**The United States College Information System**

Alejandro Zepeda

University of San Francisco

Department of Computer Science

**1. Introduction**

**1.1 Project Overview**

Often times people presented with a set of data generally don't know where to start. While raw data is useful, the average user doesn't have time to look through a large data set and find its significance. It is because of this that data visualization has become increasingly important over the years. One of the fields in which this holds true is in academia. In particular, academic institutes throughout the United States have numerous attributes of information that describe them. Some of these include tuition cost, location, revenues, financial assistance provided, and student demographics. It would be useful if these attributes could be visualized and compared amongst other institutions.

Imagine deciding on which educational institution would be the best fit and not knowing where to begin. Prospective students deciding on which institution would best suit them would benefit from finding out information from various institutes. It would be useful to have some system that was able to describe an institution based on location, financial information, educational information, and demographics. While it may be possible to search each institutions site individually, it would take the user a long time to be able to note and compare each institution amongst the others. Fortunately, this information is available and can be visualized in a way that will be more productive for the user.

In order to allow the user to access the data in a more meaningful way, our project, the United States College Index system (USCIS), aims to utilize the gathered information of the institutions throughout the United States to create different visualizations. Rather than be intimidated and confused by the large data set, our users will be able to visualize institutions on a geographical map and find out specific information on each institute. They will be able to compare different values of institutes such as tuition cost, total revenue, grants awarded, and many other criteria that are available. This implementation will be in the form of a website and be written using the D3 (a JavaScript based language).

**REMEMBER TO INCLIDE A SUMMARY OF MY DELIVERABLES**

**1.2 Team Overview**

**1.2.1 Project Sponsor:**

Sophie Engle

[*sjengle@cs.usfca.edu*](mailto:sjengle@cs.usfca.edu)

Professor Engle received her Ph.D. in Computer Science from the University of California, Davis in 2010, and her B.S. in Computer Science with a minor in Mathematics from the University of Nebraska at Omaha in 2002. Her research focus is on computer security, including topics such as vulnerability analysis, insider threat, and electronic voting. Her research interests also include topics such as information visualization, network and graph theory, and computer science education.

**1.2.2 Team Members**

Alejandro Zepeda

[*azepeda2@usfca.edu*](mailto:azepeda2@usfca.edu)

Alejandro Zepeda is a current undergraduate student working towards a Bachelors Degree in Computer Science at the University of San Francisco. He has always had a strong interest in technology and its applications in the world around him. He is particularly interested in game engineering and data visualization. He has programmed in Java, Python, C, C#, C++, HTML, and is currently working with D3.js and JavaScript.

**3. Project Design**

**3.1 Introduction**

The USCIS is designed to visualize data that is collected by the IPEDS (Integrated Postsecondary Education Data System) in order to provide the users with a better understanding of over twenty-six thousand universities, colleges and other post-secondary educational institutes in the United States.

**3.1.1 Data Set**

We have chosen 149 columns of data for each institute to include in our subset of the data. This data is collected from several different databases to make sure our data visualization system can deliver the most meaningful information. Being that our project is entirely based on the data collected by the IPEDS, we will begin by describing information included in the data set:

1. Basic information: the data set contains most of the basic information about each school, including school classification (private or public), majors that the schools offer, geographic data including cities, states, zip codes, and GPS coordinates, as well as other information that the user may be interested in.
2. Financial data: the data set contains the major financial information for each institute including revenues, debt, funding, funds from local government, federal government, and more. It also contains information on tuition and fees, as well as awards and scholarships that the institute offers.
3. Education data: the data set includes the types of certificates and degrees the school offers, the total number of degrees granted from the school and SAT/ACT scores for admission.
4. Campus population: the data set describes the student body and has information on race, academic level, award and aid received.

**3.2 Functionality**

The core functions of this project are what we expect to have finished by the end of the project. Through these functions, we will ensure that our users are provided with useful and easy to understand interpretations of the data.

**3.2.1 General University Visualizations**

When the USCIS homepage is loaded, it will display pre-set visualizations based on the data. The user will be presented with an option of which visualization they want to use. The data used in the visualization will be based on a subset of data that we choose to include. The user will be able to hover over a university, which will display information about the university such as: the university name, whether it is public or private, tuition cost, total revenue, total enrollment, and average grants awarded.

**3.2.2 Fully Customizable Visualization**

For each school, we will implement the “fully customizable visualization” which is an upgrade of the basic information function. The user can select which information he or she wants to be displayed. Users can also configure how to visualize the data. For example, a user can choose to display the ratio of asian and white students and display them in a pie chart. Users can choose any of the values we have in the system and have them displayed in a window. The system will also suggest some visualization configurations to the user.

**3.2.3 Comparing with other Schools**

The user will be allowed to choose up to eight universities that they wish to be compared. Once they have chosen a university, it will be added to a subset of the data and will be displayed through a pie chart. The user will then be allowed to choose to compare the schools based on total revenue, tuition, average university grants, average federal grants, average state grants, or average loan amount.

**3.2.4 Map Based Data Visualization**

One of the USCIS visualizations includes a map implementation of the data. The map will work in the form of a choropleth (also known as a choropleth map), in which each states total for a particular value (such as total revenue, tuition cost, or average university grants awarded) is added and compared to the total amount. This value determines the states color, where a lighter color represents a lower value, and a deeper color represents a higher value. The user will be able to hover over a state and see the state total value and the name of the state. Each individual university will be plotted on the map in the form of a point. The user will be able to hover over the point.

**3.2.5 Extra functionality**

There are various features that we hope to implement if we have extra time to work on the project. One of the functions is the ability to display each university as a point in the map. This point will provide information about the university (such as the university name, whether it is public or private, tuition cost, total revenue, total enrollment, and average grants awarded) as the user hovers over the point. Another feature we are considering is to include the data for various years, rather than just for 2010. This will allow the user to see how the values changed over the years. An example of this would be to check how much tuition has raised over the years. The challenge behind this is that we have to manually clean the data since not not all of the universities have complete data for years prior to 2010.

**3.2.6 User Interface**

The base implementation of the USCIS is a web-based system. The system is intended to be accessed through the web, allowing users to access the systems anywhere, without requiring installation of any libraries other than a working web browser. The user will be allowed to click on radio buttons, which will set which visualization is currently being displayed to them. They may choose from a scatter plot, a pie chart, or the map visualization. For each visualization, there will be another set of radio buttons that will allow the user to decide which value int the data set is being displayed, such as tuition, revenue, or average university grants awarded.

**3.2.6 Performance**

The main focus of performance in our project is to be able to load the USCIS page in a short amount of time. Our biggest hurdle with this goal is that we are implementing our visualizations using D3, a javascript based language which loads the entire data file at once. It is because of this that we can't just load certain universities in as we choose, but have to load the entire data set instead. It is because of this that we have created smaller sub sets of the main data set for faster load times. Some of the sub sets we have created include only the top 100 universities based on revenue and the top 100 universities based on average university grants awarded. It is because of this that we limited our biggest data set to only include 149 columns of data for each school as opposed to the origin set. This will ensure that our users are able to access the USCIS with efficient load times.

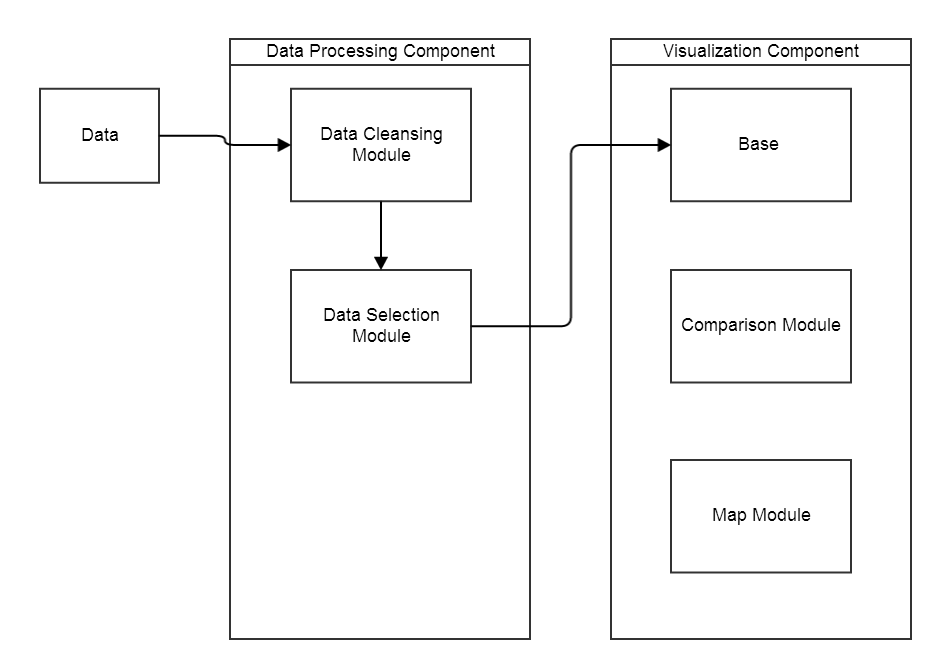
**3.2.7 Security**

As far as security goes, there isn't much required for the implementation of the USCIS at the moment. All of the data needed to use the visualizations comes from the data sets we have derived from the IPEDS data set. We don't require the users to log in or register so there isn't any user data at risk. All of our data is accessible through GitHub and the GitHub pages servers.

**3.3 Design**

The USCIS project is based on the HTML 5, JavaScript, and D3 languages. The HTML aspect of the project allows us to present the user with our created visualizations. D3 is a JavaScript library that allows developers to create elaborate visualizations based on data sets. All of the calculations and modules are written in JavaScript.

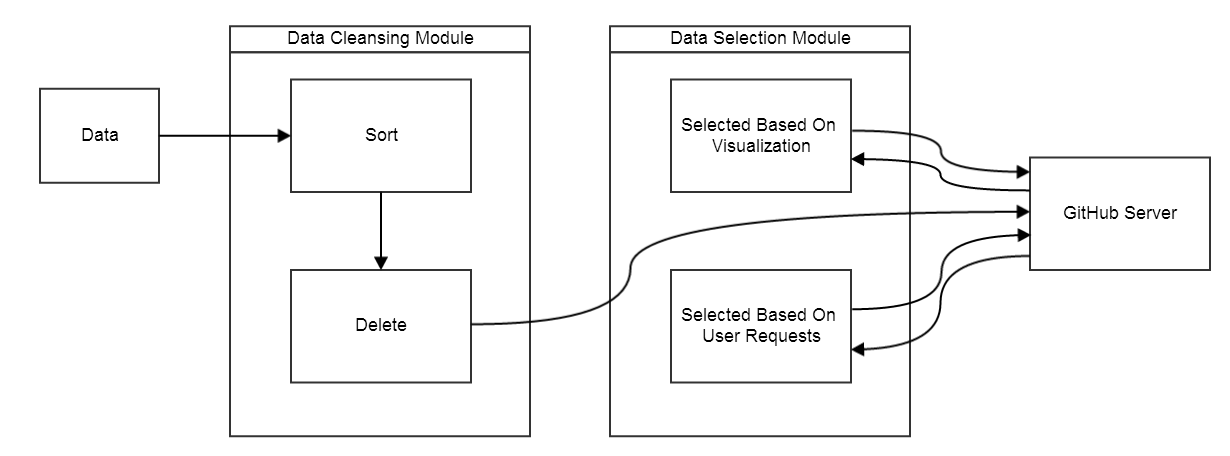
The USCIS has two main components used to create the entire system: the Data Processing Component and the visualization component. The Data Cleansing Component made from two modules: the Data Cleansing Module and the Data Selection Module. The visualization component is made from three modules: the Comparison Module, the Map Module, and the Visualization Base Module.

*Figure 1.* Flowchart of USCIS Components and modules

The Data Processing Component processes the data and stores it on the GitHub Server. The Visualization Component reads in the data from the server in order to create the visualizations for the user.

**3.3.1 Data Processing Component**

The Data Processing component reads in a set of data and performs a process on the data set. The data cleansing portion of this component is written in Python and is only used by the developers to clean and create sub sets of the original IPEDS data set. Once a data set is created, it is stored on the server to be utilized by the Visualization Component. This component consists of two modules: the Data Cleansing Module and Data Selection Module.

*Figure 2.* Flowchart of Data Processing Component

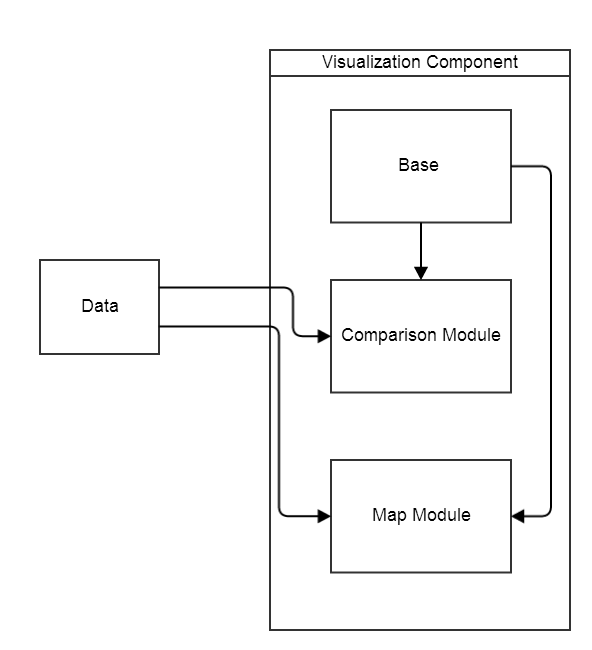
The Data Cleansing Module first reads in a data set that will be altered in some way. The sort portion of the module will perform sorting based on the developers criteria. While most of time the data will be sorted by Unique ID Number, the module can also sort based on other criteria when creating a sub set. For example, if we are creating a sub set of the top 100 universities based on average university grants given., the module will sort by average university grants given.

The origin data set we obtained from IPEDS contains information from 1987-2010. We found that we would not be using the data before 2005, since it may be to old to be relevant for our users so we decided to implement the delete portion of the Data Cleansing Module. This portion performs “deleting” of the unwanted data. For example, when cleansing the origin data set, the delete portion of the module only copied over the data for the schools for the year 2010 into the sub set and ignored the rest. We also looked at which values were most important for our users such as tution, revenue, average state grants given, average federal grants given, and average university grants given. If a university lacked any of these fields, we removed it from the data set since it is considered incomplete.

The Selection Module will select data from the data from the data set stored on the server, which will be used by the visualization module in order to create the visualizations. These selections may be hard coded into the USCIS page for predetermined visualizations such as top 100 universities based on revenue. The selection can also come from user requests through the search feature of the USCIS. This allows the user to choose the university they want to be included in the visualization.

**3.3.2 Visualization Module**

The Visualization Component is written in JavaScript. The user's web browser will be in charge of processing this component. It includes three modules: Base Module, Comparison Module, and Map Module.



*Figure 3.* Flowchart of Visualization Component

Due to the implementation of D3, all of the methods that that use information loaded from a data set to be implemented within the data loading function. It is because of this that the Comparison Module and the Map Module are required to handle input from the data.

The Base Module displays all of the figures that do not require any data from the data set. For example, the user interface doesn't require any data from the data set. This includes buttons to switch between visualizations such as a scatter plot, pie chart, or the choropleth map. Another example is the search form, which allows the user to input the name of a university they want to be displayed.

The Comparison Module handles the creation of the comparison visualizations such as the scatter plot and the pie chart. The Map Module handles the creation of the choropleth map based on the data set.

**4. Implementation**

**4.1 Results**

Now that we are at the end of our project, we have a complete USCIS website up and running. We were able to implement that Data Cleansing Component that we used to create sub sets of the original data set and the Visualization Component which creates the visualizations for the user.

The Data Cleansing Module goes through the original data set and only copies over the most recent information for each univesity into our sub set. It also allows us to choose diffent parameters to sort the data in order to create smaller sub sets such as the top 100 universities based on revenue or average university grants given. The Data Cleansing module is written in python.

The Data Selection module was also created successfully. It allows USCIS to select a set of data from the GitHub server and create a visualization based on the data. It also allows the user to be able to search for individual schools to be compared via the pie chart visualization. At the moment, the user can choose up to eight universities at a time. This seems to be the most effective for a pie chart since adding any more schools seems to overcrowd the chart. This module is written in JavaScript.

The Visualization Base Module is in charge of creating all of the figures that don't depend on loading data. This includes the buttons that allow the user to select which visualization they want to use, the form that allows the user to search for a specific university to be visualized via the pie chart, the list of suggested universities, the list of universities currently selected, and the buttons that change the value being compared in the visualization (such as tuition, revenue, average university grants, average state grants, average federal grants, average loans taken, total students enrolled, and total employees) which vary depending on the visualization. This module is written in JavaScript.

The Comparison Module is in charge of creating all of the visualizations which compare individual schools amogst each other. At the moment, this module is allows the user to create scatter plots, pie charts, and bar charts.

The scatter plot is based on a pre-selected data set which includes the top 100 universities based on tuition. The scatter plot creates the x-axis and y-axis based on the user's selection. The user can utilize the scatter plot to see the correlation between values such as: tuition, revenue, average university grants, average state grants, average federal grants, average loans taken, total students enrolled, total degrees awarded, and total employees. If the user hovers over a particular university (point on the scatter plot) they are presented with a text box that shows them the values for that particluar university listed above (tuition, revenue, average university grants, etc). They are also able to distinguish the private universities from the public by selecting to highlight public or private. The Comparison Module highlights all of the private universities gray if “highlight privates” is selected or vice versa for public universities.

The pie chart initially begins by visualizing a pre-selected data set which includes the top 5 universities based on average university grants awarded. The user also has the option to search for any universities and add it to their list of universities. Since more than 8 slices in a pie chart doesn't present well, we are limiting our users to search up to 8 universities at one time, removing the oldest universities they chose. The user can choose the value being compared in the pie chart such as: tuition, revenue, average university grants, average state grants, average federal grants, and average loans taken. When the user hovers over a slice in the pie chart, they are presented with all of the information for that school that can be chosen for this visualization (tuition, revenue, average university grants, average state grants, average federal grants, average loans taken, total students enrolled, total degrees awarded, and total employees).

The bar chart is based on a pre-selected data set that includes the top 50 universities based on tuition. The Comparison Module creates the bar chart and allows the user to sort or unsort the data. The user can select which value is being displayed ranging from the university's tuition, revenue, average university grants, average state grants, average federal grants, average loans taken, total students enrolled, total degrees awarded, and total employees. The y-axis represents the value being compared, while the x-axis contains each universities' bar along with its name. The Comparison Module also allows the user to highlight the public schools, private schools, or keep them the same color.

The Map Module creates a visual representation of state total values in the form of a choropleth map. Our method for determining state values calculates the total value for each state, gets the state average, and fills in the state with a certain color depth depending on the value. For example, the states with the lower average value are represented with a lighter color while the states with higher average value are represented with darker color. There is a legend underneath the map that signifies which color represents very low, low, medium, high, and very high. If the user hovers over a particular state, the Map Module presents them with the state name, state total, andf state average for the chosen value. The user can choose between tuition, revenue, average university grants, average state grants, average federal grants, or average loans taken. Each individual university is represented on the map by a point, which can also be hovered over and presents information on that particular university. The Map Module presents the user with the university's tuition, revenue, average university grants, average state grants, average federal grants, average loans taken, total students enrolled, total degrees awarded, and total employees.

**4.2 Obstacles**

There were definitely several obstacles that I encountered during the development of this project. Some of them involved lack of experience with D3 and missing data, while other's involved time management and focus. We initially started with over sixty thousand universities to include in our visualizations but soon after coding our prototypes, I noticed that there were a lot of universities with missing data. Some of them were missing their values for tutiion, revenue, average university grants, average state grants, average federal grants, or average loans taken. Since most of my visualizations relied on this information, I had to develop a module to remove incomplete universities. This brought the data set from over sixty thousand down to twenty-six thousand. While this wasn't too much of a delay, it still took some time to create the data cleansing module. Another obstacle that came up was having issues with my prototypes. I often found that when trying to include text when hovering over a point, state, or pie slice, there was always an issue with the outcome. For my scatter plot text, I couldn't figure out how to get the background for the hover text in the right place, which caused it to overlap with the points and not be legible. As for my choropleth prototype, I had an issue with the function that chose the color depth for each state. Instead of getting a range of colors depeding on the states value, I was getting either too dark of color depth or none. After consulting with Professor Engle, I changed my color scheme to a better color palette, with more range in values. Additionally, I changed the function that determined color depth to use state averages instead of state totals for the color depth ratio.

Above all of the coding obstacles I came across, the most significant one was time management. I decided to take three computer science courses this semester which depend heavily on creating projects, which proved to be more difficult than anticipated. To add to the diffuclty each course required coding in different language (Java, JavaScript, C++, Ogre3D). This caused me to fall behind on some of my milestones and ultimately led to me having to branch off into my own project since I was not on the same schedule with my teammates. It was due to this change that my timeline changed from our original timeline.

**4.3 Timeline**

|  |  |  |
| --- | --- | --- |
| **Milestones** | **Timeline** | Date Completed |
| Selecting the dataset to visualize. Choosing development tools and installing D3. | **week 03** | **2/7/2014** |
| Cleaning the dataset: picking the features which are useful. Begin to study D3. | **week 04** | **2/10/2014** |
| Implementing some simple D3 examples for presenting several features from the dataset. | **week 05** | **2/17/2014** |
| Adding more features to the dataset and experimenting with more D3 techniques for presenting the data. | **week 06, week07** | **2/24/2014,**  **3/3/2014** |
| Deciding what kind of graphs we should implement in order to compare different universities’ features. (ex: pie graph, tree diagram, tube, etc.) | **week 08, week09** | **Missed meeting** |
| Preparing midterm presentation: We should implement the functions for showing the specific information of universities and comparing it with different universities. | **week 09** | **Missed meeting** |
| Adding the map to our project: We need to locate the universities on the map by their longitudes and latitudes. We should begin to build the website for users: we need to publish the data on the website and have an index or a guide page for users. | **week 10, week 11** | **Missed meeting** |
| Draft test plan; By this time, our server should be able to handle the requests from users (ex: Register and login/logout, Searching, leaving feedback, etc.) | **week 11, week 12** | **Timeline changed** |
| Draft documentation; Testing the project: making sure all the information is correctly presented and checking any possible bugs on graphs. | **week 12, week 13** | **Timeline changed** |
| Adding more functions for users after receiving feedback. Preparing for the final presentation: Our project should have full functionality by this time. | **week 13** | **Timeline changed** |
| Final presentation: we should be able to show and select universities on the map. We also need to create a demo that introduces all features users can utilize on the website. | **week 14** | **Timeline changed** |

|  |  |  |
| --- | --- | --- |
| **Updated Milestones** | **Timeline** | Date Completed |
| Selecting the dataset to visualize. Choosing development tools and installing D3. | **week 03** | **2/7/2014** |
| Cleaning the dataset: picking the features which are useful. Begin to study D3. | **week 04** | **2/10/2014** |
| Implementing some simple D3 examples for presenting several features from the dataset. | **week 05** | **2/17/2014** |
| Adding more features to the dataset and experimenting with more D3 techniques for presenting the data. | **week 06, week07** | **2/24/2014,**  **3/3/2014** |
| Deciding what kind of graphs we should implement in order to compare different universities’ features. (ex: pie graph, tree diagram, tube, etc.) | **week 10** | **3/28/2014** |
| Create another prototype using a dataset with five categories. Modify Website to remove former teammates prototypes and work. | **week 11** | **3/31/2014** |
| Create another prototype that is not a scatterplot (pie chart, bar chart, scatterplot matrix). I created a pie chart. | **week 12** | **4/12/2014** |
| Adding the map to my project: Be able to show state values for each school in the state | **week 13** | **4/19/2014** |
| Fix any bugs with prototypes and create a webpage displaying all of the visualizations in one place. | **week 14** | **4/26/2014** |
| Prepare final presentation to present next week. Add another visualization prototype. I added a bar chart and added a search feature. | **week 15** | **5/2/2014** |
| Final presentation: we should be able to show and select universities on the map. We also need to create a demo that introduces all features users can utilize on the website. I fixed bugs with pie chart and map visualization. | **week 16** | **Missed Presentation.**  **Fixed bugs 5/9/2014** |
| Fix any bugs found by user testing/self tests and prepare final report. | **Finals week** | **5/14/2014** |

**5. Test Plan**

**6. Deliverables**

**6.1 Primary Deliverables**

**File LOC Filesize**

index.html 89 6kb

data/RemoveIncompleteRows.py 45 2kb

data/RemoveIncompleteRowsFrom11Columns.py 36 2kb

data/RemovePastYears.py 27 1kb

data/GetNamesForm.py 19 1kb

javascripts/map.js 262 14kb

javascripts/barchart.js 173 7kb

javascripts/piechart.js 292 13kb

javascripts/scatterplot.js 231 10kb

prototypes/AlexFirstPrototype.html 108 6kb

prototypes/barchart.html 151 7kb

prototypes/AlexSecondPrototype.html 224 11kb

prototypes/Map.html 190 9kb

prototypes/thirdPrototype.html 162 8kb

prototypes/piechart.html 36 1kb

USCIS.html 2861 157kb

Number of files: 12 Total lines of code: 4906 Size: 255kb

The only starter code that was used for this project is the D3 library, the us.json file which includes the geometry and names of the states, and the stylesheets for the website.

**6.2 Other Deliverables**

data/map.csv

data/allUniversityData11Columns.csv

data/top100Tuition.csv

data/top50Tuition.csv

data/allUniversityData.csv

data/top50with5categories.csv

data/top5Grants.csv

data/top100Grants.csv

stylesheets/pygment\_trac.css

stylesheets/stylesheet.css

images/bg\_hr.png

images/blacktocat.png

images/icon\_download.png

images/sprite\_download.png

json/us.json

params.json

**6.3 Documentation**

**6.3.1 User Documentation**

You can access the user documentation at: <https://github.com/azepeda2/DataVisualizationProject/blob/gh-pages/README.md>

**6.3.2**