GOALS AND PURPOSE

Upon learning of the freedom of subject matter this project offers, my mind immediately went to the roman emperors. I chose this subject for numerous reasons, my interest in ancient history, previous use as the subject of project some weeks ago, and the object-oriented nature of human profiles.

Within the scope of this project, I hoped to create a spiritual successor to the previous project. That is to say, I wanted to create a program that would allow for the viewing, categorizing and filtering of a number of roman emperors.

To accomplish this, it was my plan to create a central, scrollable area that contained a series of cards embossed with pictures, attributes, and a short description of a list of emperors. Furthermore, this page would include a filter bar that could accept a number of inputs such as dynasty, name, and minimum years reigned. These inputs would then be processed and provide search results based upon the provided values.

APPLICATION USE

Within the page, there are two important areas of interactivity. The first is the filter bar, which is located directly beneath the header. This bar contains the various filters that can be applied to the list of emperors. To provide a quick breakdown of the input types, I discuss the filters from left to right.

The first filter is the 'Dynasty' filter. This is a simple drop-down box that displays the various dynasties to which the emperors belong.

The second is the 'Start After' filter. This filter is a text input filter that accepts a year from which acts as a lower bound for the beginning of the imperial reigns.

Please note that both the 'Starts After' and 'Starts Before' filters work in conjunction with the neighboring BC/AD filters. While having been necessitated only by the Emperor Augustus, whose reign began before the coming of Christ,

Additionally, note that only positive numbers will be accepted. Should you input any non-numeric characters or any negative numbers, an alert should pop up asking you to input an acceptable value. As I write this, it occurs to me that I forgot to float proof the input, so decimals are fair game.

The third filter is the 'Start Before' filter. This filter is constructed the same and has the same restrictions as the 'Starts After'. The only difference is, of course, that 'Starts Before' accepts a year to act as an upper bound for the beginning of imperial reigns.

The fourth filter is the 'Minimum Years Reigned' filter. This is a text input filter that accepts a number by which to compare the total ruling years of the emperors to. Note this filter is susceptible to the same restrictions as the previous two filters.

The final filter is the 'Name' filter. This filter accepts a string which is then checked against the names of the emperors. If I have programmed this project correctly, you should only ever receive one hit after searching. It should also be understood that this field is case sensitive. The value 'Augustus' will not return the entry for the emperor as the 'a' is not capitalized. Furthermore, should a string be entered that is not equal to any name, no card will be provided.

To the right of the filters, you can find the 'Search' button. Press this when you have filled all desired filter inputs.

The second of the important areas that I mentioned earlier is the emperor card itself. Towards the end of the cards, beneath the line, there are the first words of a larger descriptive paragraph. Upon hovering over the text, a white shadow will appear. This is the first of the events created using JavaScript and jQuery. Clicking on the highlighted text will reveal the entirety of the paragraph. Please note that these briefs have been directly sourced from the emperors' individual Wikipedia pages. Unfortunately, I do not have the time to research and write summaries for each of the twenty emperors. If you wish to close the brief, all you need to do is click on the paragraph. It will then shrink back down to the collapsed view.

Lastly, note the scrollbar towards the bottom of the page. This can be used to view the rest of the offscreen emperors. It kind of blends into the background, so I deemed it prudent to highlight.

RUBRIC

|  |  |
| --- | --- |
| Interface is Authentic, Professional, Balanced | I feel that i have created a navigable layout and acceptable color palette. |
| Interface links to functional jQuery | Every interactive area on the page links to functioning JavaScript and jQuery.  Example from line 47 of *index.html*  <button onclick="filterEmperors()" class="search\_option" style=" margin: 0; position: absolute; right: 5em;">Search</button> |
| Events are Effective, jQuery | Effects are functioning and noticeable when interacted with.  Example from lines 86 – 91 of *script.js*  $('.card\_content\_para').on('mouseover', function() {          $(this).css({'text-shadow': '1px 1px 1px rgb(150, 150, 150)'})      });      $('.card\_content\_para').on('mouseout', function() {          $(this).css({'text-shadow': 'none'})      }); |
| DOM Access, jQuery | jQuery is used to determine the ready status of the document as well as source various data.  Example from lines 112 – 124 of *script.js*  let filters = {          dynasty : $('#dynasty\_select').val(),          start : {              year : $('#reign\_start\_year').val(),              bcad : $('#reign\_start\_bcad').val()          },          end : {              year : $('#reign\_end\_year').val(),              bcad : $('#reign\_end\_bcad').val()          },          length : $('#years\_reigned').val(),          name : $('#nomen\_imperatoris').val(),      } |
| DOM Dynamically Built, jQuery | jQuery is used to append markup to a selected div each time parent function is called.  Example from line 30 of *script.js*  $('.content\_wrap').append(card\_markup); |
| Documentation: Program Code and Project Description | Notes are made throughout *script.js.*  Example from lines 82-84 of *script.js*  // --- Events --- //      // Initializes events embedded in the markup for the cards.      // jQuery is used to discover the necessary divs and the .css() method is used to apply needed styling. |
| Style, jQuery | jQuery .css() method is used to apply styling to the divs within the emperor cards.  Example from lines 59 -63 of *script,js*  $('.card\_img').css({              'max-width' : '20vh',              'margin' : '.5em auto .5em auto',              'border' : '2px solid rgb(97, 82, 0)'          }); |