Customizable Analysis and Visualization Tool for COVID Cases

Milestone 1

Team Members

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

Faculty Advisor/Client

- Dr. Philip Chan, pkc@cs.fit.edu

Progress Matrix

Task	Completion %	Stian	Sam	Nicole	CJ	To do
1. Investigate tools	100%	25%	25%	10%	40%	none
2. “Hello World” demos	100%	15%	35%	5%	45%	none
3. Requirements Document	100%	10%	10%	70%	10%	none
4. Design Document	100%	70%	10%	10%	10%	none
5. Test Plan	100%	10%	10%	70%	10%	none

Task 1 Summary - Investigate Tools

- Frontend
  - HTML/JavaScript (Using Django Templates)
- Plotting
  - Charts.js
  - Leaflet.js
- Backend
  - Python/Django
- Databases (with GIS support)
  - SQLite with SpatiaLite extension
  - PostgreSQL with PostGIS extension
    - Provided easy-to-use admin portal that may make our lives easier
    - Leaning towards that option

Task 2 Summary - Demos

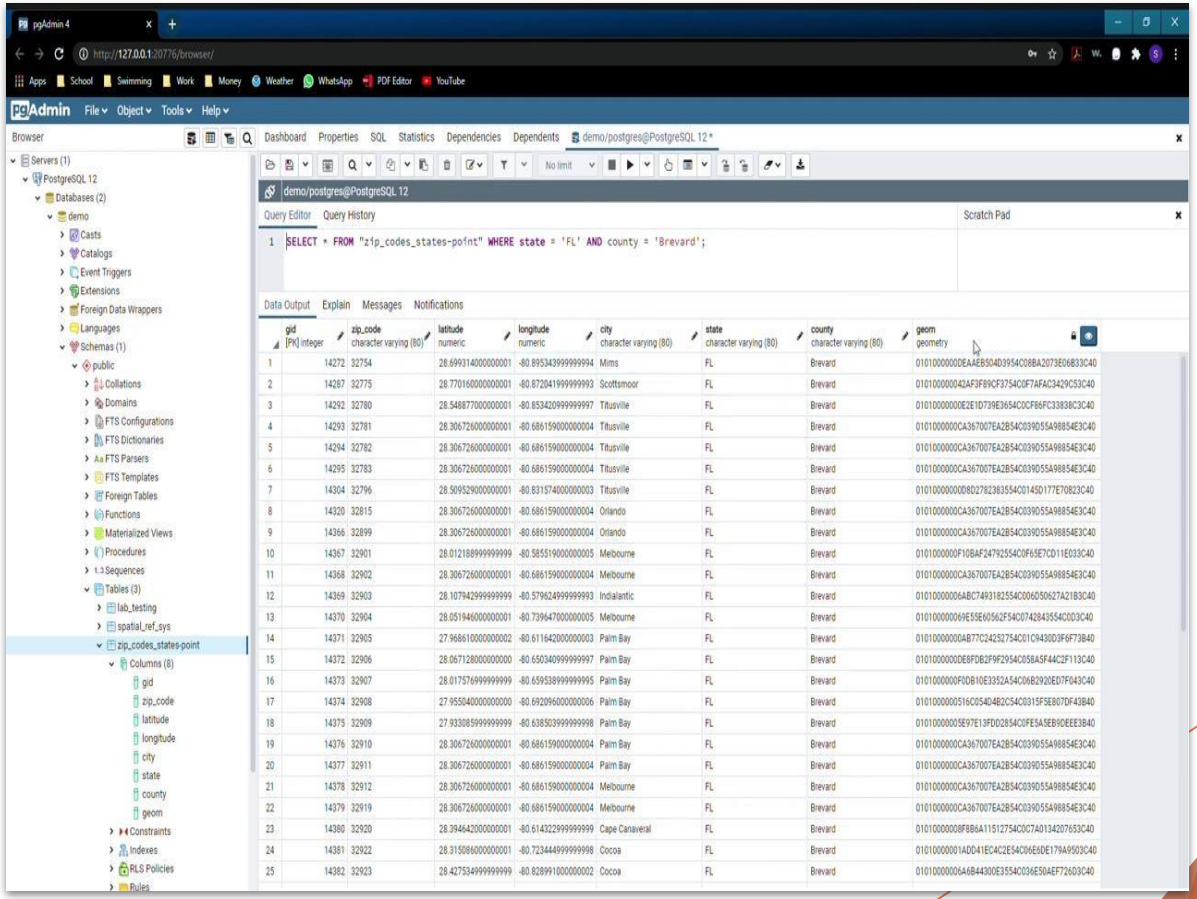
- 4 “Hello World” demos
  - Plotting graphs using Chart.JS
  - Ability to layer GIS data on a web browser
  - Taking user input from the frontend and immediately displaying it
  - PostgreSQL with GIS support (PostGIS)

Demo - Plotting graphs with Chart.JS

Demo - GIS Layering on a Browser

Demo - User Input

Demo - PostGIS



Task 3 Summary - Requirements

- Defined the system's individual features in detail
  - Explained functional requirements for each of them
  - Priority levels were also defined for each feature
- Obstacles included an initial lack of specificity on certain features
  - Most prevalent with higher priorities
  - Feedback from advisor/client helped us understand choices we must make to fulfill the requirements of higher priority features

Task 4 Summary - Design

- Defined the system architecture
- Layed out roles and methods each type of user will have available
- Defined various methods for handling data on both frontend/backend
- Database design, multiple “mock-ups”, and their objects/actions were included near the end of the document for a proposed UI design

Task 5 Summary - Test Plan

- Layed out features that needed to be verified for correct behavior
- Defined expected functionality
  - Provided at at least 2 test cases to verify sub-features
- Feature usage/inputs were formulated from the *user's* point of view
- Obstacles included an initial lack of test cases for testing sub-features

Challenges

- Lack of usage among group members for certain tool categories
  - Databases with GIS support
  - Frontend plotting frameworks
- Configuration process for database demos
  - CSV to Shapefile conversion for storing geometry (GIS) data posed a challenge
- Initial lack of specificity for Requirements, Design, and Test Documents

Task Matrix for Milestone 2

Task	Stian	Sam	Nicole	CJ
1. Set up Django environment	Configuring local machine (20%)	Configuring local machine (20%)	Configuring local machine (20%)	Will write majority of bash script (40%)
2. Create database model (#s are order of completion)	2: Location (easy), 8: Dashboard (JSON: Will need research)	3: Metric (easy), 6: Operation (Need to learn about SymPy)	1: User (easy), 7: Plot (Chart.js)	4: DataSet, 5: DataPoint (database relations and queries)
3. Import data to newly created database via API or CSV	Import Lockdown Data	Import Population Data	Import Mask Mandates	Import FDOH case line data via Python Script
4. Implement Feature 4.1 (Customizable Operations on Variables)	Create basic UI and display choices for variables and operations. Allow users to *select* variables and *perform* operations on them. Pass selected variables and operations to backend	Implement 40% of available operations on appropriate variables. Assist Stian with frontend development.	Implement 20% of available operations on appropriate variables. Assist Stian with frontend development.	Implement 40% of available operations on appropriate variables. Assist Stian with frontend development and oversee frontend, backend, and database interactions

Questions?