Interactive Map

1.1 Objective:

Create a website which shows the interactive map. When the user clicks on any country(marker), details of the country should be shown to the user.

1.2 Design and Implementation:

Userstory#1: Register with Google Cloud API and obtain the API key to call the Google Map API. Create a React application InteractiveMap.

Userstory#2: Create a react component should display the user google Map

Userstory#3: Set initial center (longitude and latitude), country, Zoom, height and width for the Map.

Userstory#4: Display markers(countries) on the map locations.

Userstory#5: When user click on any marker an Info window will be opened to display the country details.

Userstory#6: When user click on marker display loading image until the country details is fetched.

Userstory#7: When user click outside the info window it will get closed.

Userstory#8: Write automated test cases to test the functionality of the interactive map.

1.3 Technical Specification:

Using React Js library create an interactive web application to display the country details on clicking the marker on map.

1. Create a component 'InteractiveMap.js and the component should be included in the index.html <root/> element.
2. InteractiveMap.js component is stateful component
   InteractiveMap class have following states
   - *showingInfoWindow:* To store the state of the information window when location marker is clicked.
   - *showLoading:* To store the state of the Loading Spinner image inside the information window.
   - *activeMarker:* To store the active marker
   - *countryCode:* To store the selected country code
   - *responseData:* Object to store the Graphql API response

3. Interactive Map component have the following elements,

Sno	userstory	Element/Component	name	description
1	#2	Map		Display the google map
2	#4	Marker		Display the marker on specific country based on the longitude and latitude
3	#5	InfoWindow		Opens when the marker is clicked on the google map. Displays country details.
4	#6	ShowLoading	Functional component	Appears when Graphql API is called and disappears when data is fetched.
5	#5	ShowCountryDetails	Functional component	Displays the country information

4. Functions

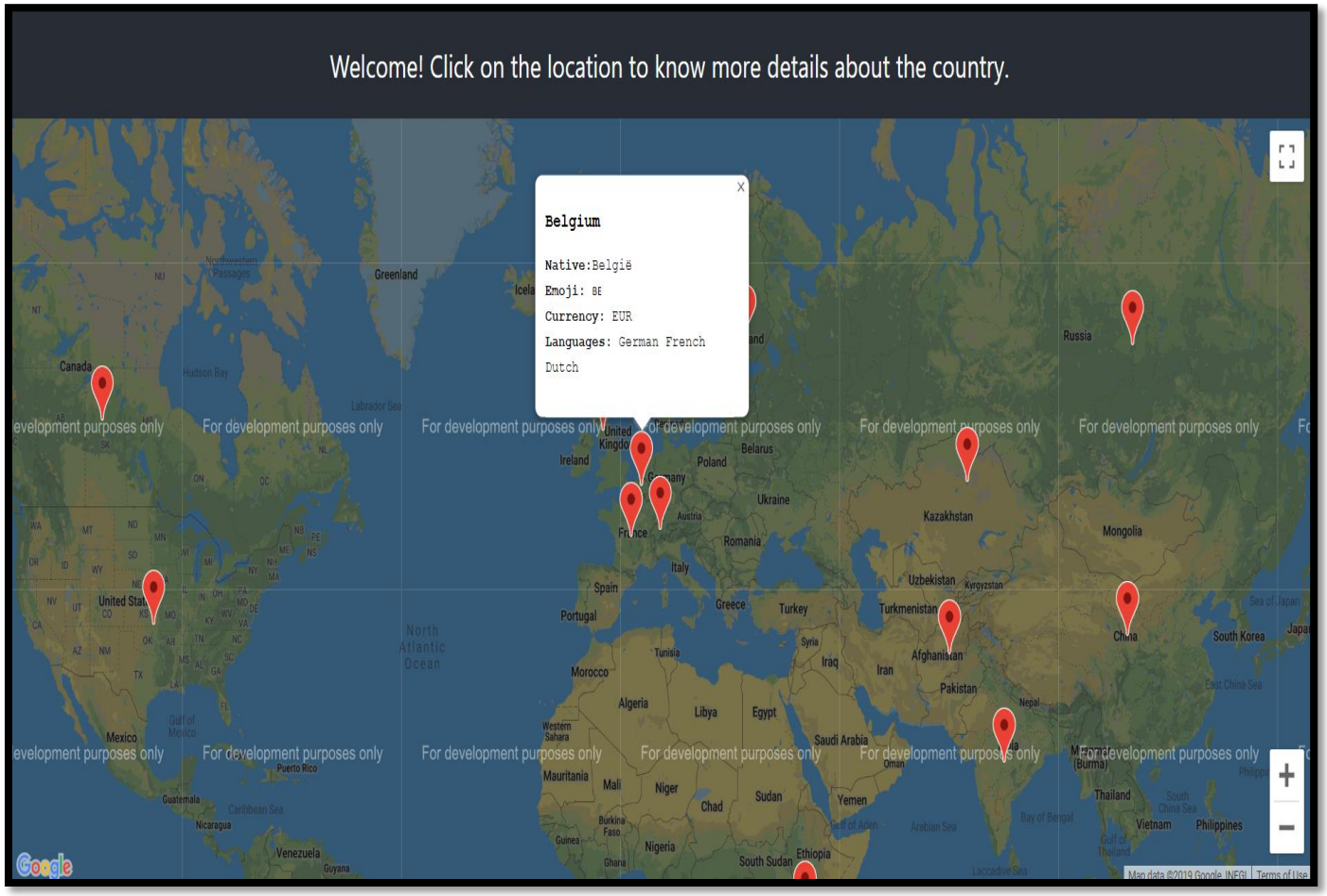
s.no	userstory	Function name	Event	descrption
1	#4	MarkLocations	Called when map is loaded.	• When map component is loaded, Marklocations() method is called to add the Marker to the map. • Loops through the locations array which contains country, longitude and latitude and forms the Marker.
2	#5	onMarkerClick	Marker.click()	• Triggers when marker is clicked. Displays the Information window.
3	#5	fetchCountryInformation	Marker.click()	• Method fetch the Country Information's. • Calls the Graphql API by posting the country code and retrieves the country information.
4	#7	onMapClick	Map.click()	• If information window is open it get closed when user clicks anywhere on the map.

5. **Userstory#8** :
   Automated test were written in the “cypress\integration\interactivemap_spec.js” file.

   Testcases:

s.no	userstory	description
1	#2	Test the map is loaded
2	#5	Test whether API called is returning the data.

1.4 Output



Questions:

1. How did you choose your technical and architectural choices used as part of your solution?

    - **Dev Stack** : *Visual Studio Code, eslint, Cypress.io, GIT*
    - **Application Core:** *state management, DataFlow, google maps react and GraphQL API.*
    - **Interface :** *Interactive map Component, interactive map.css*

    VisualStudiocode: It is a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

    Cypress.io : Helps to test the application in a browser mode. Helps to point the testcases to the particular dom by highlighting in the browser which is really helpful for users.

    GIT: To save the daily changes to repository branch and have the source code in network.

    State Management: Since it is a single page application with simple features the state were maintained with state variable in the interactive map component.

    Dataflow:

    - During the page load, the **interactivemap** component loads the google map and show marker for particular countries. Country locations is hardcoded in the page for specific countries.
    - For each marker attribute name is set with the country name. When marker is clicked onmarkerclick() function will be triggered.
    - API https://countries.trevorblades.com/ will be called with the country code and the fetched details will be displayed in the information window.
    - Information window closed when user clicks outside the window or the close button.

    google-map-react: is a component written over a small set of the Google Maps API. Simple to user and it allows to render the react component on google map.

    GraphQL API (https://countries.trevorblades.com/): This helps us to fetch the information by using query. We can customize the query as per the data we need which helps in improving the performace.

    Interactive map component : Class component which maintains the state of the application and renders the google map component. It also has functional child component Header, ShowLoading, ShowCountryDetails.

    Interactive map component.css: style for the interactivemap component is written.

2. Are there any improvements you could make to the final piece?
    - *I could not able to access the DOM elements loaded inside the GoogleMap API so was not able to write the testing based on the Marker components and its events.*
    - *I will try to write test cases for the markers inside the google map and marker click event.*
3. What would you do differently if you had more time?

Currently, the markers are shown from the hardcoded Locations array, this I would have changed to form Locations array dynamically by the below mentioned steps.

- *Using the following Graphql query would have tried to fetch the countries list before the map loads*

```
{
        countries {
                name
                code
        }
}
```

- *Fetch the longitude and latitude for each country by passing the country code to geolocation API. Store the response in Locations array. Hence the markers will be showed from the Locations array.*