My interactive website is a webpage that provides basic information over the first 9 Pokémon starters in the first three generations of the games. I used a combination of HTML, CSS, and JavaScript to provide a clean look and more interactive options for the user to experience, and to learn about some of the starter Pokémon.

For the HTML section of my website, I had 10 separate files. There is one main file, titled ‘CSC 496 Project1.html’ that should be launched when a user wishes to view the website. This main page contains a title, introductory information, 9 buttons, and a dropdown menu. The other 9 files are of the individual starters. These files contain basic information about each starter Pokémon, along with 3 buttons, 2 of which navigate to the other starters of the same generation, and the Pikachu button, present in all 9 files, which navigates back to the main file. These 9 files are structured very similarly to the main page to keep a consistent “feel” throughout the entire website. Each part of the files were sectioned into differently named ‘div’ classes, which allowed for easier CSS or JavaScript customization on any part of the page, as well as sectioning the overall code into more readable and smaller sections. Each button had a unique ID for CSS customization and used ‘href’ to hyperlink to the other files. These buttons used the ‘a’ tag, which allowed the use of ‘href’ as well as the option to use HTML text, which I chose not to. There is a sprite on every file not enclosed by a box. This sprite uses the ‘a’ tag to hyperlink to a website called ‘pokemonDB’, which provides extra, in-depth information about the sprite shown. Also included in every file is a ‘link’ tag that references the master CSS sheet named ‘CSC 496 Project1 Style.css’ where all CSS is handled for the website. On the main page, the dropdown menu prompts the user to select a Pokémon generation and activates a JavaScript function upon clicking the button next to the dropdown box. A footer is included that mentions where the sprite images were gathered from. Overall, the HTML section of my website takes up most of the code, and the improvements I can see myself making are eliminating the need to have 9 near identical pages, instead having a JavaScript script that automatically fills in information based on the specific button clicked, putting all of the starters onto the page, and generally having a more user-friendly wording or more in-depth explanations of the starters.

For JavaScript, I have a file named ‘script.js’ that has all JavaScript functions that I included into the website. The function stored on the file is named ‘show()’ which directly correlates to the dropdown menu on the main HTML file. This function activates upon the button with the text ‘Show the best Starter for the Generation’ and checks which dropdown box option was selected and uses a var ‘drop’ to gather the element from the dropdown box. The program then loads the value from ‘drop’ into a var ‘selected’. The program then activates a switch statement, and loads text into a var ‘output’, and then displays an alert with the specific text. The text contained in the file gives the ‘best’ starter choice for each generation, with brief reasoning on why the starter is the best choice comparatively. I believe that I could have more JavaScript integration into my program, such as basic animations, more interactive features with each Pokémon, such as listing abilities and moves, or alternatively having a function that gathers 2 separate types and gives the type effectiveness chart associated with them. There could also be functions to play the Pokémon cries upon clicking their specific button or a general Pokémon cry button underneath their buttons.

For CSS, I fully customized all pages with different text alignment, colors, and styles. All CSS was coded in one file, named ‘CSC 496 Project1 Style.css’ to make the HTML files more legible and to reduce the overall lines of code, since repeat information is used. For all pages, there is a gray background. Each button is bordered and contains the image of the Pokémon that is hyperlinked, as well as Grass types having a green button color, Water types having a blue button, and Fire types having a red button. The home file Pikachu button is also bordered and has a yellow button. Each button has a margin to separate them and make the website have a ‘cleaner’ look. Each button has its own section in CSS, mostly identical except for the image shown and the color of the buttons, such as having an identical unique size. All buttons and text, except for the footer, are centered to make the page more unique and to provide a cleaner look on the website. Every button also has a horizontal margin to space them out more and reduce the chance of the user clicking the wrong button. The image in each button is linked from a website to eliminate the need of sending photos along with the program. The title of each file contains a border to make the title stand out more compared to the rest of the page. The information on every page, such as instructions or basic Pokémon information, is bolded and separately sized to differentiate the sections. Every section has a consistent margin to make the website more consistent. Some areas that could be improved in CSS are more vibrant and interactive colors, such as a color change upon button clicking, better text orientation, and overall, more fine-tuned elements to make small differences in how the website is structured and presented for user enjoyment.

Overall, the scope of this project was to create an interactive website that had features from HTML, CSS, and JavaScript. The page provides basic information about the 9 starter Pokémon from the first 3 generations using HTML, as well as providing basic JavaScript functions to provide extra information to the user, along with CSS customization to make the page unique, interactive, and uniform throughout all files. There were several areas mentioned that could be improved to provide a more unique and informative experience for the user, although the product accomplishes the scope of the tasks assigned to it.