SVGCodingTest Technical Documentation Nida Malik

Approach

When given this coding test I thought the best approach to handle all the business requirements was to build out a MVC (model, view, controller)application. One of the main reasons that I thought building an MVC application was necessary is since MVC allows a separation between the data, the business requirements, and the presentation. My code will be cleaner and clearer. As I began my development career I really appreciated MVC the more I have been using it. Gaining more experience with MVC is very important which is another reason I choose this approach.

**Procedure**

After reading the requirements I began with building out the view. I spent an hour or so researching 3rd party libraries that would render SVG and I was blessed to find Highcharts . This is a library in JavaScript that allowed me to add charts to my application. So, then I built out two views one for the Change in Quantity Sold per Type between 2018 and 2019 which is represented in the view as **Index.cshtml** and another one for Market Share (2019) which is **MarketShare.cshtml.** For the Marketshare.cshtml view I had to a little bit more research on what exactly a Market share was and how to calculate it this provided accuracy in the data representation which is highly important. My research provided me with this formula which is Market Share by Units = (Unit Sales / Total Market Unit Sales) x 100%. Given our data set for 2019 for each type of vehicle which is 220 , 94, 168, 55, 14. I added all of these values together which equaled to 551. Then used the formula above and calculated my values. The results are below:

1.) (220/551)\*100 =39%

2.)(94/551)\*100 = 17%

3.)(168/551)\*100 = 30.4%

4.)(55/551)\*100= 9.98%

5.) (14/551)\*100=2.5%

The next step was to figure out a way to have my controller handle my data values. I did not want to pass in data values in through HTML or JavaScript. So then I built out my controller. A controller acts as a liaison between the model and the view. In the controller I had two action methods which would connect to the view which was Index and MarketShare. The Index action method is where I built out 3 arrays. The first array had the 2018 quantity sold, the second array had the 2019 quantity sold, and the last array had the categories. The main purpose of this was to create the arrays in C# and display the values using a view bag. In the Market Share action method, I placed the values for the type of vehicles and the market share values in a list. I used get and set accessors in the model the values for the Name (type of vehicle) and the Y values (which is the market share). Lastly, I changed up my navigation so the end user could easily navigate to determine which graph they wanted to see. This just allowed for better user experience.

**Testing**

This application I manually tested everything to make sure everything was working fully functional. Starting with the navigation pane and making sure that switching back and forth the correct views were displaying. Ensuring that my application was responsive on mobile and desktop. I am a big on accessibility. Lastly, checking my console to see if there were any bugs. This is huge since I was using a JavaScript library.

**End Result**





