# **Final Project – A Visualization of Japan’s Population Plateau and Decreasing Birth Rate, and USA’s Population Growth**

# 

**Jefferey Lin**

**Antoine John**

**Bhaskhar Verma**

**Professor Amir Masoumi**

**Data Visualization CIS568**

**Spring 2024**

# **Overview**

At the beginning of the semester when initially thinking about the project, I had a broad idea of what I wanted to display, but obviously did not have the tools and skillset to visualize anything. I come from an Econ undergrad and had never learned any languages other than Stata, so HTML, JS, CSS, was all new to me. I knew of the birth rate issue from my time studying and living in East Asia. I retrieved some data sets that varied in data, some were about population, others about birth rate and exports/imports.

When scouring kaggle for another course, I stumbled across a Kaggle dataset that had 40 features that depicted Japan's population, birth rate, age, and others. I then thought to just focus on telling a story based just on the stagnant population and declining birth rate rather than incorporating many other aspects. I wanted to simply visualize for the viewer how society in Japan has placed less importance on raising children. I would end up doing this by having a time series chart that depicts the decreasing number of births per year, the population plateauing, the average number of children decreasing and the average age of the parents, when the kids are born, increased. I ended up with 5 html files, three that use my data set, Japan\_birth.csv to create different visualized charts, one to be the main page that links all the charts together, and the final html file to be a choice between USA and Japan. My partners added two of their own HTML files that would be a USA map and line chart.

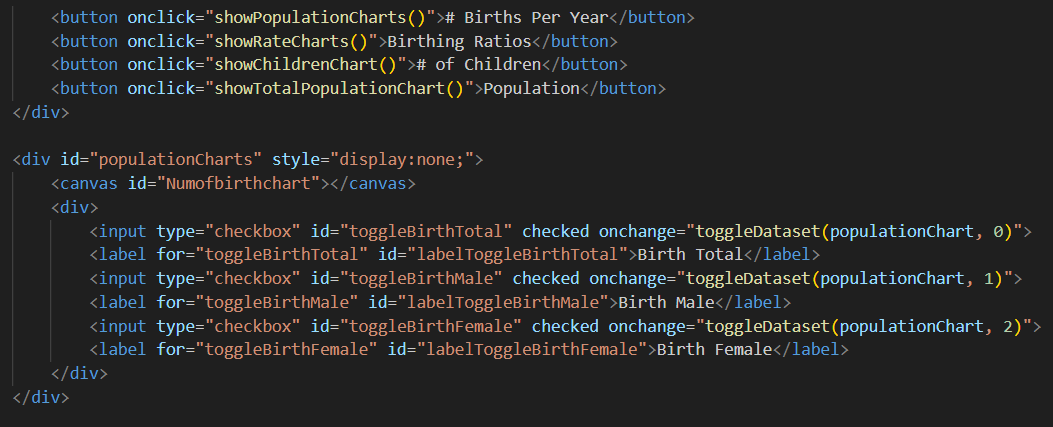
All the html files and jpegs/images are in the single folder. The project was done in visual studio code and was visualized in a live server. The user only needs to go live with the MainPage.html file to be able to see and interact with all 7 html contents.

# **Project Framework**

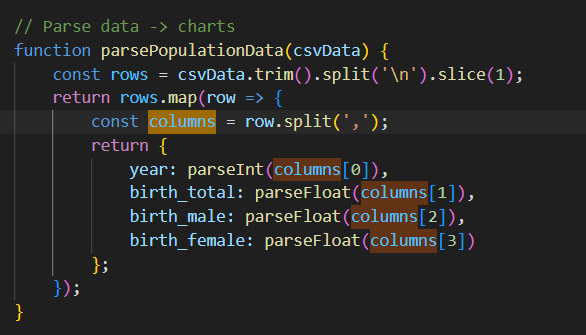
I first looked at which of the columns in the excel file I would want to use to visualize the idea of Japan’s birth rate being a major problem and its societal impact. I then created two html files that would be time series charts, using chart.js that would show some of the data from these columns. Afterwards, I thought about D3.JS, since we've been doing assignments with this for the semester, and looked up different chart styles they had on their website and my eyes targeted the bar chart race, which would be then my third html file. Then, I created a new file that would be the main page for the Japanese section of our project, and would have buttons linked to open each of three html charts. I then created another new file that would be the main page for the project when my partners sent their files the day before the due date.

HTML 1: Population.html

In this HTML document, I created a dynamic web page that visualizes time series data related to birth rates and trends in Japan. It uses CHart.js library to generate the interactive line charts based on the data from Japan\_birth.csv. After adding the initial HTML structure, the body section started with division elements to organize charts and provide toggling functionality for the charts. There are four charts within this one file, births per year, birthing ratios, number of children and population. In the births per year chart, there are three lines representing data from three columns. To add interactivity in the chart, the user can toggle which lines to see. A snippet of the code is shown below.

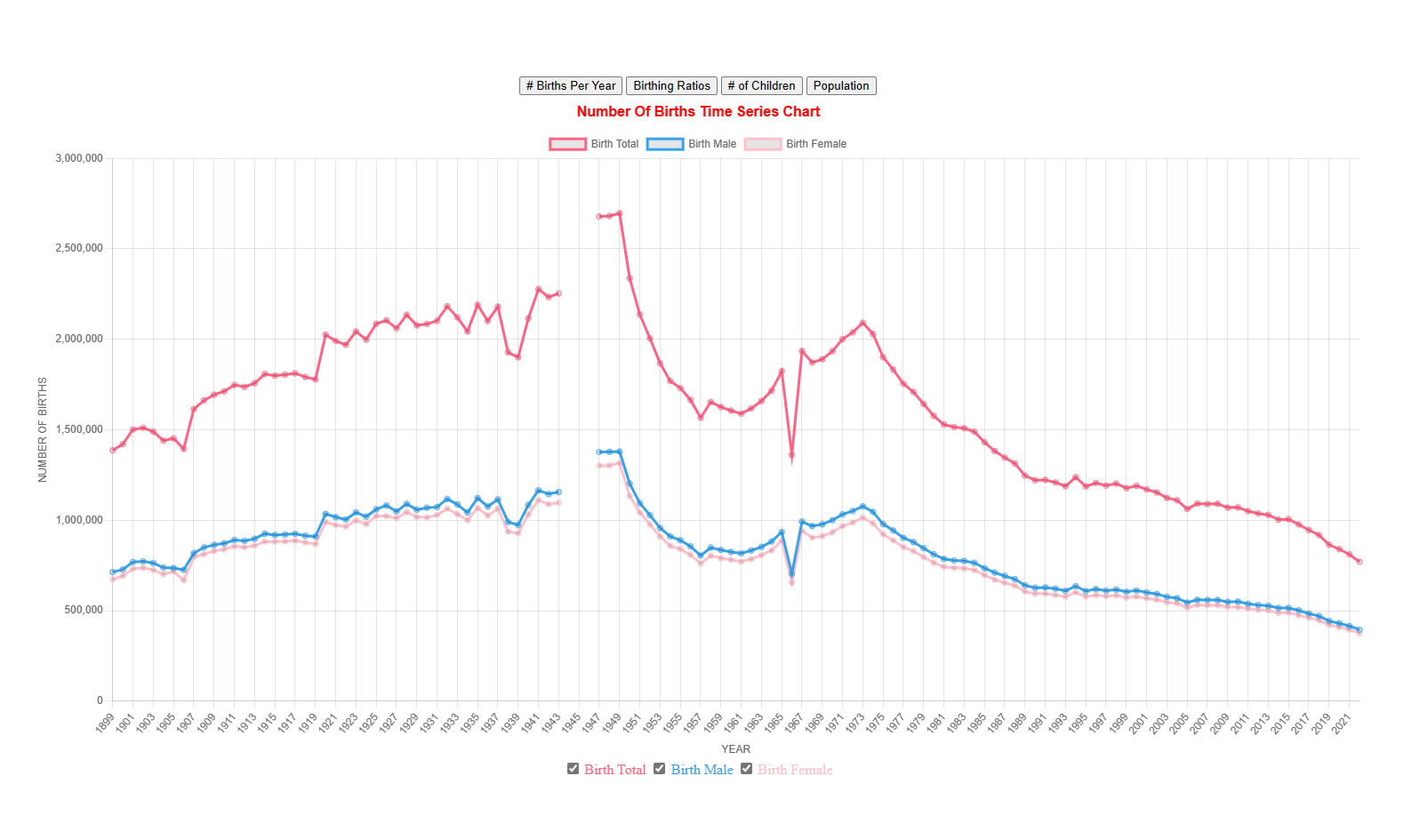


The JS code embedded in the document then fetches the data from the csv file and parses the data into objects presenting the different aspects of birth and population statistics in Japan over time.



After parsing the data, using chart.js, it created line charts for each category of data and configured options for chart appearance and interactivity. The toggle visibility in each of the charts is implemented through checkboxes, allowing whoever is viewing the chart to customize their viewing experience. The script also makes sure that the color of the labels match the corresponding chart lines. 

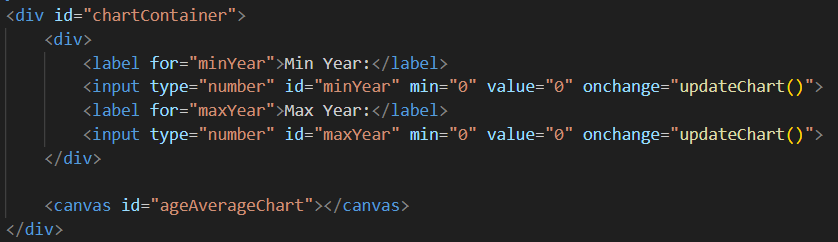
When running the live server on visual code, I ended up with this interactive chart, and the toggle options.



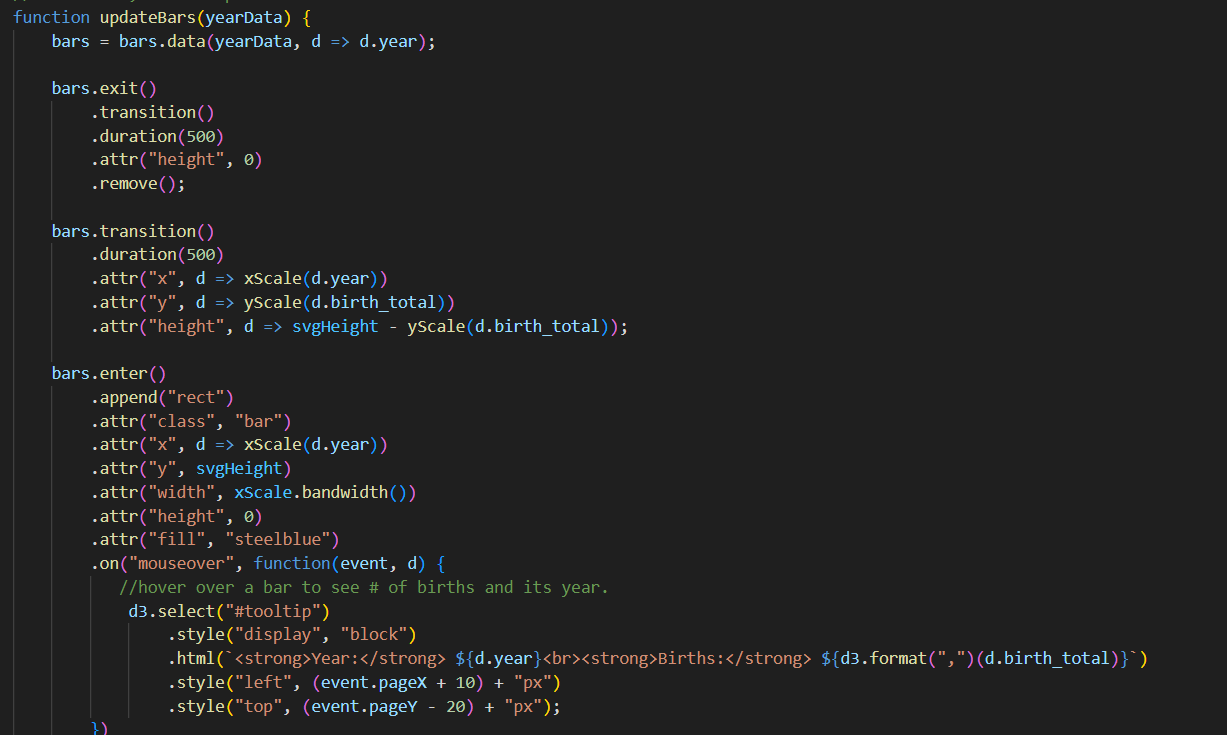
## HTML 2: ParentsAge.html

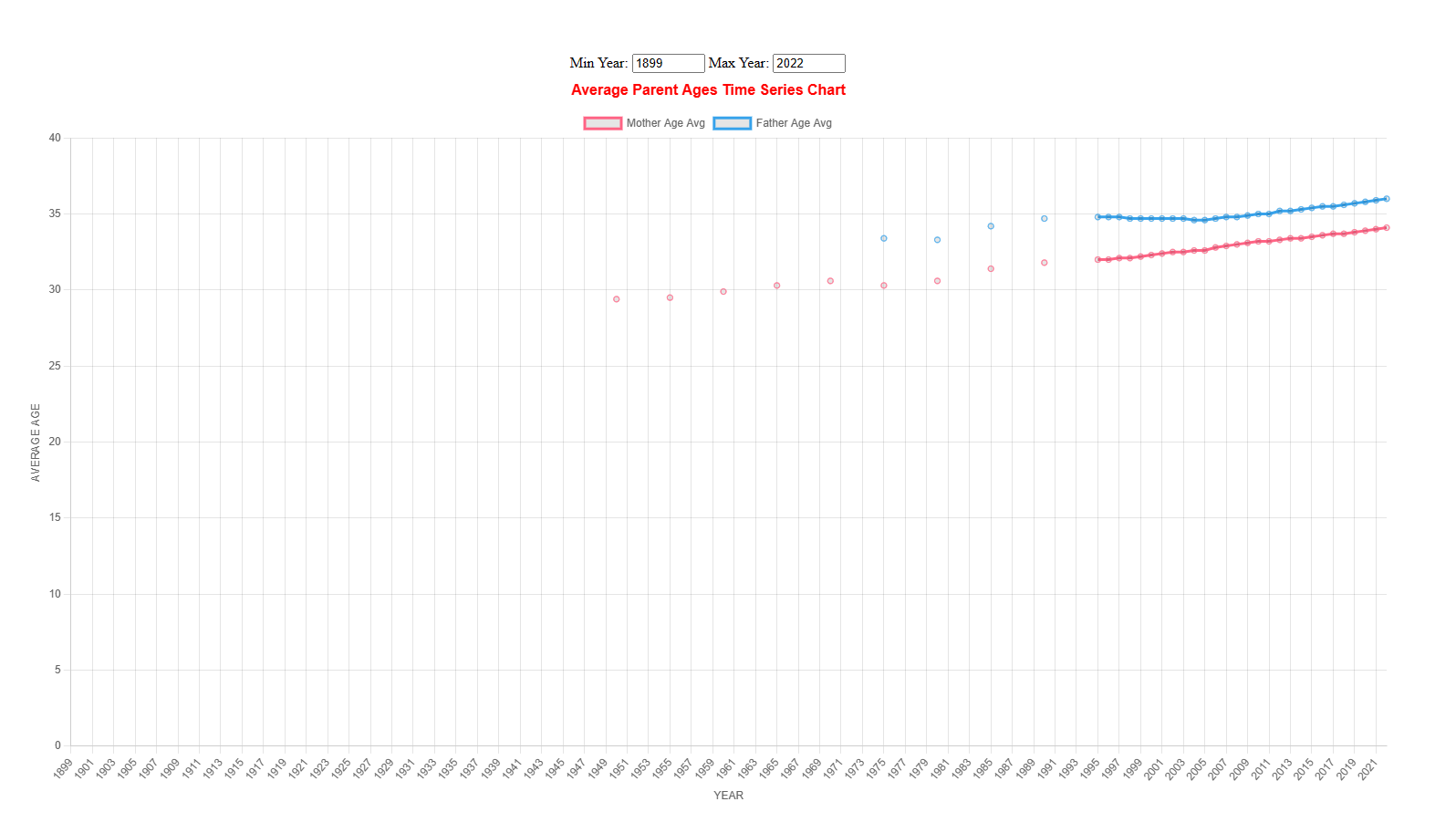
In this HTML document, I created a dynamic web page that visualizes the average ages of the parents over time in Japan. Similar to HTML1, it also uses Chart.js to generate an interactive line chart based on Japan\_Birth.csv’s data. The document, again, starts with a standard HTML structure and imports the Chart.js library.

The body section contains a chartContainer to contain the chart and input elements for specifying the range of years to display. This chart is still a time series chart, however, it differs from the first html file’s chart by changing the interactivity. The data set’s year ranges from 1899 to 2022, in this chart, the user is able to change the years and the chart will be updated.



After fetching the data, parsing the data and creating the chart by initializing the chart.js instance, I then created an update chart function that will update the chart’s data based on the years the user inputted. Finally, I added an event listener to resize the chart dynamically depending on the window size as the monitor I use and my computer’s screen is very different.



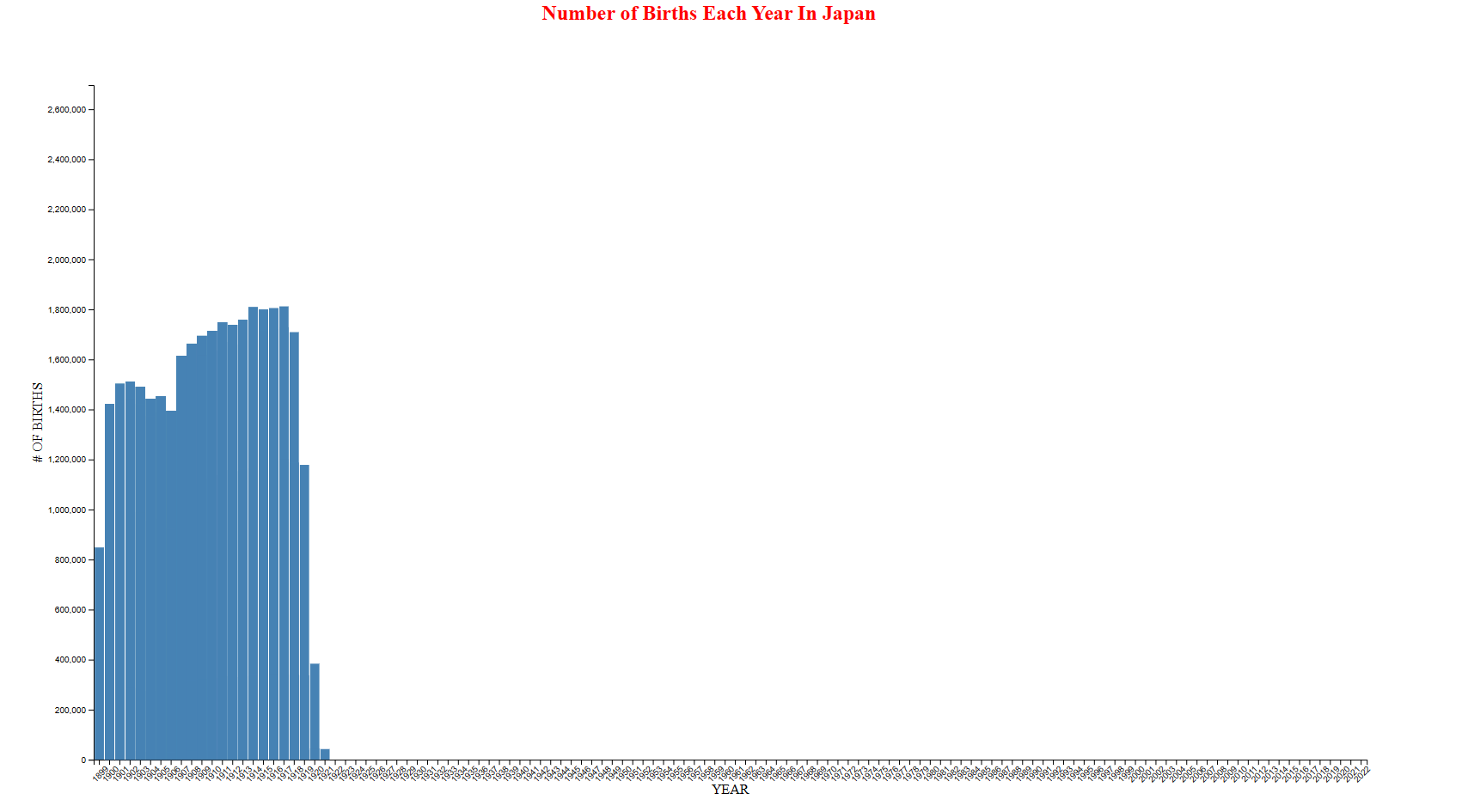


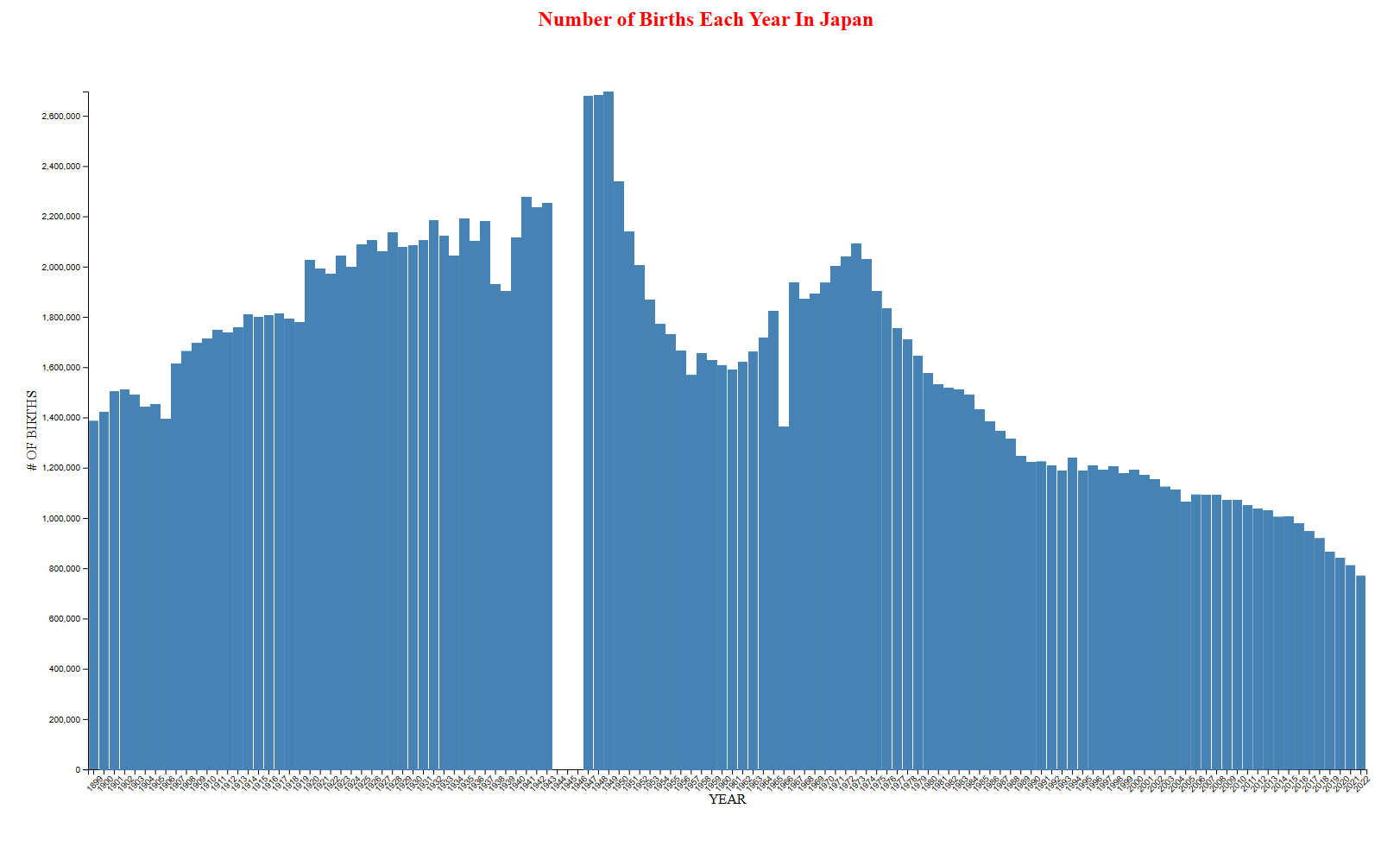
## HTML 3: Barchartrace.html

The final chart I created was a bar chart race to visualize the number of births each year in Japan. This chart utilizes D3.JS for its visualization. The chart dynamically updates to display the births over the time frame of the data set, 1899 to 2022. The tool tip feature provides additional information when hovering the individual bars, showing the year and number of births. It also automatically adjusts based on the window dimensions.



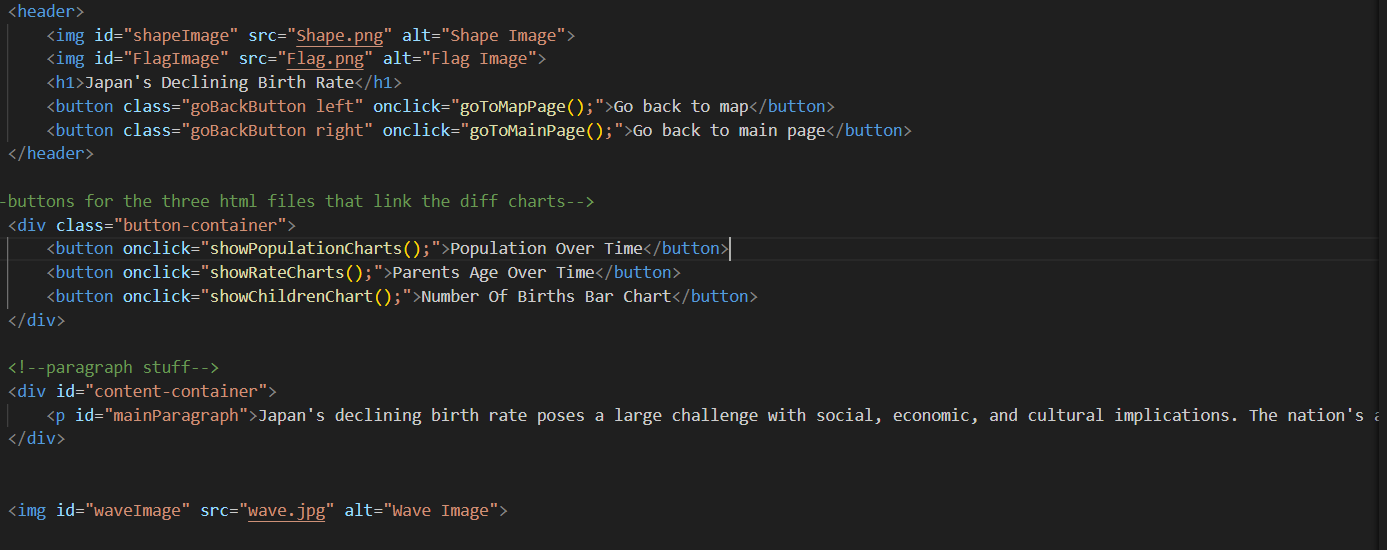




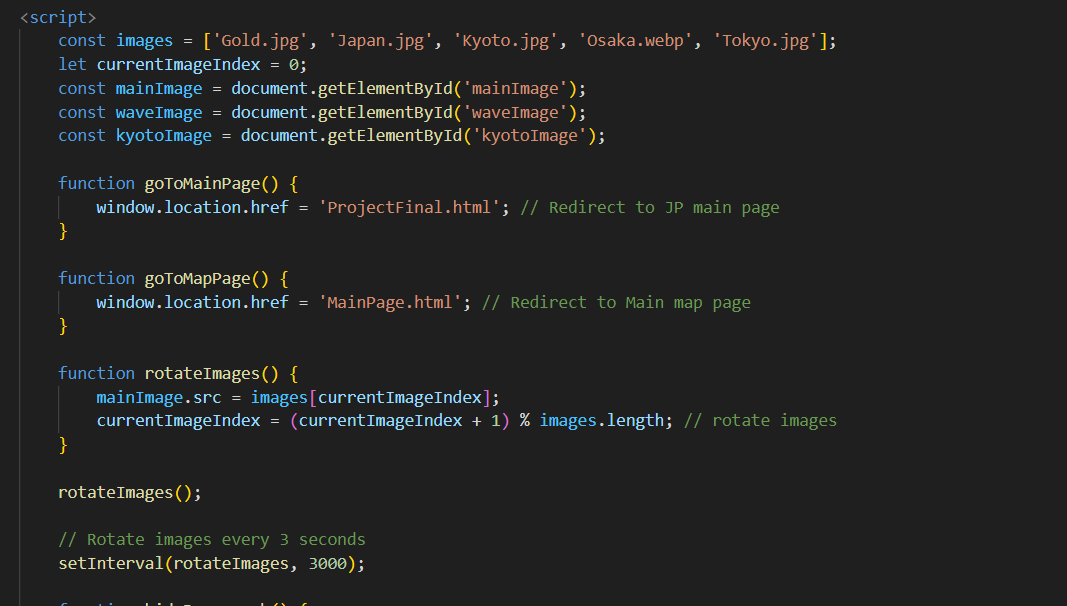


## Main Japanese Page HTML: ProjectFinal.html

The final HTML for the Japanese section of the project presents information about Japan’s declining birth rate by using a paragraph and a container, accompanied by visual elements and nivation buttons that are linked to the three previous HTML chart files. Also included are two buttons, go back to main page, or back to map page, which will automatically go to the start of this html file or to the Mainpage.html map where you are able to choose USA or Japan’s section.



Throughout the main page, there are a number of fixed pictures of Japan and also a main rotating image, these are just aesthetics for the page, for visual appeal and context to the content.

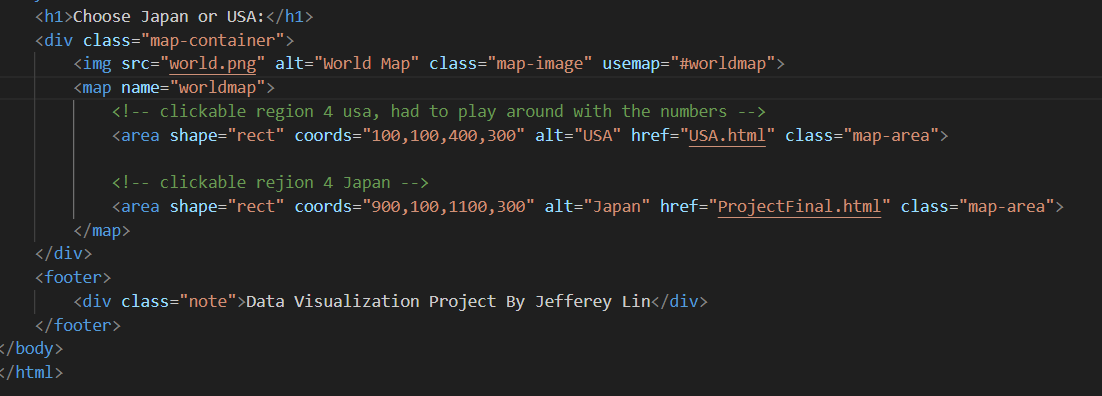


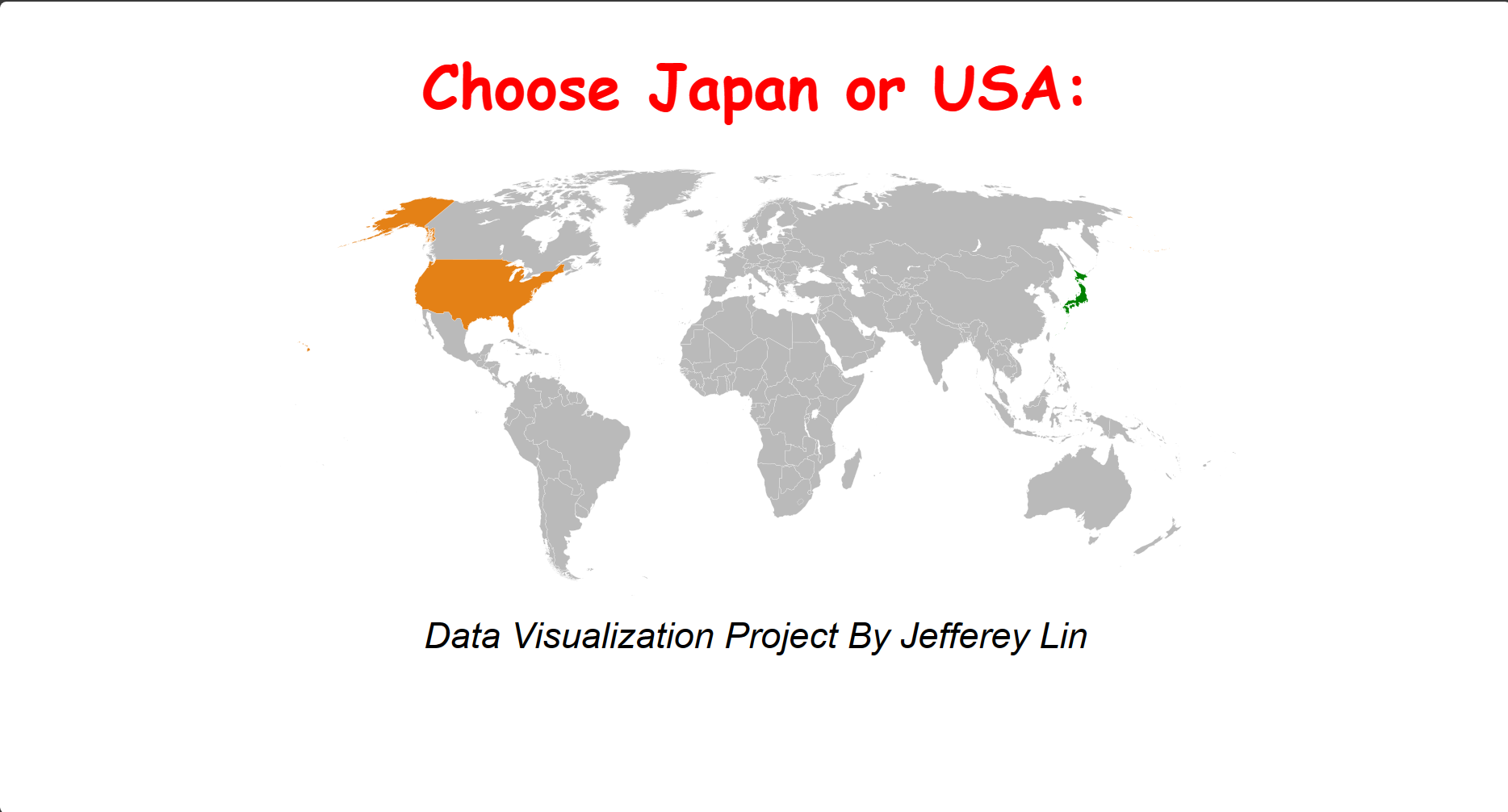
Finally iframe elements are used to display different charts and block different elements depending on which button is clicked, and it will load the corresponding chart HTML file in the iframe.



## Main Project Page HTML: MainPage.html

Lastly, this HTML document represents a world map with clickable areas for Japan and the USA, allowing users to choose between them. I had to implement rectangle coordinates for just the area of Japan and USA to be clickable, the rest of the world is not. When clicking Japan, you are then linked to the Japanese main page.

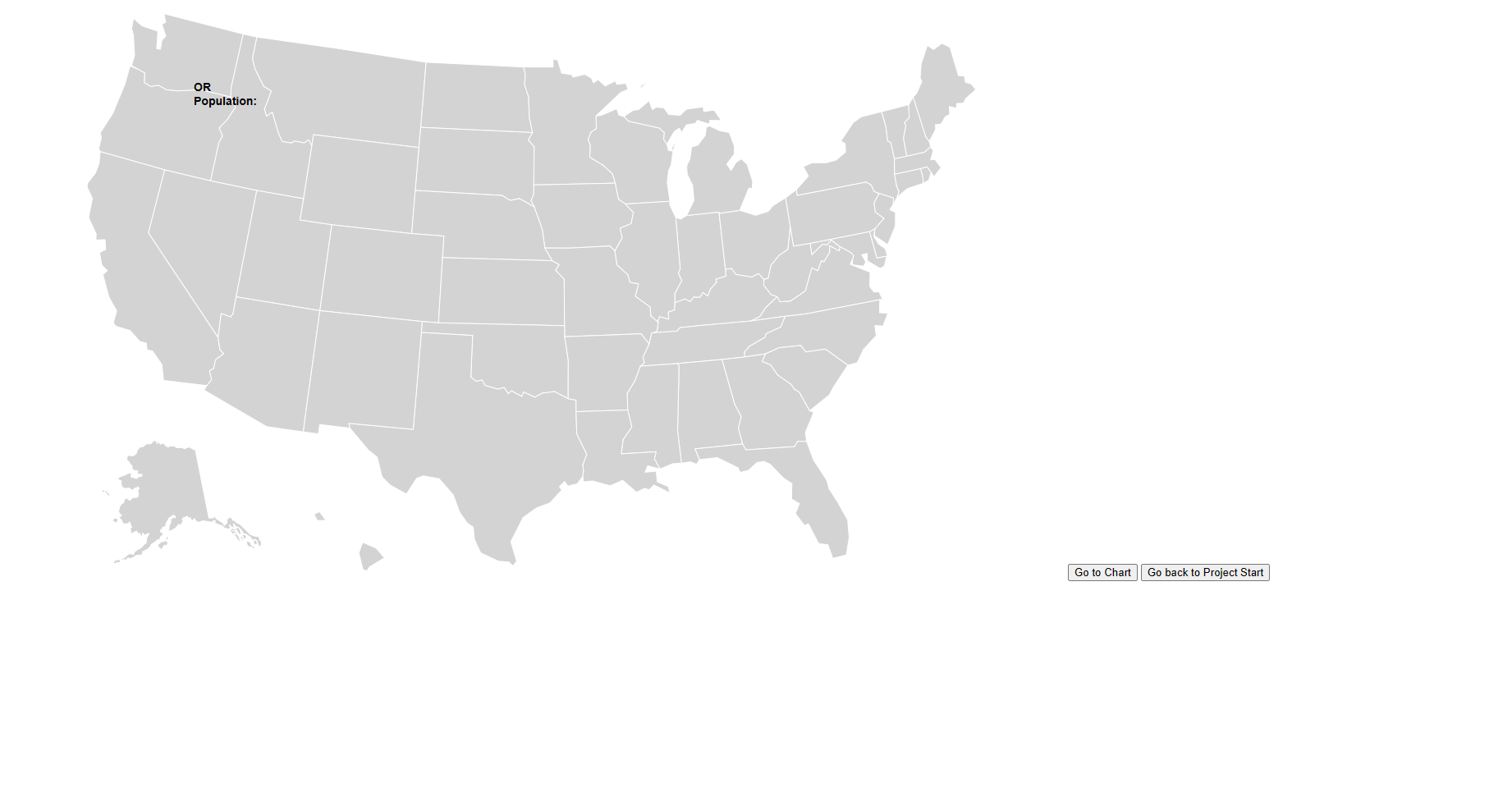




## USA MAP HTML

My partners used d3js for both of their html files. The first file was to make a USA map, originally, they were trying to make it so that the population would show up when hovering over one of the states, however, this didn't work and so they had to create a chart. I added a go back to main page and go to chart button so the whole project was connected.





## USA CHART HTML

The second html file they created was a time series line chart that showed the population over time for each of the states. The interactivity that they created was that you could see the population of the state corresponding to the line when you hover over that line. However, it only showed you the 2022 population, hovering over different years of the line did not change the population shown, so I then quickly added a table that starts of in descending order of population, and the user is able to select the year, though the rest of the years don't show the descending order in terms of population. I also added a go back to us map button.

