1. In the Navigation Bar css code, the nav class is given the “fixed” position so that despite the scrolling down on the page the Navigation Bar will stay in its place.
2. In the Navigation Bar css code, there was added an “.nav.active” styling rules to change its background color as well as its box-shadow when active is applied to the nav class. The same was done to the “.nav.active.container” of the nav class where the padding was changed from 20px to 10px when the active class is applied to it.
3. In the Navigation Bar javascript code, the nav class selected with a “querySelector” and given a constant “nav” and the window had an “addEventListener” to monitor scrolling to know when to run the “fixNav” function. When that function ran it checked how far down the Y axis the user went. When it reached a far enough point down the page it added the active class to the nav class thereby implementing the styling of the “.nav.active.” and “.nav.active.container” changing the appearance of the nav bar.
4. In the Followers html code, the body had div boxes with class of “counter” and an attribute of “data target”. This was later used in the Followers js code when the constant “counters” was created and the “querySelectorAll” was used to create an array containing all the items of the counter class.
5. In the Followers js code, the “forEach” method was used on counter to set the inner text of the html elements with the counter class to 0. A const “target” and const “increment” were created to get the “data-target” attribute of these classes as well as create an increment by which the target was divided by 200. (Notice the + sign which converts the string into a numerical type).
6. The function updateCounter was used creating a constant “c” which was the inner text of the counter class. An if statement was run which tested if “c” was less than the “target”. If so, the inner text was changed to the “c” value added to the “ increment” value. The “setTimeout” function was used to wait 1 millisecond before running the “updateCounter” function again until the target was reached which would then run the else statement making the inner text the “target” value.
7. In the Picturedisplay html there was a class of “slide” added to each div box which also contained a background image. Two buttons were created one with the id of “left” and the other with the id of “right”. From these classes and ids three constants were created in the javascript code: a constant of “slides” creating an array of all the slides, a “leftBtn” which selected the left button, and a “rightBtn” which selected the right button. (A “body” constant was created to get the body of the html document.)
8. A variable was created called “activeSlide” which was used to index the slide array. The function “setBgToBody” was made which set the background image of the body based on the background image of the “activeSlide” . Another function was created “setActiveSlide” that removed the “active” class from each slide class and added an active class to the slide that was at the index of the numerical value of “activeSlide”.
9. The “activeSlide” variable was incremented based on the click of the user as two “addEventListeners” were added: one for the right button which incremented the “activeSlide” value and one for the left button which decremented the “activeSlide” value. The two functions of “setBgtoBody and “setActiveslide” were called for both of these clicks.
10. To make sure that once the user went past the end or beginning of the number of slides that were in the html code (ie. that the numerical value of “activeSlide” would be larger than the number of items within the “slides“ array) the webpage would go back to the first slide of last slide of the array the right button and left button functions were given an if statement. For the right button it set the “activeSlide” variable to 0 if “activeSlides” went over the number of slides and for the left button it set it to slides.length-1 or the last index in the array when “activeSlides” went to less than the number of slides (ie. to less than 0).