Name: - Prakash Anilkumar Rai
HWU id: H00423994
Topic: Total population and GDP per capita

https://youtu.be/sFVwprLFWlo

Course code and name:	**F21DV - Data Visualisation & Analytics coursework 2**
Type of assessment:	**Individual**
Coursework Title:	**Total population and GDP per capita**
Student Name:	Prakash Rai
Student ID Number:	H00423994

Declaration of authorship. By signing this form:

- ☐ **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.

- ☐ I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.

- ☐ I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

Student Signature *(type your name):* *Prakash Rai*

Date: *20/04/2023*

Copy this page and insert it into your coursework file in front of your title page.
For group assessment each group member must sign a separate form and all forms must be included with the group submission.

Topic:-Total population and GDP Per Capita

Our scenario

we have used 3 different datasets to represent and visualize the life expectancy and GDP per capita. The csv file contains the countries, GDP and life expectancy.
I used promise all () to connect all the datasets and visualize using charts according to the data.

Design Planning

Data Preparation

Preparing the Data Is the First Step in Creating Graphs with D3.js The first step in creating graphs with D3.js is to prepare the data. CSV files, which stand for "Comma Separated Values," are a typical type of data format that are utilized for the storage of tabular data. It will be necessary for you to read and parse the CSV data by utilizing the built-in methods of D3.js such as d3.csv () or d3.dsv () in order to convert the data into a format that can be easily utilized for the creation of visualizations.

DOM Selection and Creation

The next step in the rendering process involves selecting and creating DOM elements for the document object model (DOM). The Document Object Model (DOM) is utilized by D3.js in order to modify HTML components. You can choose existing HTML elements or build new ones for the graph by making use of the d3. select () or d3.selectAll() functions that are available in D3.js. Connecting the Data to the DOM Elements After you have both the data and the DOM elements, the next step is to tie the data to the DOM elements. You can bind data to DOM elements based on a key or index using the methods that are provided by D3.js. Some examples of these methods include data () and enter (). During this step, a link is established between the data and the DOM elements. This connection ensures that any changes made to the data will be reflected in the graph.

Creating the Graph

Now that the data have been connected to the DOM elements, you are able to actually create the graph. D3.js offers a wide variety of methods for the creation of various kinds of visualizations. These methods include d3.scale(), which is used for the creation of scales that map data to visual attributes; d3.axis(), which is used for the creation of axes; and various d3.shape functions, which are used for the creation of various kinds of graphical elements, such as lines, bars, and circles. After you have generated the fundamental graph, you will then be able to add styles and make adjustments to it in order to make it more informative and visually beautiful. D3.js has significant support for styling and customization, including the ability to specify attributes, apply CSS classes, and use transitions to create animations that are fluid. You can also add interactivity to the graph by making use of the event handling functions that are provided by D3.js. Examples of this include hover effects and click interactions.

Data Updates and Transitions

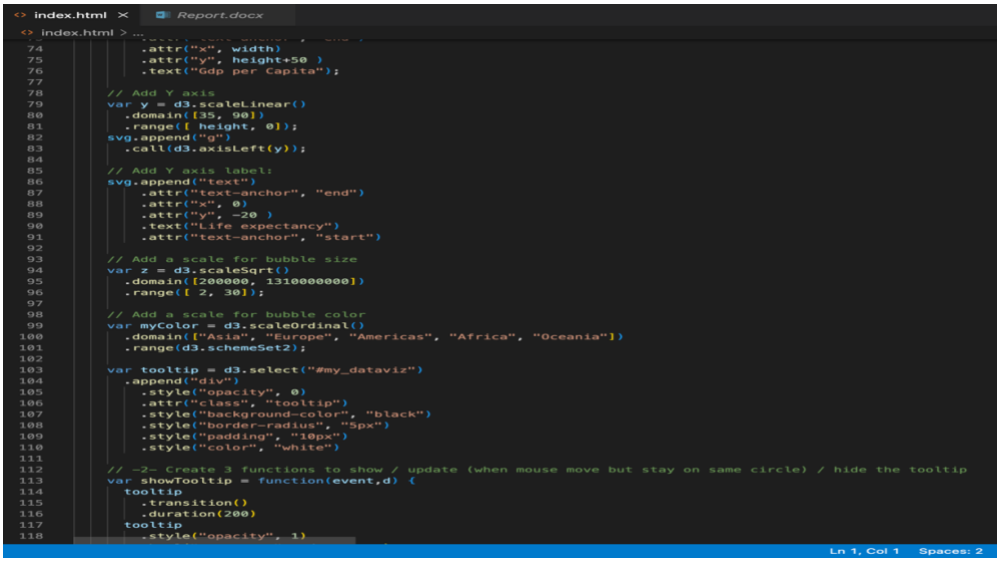
One of the powerful characteristics of D3.js is its ability to manage data updates and transitions in a smooth manner. This capacity is one of its key selling points. The graph can be dynamically updated in response to changes in the data through the use of the data manipulation and transition capabilities

provided by D3.js. This enables you to construct dynamic and interactive visualizations that are capable of reacting in real time to changes that occur in the data that they are based on.

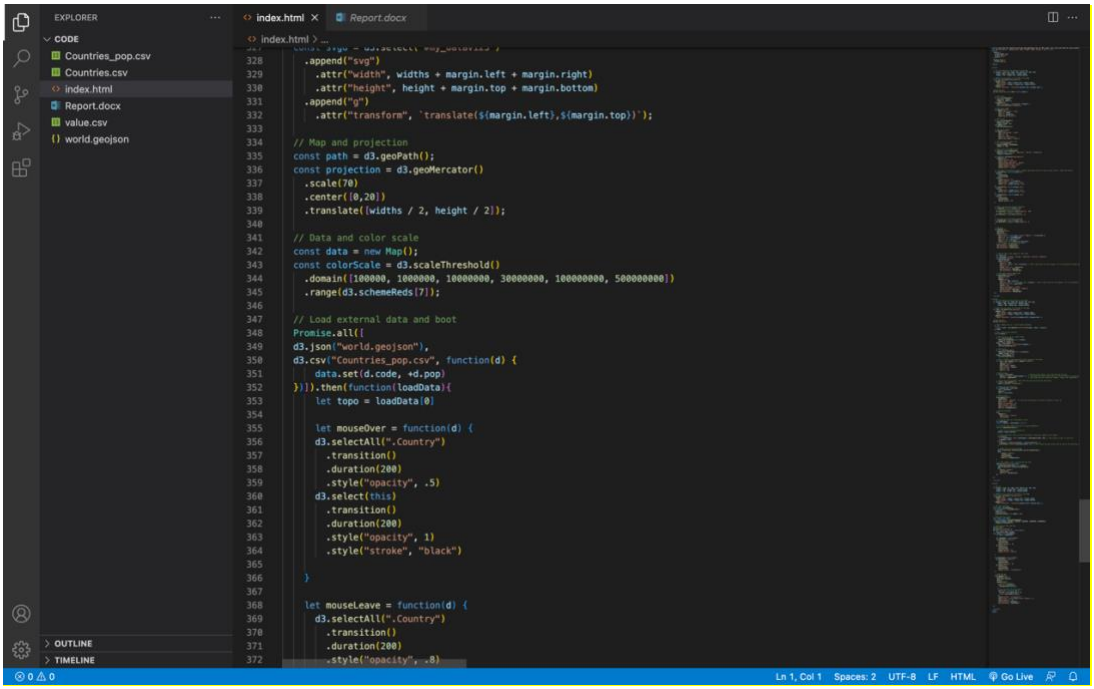
Testing and Refining

As a last step, it is essential to carefully test and improve the graph in order to guarantee that it is accurate, responsive, and performs effectively. Identifying and resolving any problems can be accomplished with the help of the built-in debugging and error handling functions of D3.js. In addition, the design of the graph can be improved further by soliciting comments and suggestions from users and then making any necessary adjustments in response to that feedback.

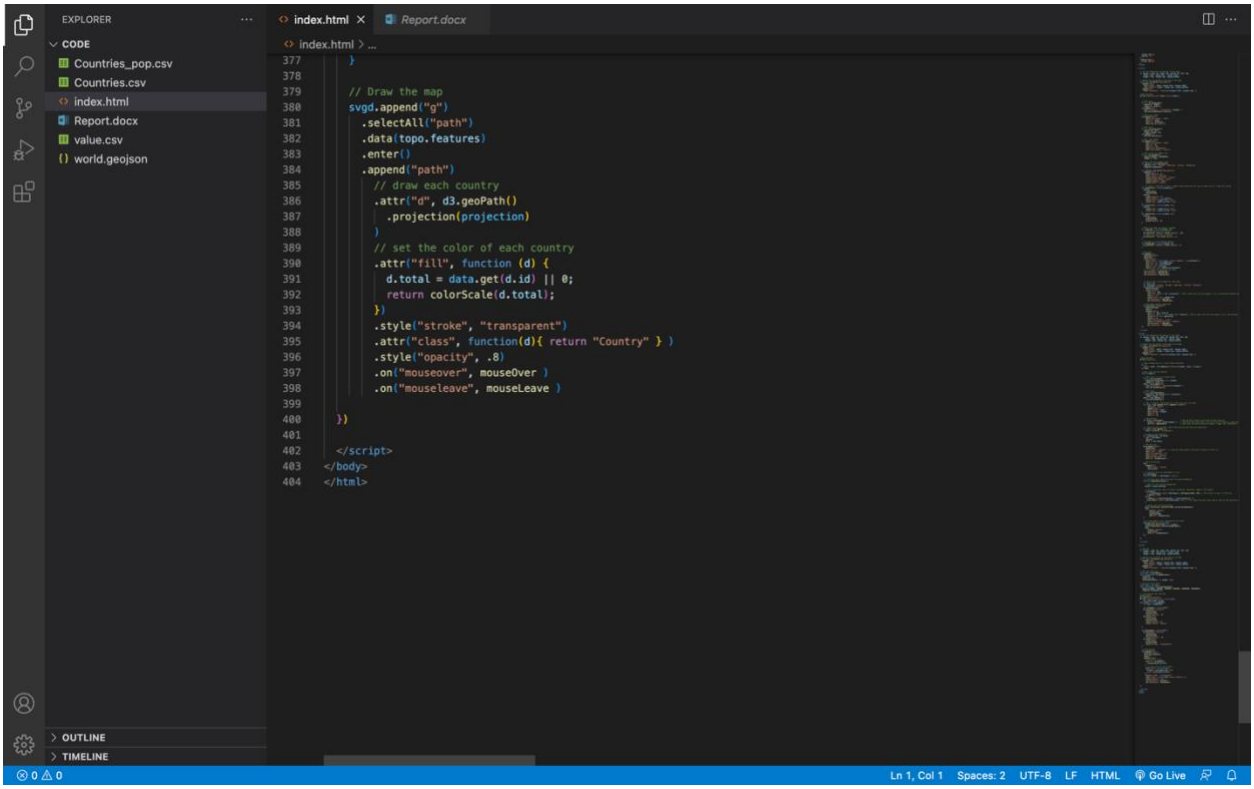
Screenshots and explanation



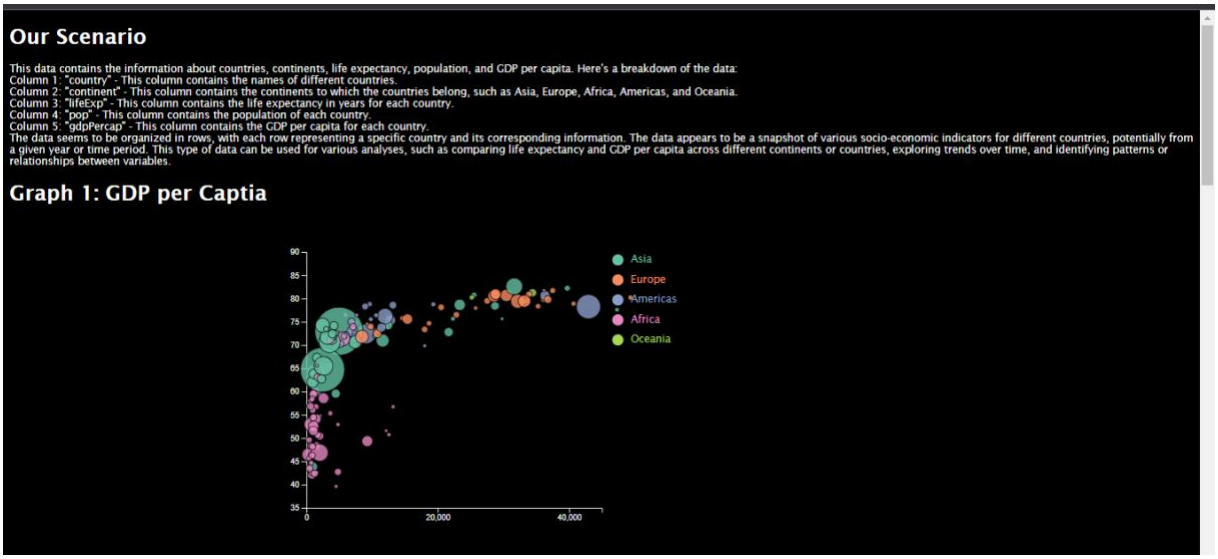
Here we have added a scale for bubble size and color used in first chart.



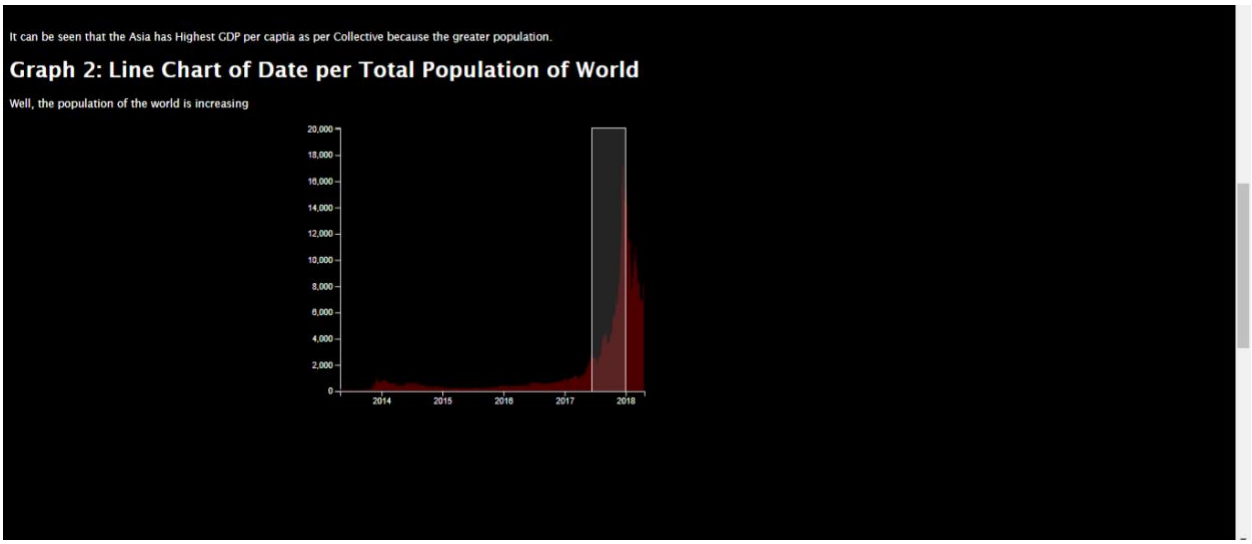
This is screenshot of a code where I have loaded the map and projection and I have used Promise.all() for loading multiple dataset.



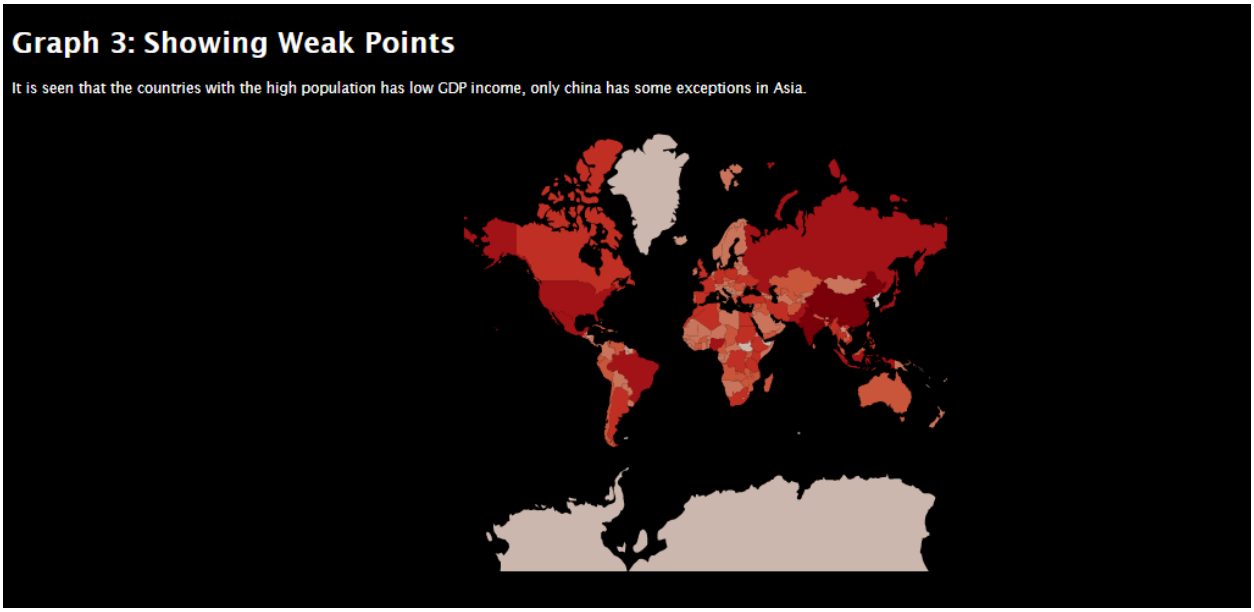
This is the part of screenshot where we draw the map



This graph is basically showing the bubble chart of the GDP per capita income. The density and the size of the bubble shows the size of the values. In USA there is highest GDP per capita. It has also multiple factors that could be impacted on the GDP per capita



This line chart is showing the population of the world. It can be seen that in the 2017 to 2018, the population kept on rising and then due to Covid outbreak, it decreases. The interactivity is applied on this graph which offers the user to zoom into the graph as much he wants. The user can convert the years into the months and days.



This graph is showing the population. The dark area means more population and light area means less population. As in the First of the graph, it was showing that the population has link with the GDP per capita. But it is wrong the China and India has one of the densest populations in the world, but its GDP is good as compared to the low population countries.

Conclusion

Through this complete visualization we can understand that GDP and population are inter-connected with each other, as the population of country increases the GDP of that country decreases, but it's not true for some countries like India and China with good governance or plans.