Our product meets the vision set forth by the introduction, an interactive map which is colored based on user statistics input. We implemented a ranking system for each statistic, so the user can choose how much they want each statistic to factor into the states’ overall scores. We’ve also added an invert for negative statistics, so if a user selects “GDP,” states with higher GDP will be colored darker. However, if the user selects “Violent Crime Rate” then states with lower Violent Crime rate will be colored darker.

Our website is intended to be easy for the user to understand, and we believe we have achieved this by making all clickable elements hoverable and having self-explanatory pop-ups. **[WHAT IF WE ADD A “CLICKME” TO A STATISITIC FOR FIRST TIME USERS?]**

We have done a good job in not going over budget by carefully monitoring the GCloud platform server which is hosting our website. Obviously, we cannot afford to be online in perpetuity, but we have enough Gcloud credit to certainly keep us online until the end of the semester. We were originally concerned about 3rd party licensing, but we ended up not using the libraries we had looked into before starting the project.

We have passed all of our use cases from the requirements document. Use case 3.5 “Viewing Open Source Licenses and Site Information” says that “open source licenses and data credits” will be in the about section. Since we didn’t use any open source code, we do not need any open source licenses. We later changed our metadata design to include data credits in the metadata containers, so the data credits will not be in the about section to avoid redundant information in the site. Additionally, many use cases reference the containers by the names “Statistic Settings” and “Statistic Selection”, but those names have been changed to “Your Selected Statistics” and “Available Statistics,” respectively, to improve clarity and user experience.

The user interface meets the expectations set forth in the requirements document. We have also met the requirements set in the Software Interfaces section. We manually loaded the data obtained from Statista and cited the data appropriately in the metadata popup. We also used NodeJS, ExpressJS and Jest like we set we would. We also cited some other libraries which a) shouldn’t have been in the requirements document and b) were not used in the final product.

We have met performance requirements set by the requirements document. Our website loads in under 5 seconds. It runs running on all major desktop browsers updated within the last year. We have tested Firefox, Google Chrome, Safari and the latest version of Microsoft Edge. The website server is always running because we never deploy broken code to the master branch and the webpage only runs off the master branch.

Our only safety requirement was to check the accuracy of our data before uploading it into our database. We did that.

Our only security requirement was Denial of Service (DoS) if too many people or bots used the website at the same time or simply simultaneously refreshed the page. Since we are not the only ones with this data on the internet, and our website is simply a visualization tool. We have decided that it is very unlikely that a) too many people use our website at the same time or b) someone tries to cause a DoS error by making too many website requests. We think the worst thing that would happen in either of these cases is that we use up our Gcloud budget. A bad DoS attack would shut down the website before we are charged more than our credit allows, and the expected lifetime of the product is now less than a month. Given the short lifespan of our product and the unlikelihood of a DoS error, intentional or unintentional, we have not further addressed the issue.

**Our code is thoroughly commented, and unit tested. Our server side test coverage is \_\_\_\_\_ and our client side test coverage is \_\_\_\_. We have also taken great care to make our website intuitive to new users. This meets our software quality requirements.**