Title: Customizable Analysis and Visualization Tool for COVID Cases

Team Members:

- Sam Hartle, shartle2017@my.fit.edu
- Calvin Burns, cburns2017@my.fit.edu
- Nicole Wright, nwright2017@my.fit.edu
- Stian Olsen, shagboeolsen2017@my.fit.edu

Advisor: Dr. Philip Chan, pkc@cs.fit.edu

Client: Dr. Philip Chan, pkc@cs.fit.edu

Date(s) of Meeting(s) with the Client for developing this Plan:

- Wednesday, 8/26/2020 @ 11 AM

Goal and Motivation:

- **Goal:** To create a website that shows COVID case data, allows users to perform customizable analyses/visualization of results, and allows users to add additional pieces of data.
- **Motivation:** COVID is still ongoing. There are various COVID dashboards out there, but the analyses are pre-determined. We are developing a COVID dashboard where the analyses are not predetermined. The user has the ability to customize and get the results they are looking for rather than being stuck with certain results.

Approach (key features of the system):

Feature 1:

Users can select specific variables and perform custom operations (ranking, average, etc) on the selected variables. This allows users to compare filtered data across a multitude of different categories (age, location, schools, month, hospitalized vs deaths, etc).

Feature 2:

Users can make comparisons among variables according to some common standard. For example, in order to compare the number of cases between Brevard County and Orange County, one might try to compare the number of cases per *n* number of people rather than the actual number of cases. This allows for a fair and proportionate comparison. Users will make these kinds of comparisons rather than making comparisons between actual raw data.

Feature 3:

Users can display their results from variable operations using various charts, graphs, and plots. This provides the ability to see any trends from the selected data and does not limit users to using predetermined statistics and plots.

Feature 4:

Users can save custom visualizations to their own unique workspace. These charts/graphs/plots will give the user the ability to easily access the custom analysis at any time and see the analysis with data updated daily.

Feature 5:

Users can make use of the custom visualizations and layer plots on the visual. Overlaying related data and plots makes statistical comparisons much more efficient and puts all the data in one location for users.

Feature 6:

Users can add additional types of datasets (airline travel data, school data, etc). Each user will be associated with a set of data, where their uploaded data is private to them and only them. For example, this feature could allow users to analyze the infection rates in areas with face-to-face classes versus areas with online education.

Feature 7:

Users can add and analyze multiple datasets at a time, delete existing datasets, and have their current datasets automatically updated. For an existing dataset, automatic pulling of the most recent data will occur daily.

Feature 8:

Users can submit an application to make their private datasets public. Users with admin level permissions can access a tool for reviewing and approving data sets. An application will require a sample of the data, source, API URL (if applicable), description, and contact information for further questions.

Novel features/functionalities:

1. Users being able to create/save custom graphs (Features 3 and 4) is not available on many other dashboards. Most of the plots are pre-set. Giving users the ability to pick their variables and operations to plot and show comparisons expands the functionality of a traditional dashboard.
2. Allowing users to layer plots (Feature 5) will show relationships between data plotted on a shared x-axis. This can help show correlations between trends. Other dashboards do not have this functionality.
3. Giving users the ability to add additional types of datasets is not possible in other dashboards (Features 6-8). This feature allows users to perform analysis on data that is currently not available and potentially share data for other users to use.

Technical Challenges:

1. Learning how to use Django or AWS
2. Learning about various Javascript frameworks and integration with Django/AWS
3. Learning about GIS and Spatial Database Relations
4. Learning about RestAPIs and file formats for reading data (JSON, CSV, etc)

Milestone 1 (Sep 28): itemized tasks:

A. Compare and select technical tools
  a. Web Backend Candidates
    i. Python (Django)
    ii. AWS
  b. Web Frontend Candidates
    i. JQuery
    ii. React.JS
    iii. Leaflet.JS (Mapping)
    iv. Chart.JS (Interactive Plots)
    v. Typescript
    vi. AngularJS
    vii. AG-grid
  c. Database Candidates
    i. PostgreSQL
    ii. SQLite
  d. GIS
    i. QGIS
    ii. PostGIS
  e. Interactive Graphs
    i. Char.js
B. Provide demos using chosen technical tools that shows
  a. Importing data from API endpoint
  b. Performing basic operations of data like sorting
  c. Displaying a graph on the frontend
  d. GIS display and edit
C. Resolve technical challenges
  a. Creating a script to get the same environment on all teammates machines (Windows and MacOS)
  b. Designing/configuring a cloud-based server for testing application in a live environment
  c. Setting up Frontend web framework and graphing framework
  d. Create abstract database structures using UML
D. Compare and select collaboration tools for software development, documents/presentations, communication, task calendar
  a. Software Development: GitHub
  b. Documents/Presentations: Google Drive
  c. Communication: Slack, WhatsApp, Email
  d. Task Calendar: GitHub/Google Drive/Google Calendar
E. Create Requirement Document
F. Create Design Document
G. Create Test Plan

Milestone 2 (Oct 26): itemized tasks:

A. Implement, test, and demo *selecting different variables and performing various operations on them.*
B. Implement, test, and demo *making proportionate comparisons between counties.*
    a. Example: Comparing/ranking counties by cases per 10,000 people
C. Implement, test, and demo *ability to plot data on graphs and maps.*
D. Implement, test, and demo *saving custom visualizations to specific users.*

Milestone 3 (Nov 23): itemized tasks:

A. Implement, test, and demo *the option to layer plots.*
B. Implement, test, and demo *importing new/other datasets.*
    a. Airline/school action data
C. Implement, test, and demo *adding/analyzing multiple datasets at a time, deleting existing datasets, and having current datasets automatically updated*.
D. Implement, test, and demo *admin dataset approval process.*

Task matrix for Milestone 1

Task	Stian	Sam	Nicole	CJ
Compare and select Technical Tools	Investigate web frontend candidates	Investigate database candidates	Investigate GIS tools	Investigate Web-backend
"hello world" demos	Hello World with selected web frontend tool	Initial schema design for selected database	Small GIS demo	Web Server Demo
Resolve Technical Challenges	Setting up frontend tool and necessary packages	Create abstract database structures for locations, users, datasets, and plots.	Learn how to setup and use GIS tools	Create a local environment that all teammates can run. Design a cloud based server for testing the live application.
Compare and select Collaboration Tools	Software development	Presentations	Documentation, meeting notes, organization	Communication, Task calendar
Requirement Document	write 25%	write 25%	write 50%	Proof/Edit
Design Document	write 25%	write 25%	write 50%	Proof/Edit
Test Plan	write 25%	write 50%	write 25%	Proof/Edit