJAX and Rendering Dynamically – 10 pts

Now that your employer is sure of your skills, the time has come for you to combine what you have done in the previous tasks into one completed application.

You have been tasked to extract data about **continents** from a Firebase database, and render it into the **HTML** into a specific format.

You see where we are going? All the tasks you’ve done so far, combine in order to implement a complete application. Check your previous code, and **REUSE EVERYTHING POSSIBLE**.

* **1st**, send a GET request extract all continents, but **RENDER** **ONLY** the **CONTINENT NAMES**. Render them in the **.continents div**. Then attach their **click events**.
* **2nd**, when a continent is **clicked**, you should send **another** **GET request** to **LOAD** the continent’s data and countries (**only** **COUNTRY NAMES**). **Render** the **data** in the **.continent-data div**.
* **3rd**, when a country is **selected** from the **dropdown menu**, send **another GET request** to **LOAD** the country’s data. **Render** the **data** in the .country-data div.

The **following URL’s** are for the **Firebase data**:

* **All Continents**
  + https://continental-drift.firebaseio.com/continents.json
* **Single Continent**
  + https://continental-drift.firebaseio.com/continents/{continentName}.json
* **Single Country**
  + https://continental**-**drift.firebaseio.com/continents/{continentName}/countries/{countryName}.json

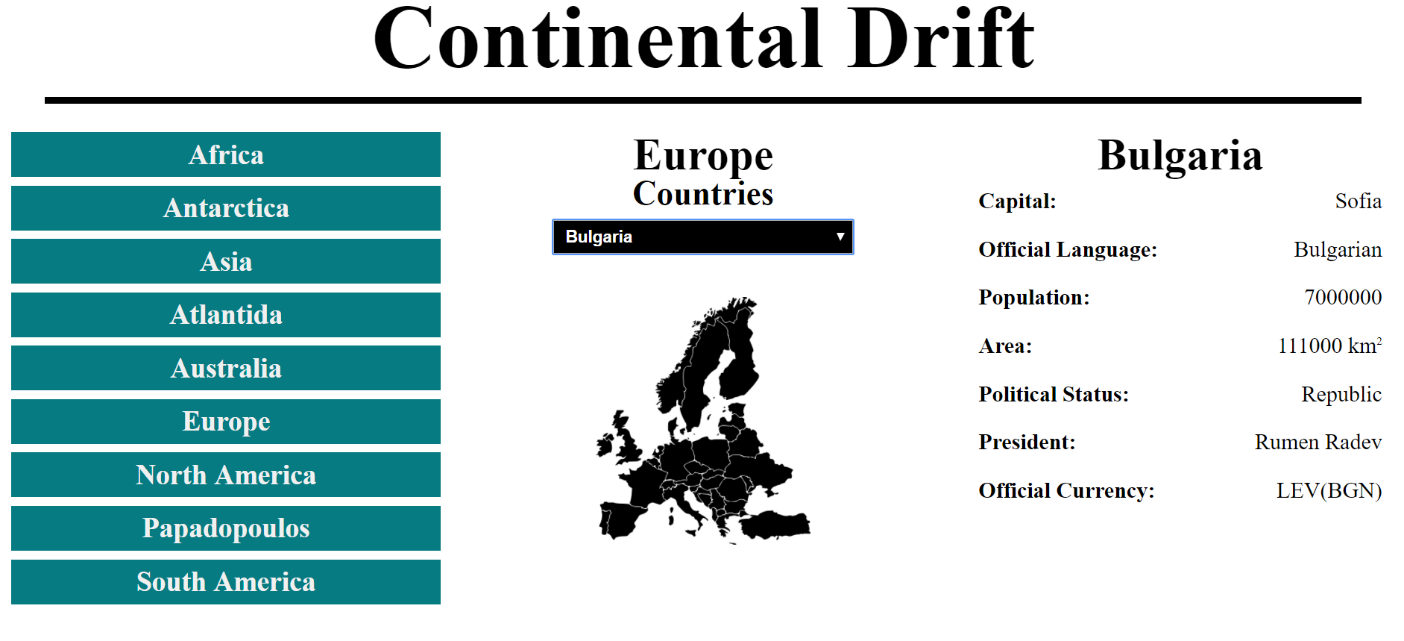
**REUSE all functionality** from the **previous tasks**.

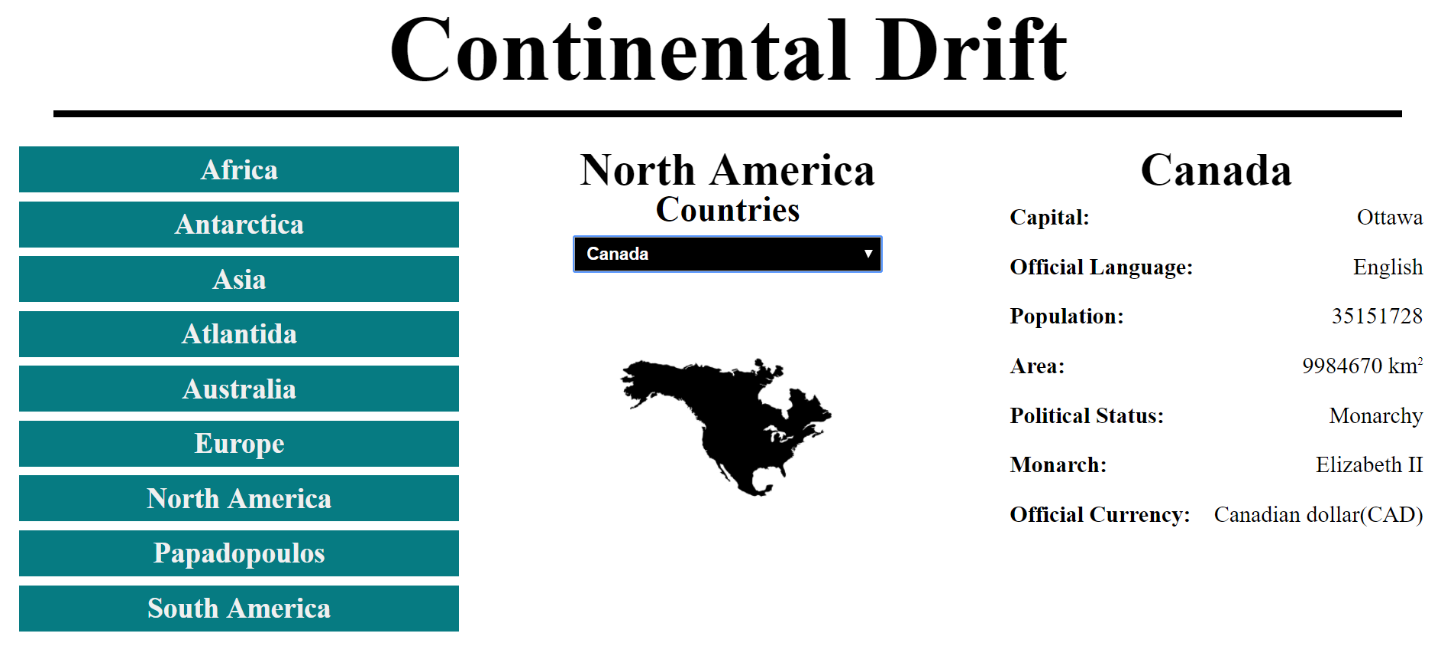
You will see several **screenshots** below which **describe** the **functionality**.

Here is the final HTML file you will need.

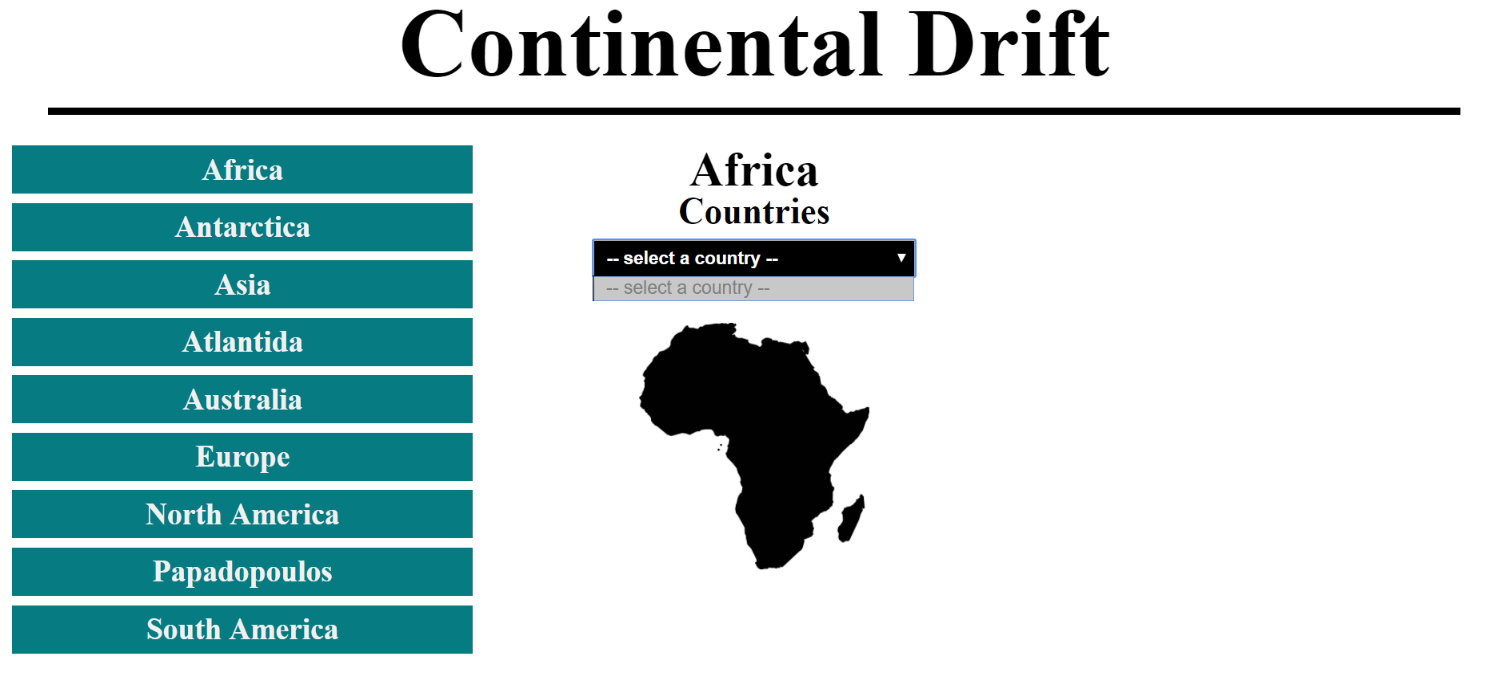
|  |
| --- |
| index.html |
| <!DOCTYPE **html**> <**html lang="en"**> <**head**>  <**meta charset="UTF-8"**>  <**title**>Continental Rift</**title**>  <**link href="reset.css" rel="stylesheet" type="text/css"**>  <**link href="style.css" rel="stylesheet" type="text/css"**>  <**script src="https://code.jquery.com/jquery-3.2.1.js"**></**script**> </**head**> <**body**> <**div class="container"**>  <**div class="title"**>  <**h1**>Continental Drift</**h1**>  </**div**>  <**div class="content"**>  <**div class="continents"**>  </**div**>  <**div class="continent-data"**>  </**div**>  <**div class="continent-country"**>  </**div**>  </**div**> </**div**> <**script src="script.js"**></**script**> </**body**> </**html**> |

### Screenshots





What if a **continent** has **no countries**? There should be nothing rendered in the **dropdown menu**, in that case.



You will also be given a **GIF** in the **resources** which shows you the whole process.

Your application will be checked thoroughly for mistakes and errors in the code. Make sure you implement everything correctly. Check your code at least **3 TIMES** before submitting it to ensure that everything is correct. Sometimes just because it looks correct, does not mean it is correct.