

## Interactive European Data Visualization Project

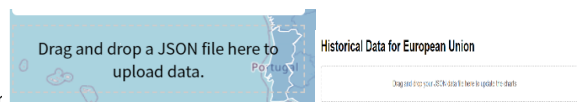
I declare that no generative AI tools or systems were used in the development of this assignment. All content, including text, data analysis, charts, and code, was created independently without any AI assistance.

### Motivation:

At the beginning of my project, I planned to have a financial markets retrieval system. However, free API cannot afford the frequent requests and limited stock index use (such as finhub and yahoo). Moreover, I wanted to plan a fundamental study of worldwide economic data. But there are no useful APIs. (For example, the IMF API was still inaccessible after I got the server. alpha vantage has data limitations and so on. Ultimately, I experiment with the APIs for EU banks to design an EU financial map. However, although I have had some success in the API, the EU banks limit my visit. I have to choose the World Bank API. It supports the five basic data, which are GDP (Gross Domestic Product), CPI (Consumer Price Index), Gini Coefficient and population. I think these numbers can reflect the financial aspect of a country. In the map area, I use the same leaflet, it is the same for the week 5 assignment. In the chart, I choose a line chart which can illustrate the trend directly. (Possible redundancy in the code due to my multiple iterations of the program )

### Functional guidance

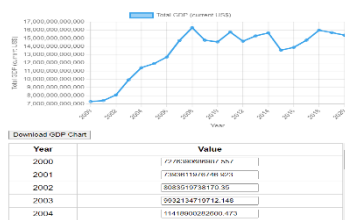
To prevent browser conflicts, go to the folder directory via CMD and then start the server using the node server.js command.



#### 1. Drag'n'drop new data to charts/map

This feature uses HTML and JavaScript. Users can drag files to a designated area on the web page, and JavaScript will process the file upload through event listeners and parse the data into JSON format. The processed data will cover the original data in Leaflet maps or line charts. The user can use Try1.js in the chart and Try2.js in the map as an example.

#### 2 and 3. The application show relevant data on a map and user has chance to change the data



By listening to input events and combining with Chart.js to update chart data in real time, users can click the edit button to modify the value in the table, using two-way binding mechanism to achieve dynamic visualization of data.

#### 4. User can switch between different layers of data on a map

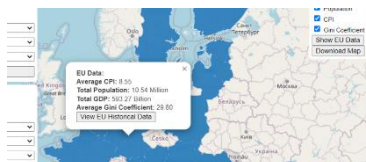
Users can select these options to change the data and color displayed by the layer. The method is to set the color and transparency of the map layer according to the relevant data of the country (such as GDP, etc.) through the `setStyle` method, so as to dynamically update the style of each country on the map.

## 5. By clicking the map user has an option to get to additional charts covering that area

The type of data displayed can still be selected by clicking on the bubble displayed by the map. If you click on View Historical Data, a twenty-year line graph of the four data is shown. The function method is same for other.

**6 and 7. There are more than two items of data available (e.g. elections data, employment rate and number of residents) – this means that there are three API calls made.** I use the slow loading method where the request is made to the api only when the user selects data and clicks on the map. If four are selected at the same time that is four requests to the api.

## 8.Data is combined and merged to generate new data, which is then visualized.



I developed a representation based on the overall EU data in the list, which can be visualized in the view in the bubble

## 9.Users can define what should be done to different data items (e.g. values are added, multiplied together etc. before visualization)

I developed two functions: the first one compares selected data from two countries and displays the results in a bubble on the map; the second one performs a quadratic operation on two data items for a single country, and the results of the calculation are also displayed through a bubble visualization. Both fetch the data through an API and display it instantly after the comparison or operation is performed.

## 10.Able to download the visualization as a PNG (or SVG) image

The download of visual content is achieved by calling the `toBase64Image()` method, which converts the map or chart to PNG format and triggers the download function.