Customizable Analysis and Visualization Tool for COVID Cases

User/Developer Manual

Authors: Sam Hartle, Calvin Burns, Stian Olsen, Nicole Wright

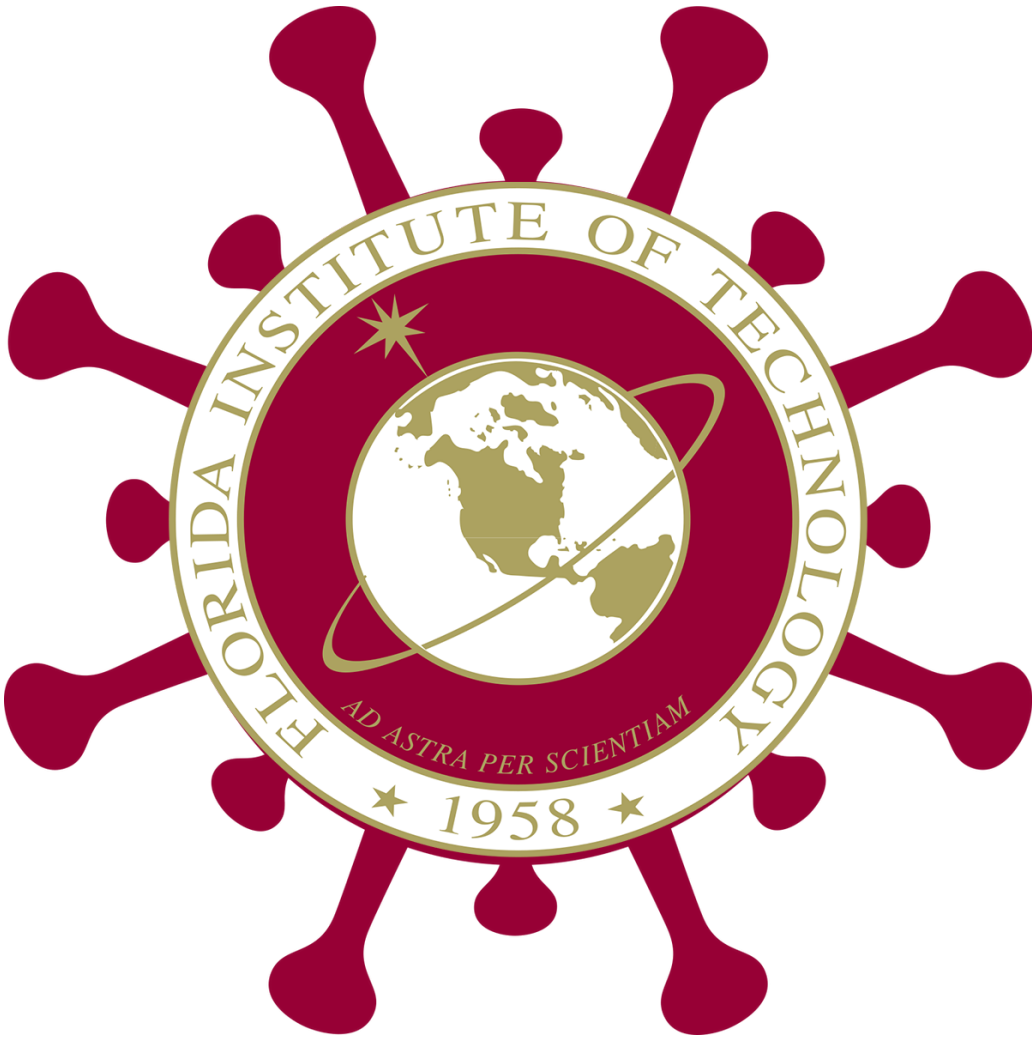


Table of Contents

1. Introduction

1.1. Mission Statement

The Customizable Analysis and Visualization Tool for COVID Cases is a web application that shows COVID case data, allows users to perform customizable analyses/visualization of results, and allows users to add additional pieces of data. There are various COVID dashboards out there, but the analyses are pre-determined. We developed 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.

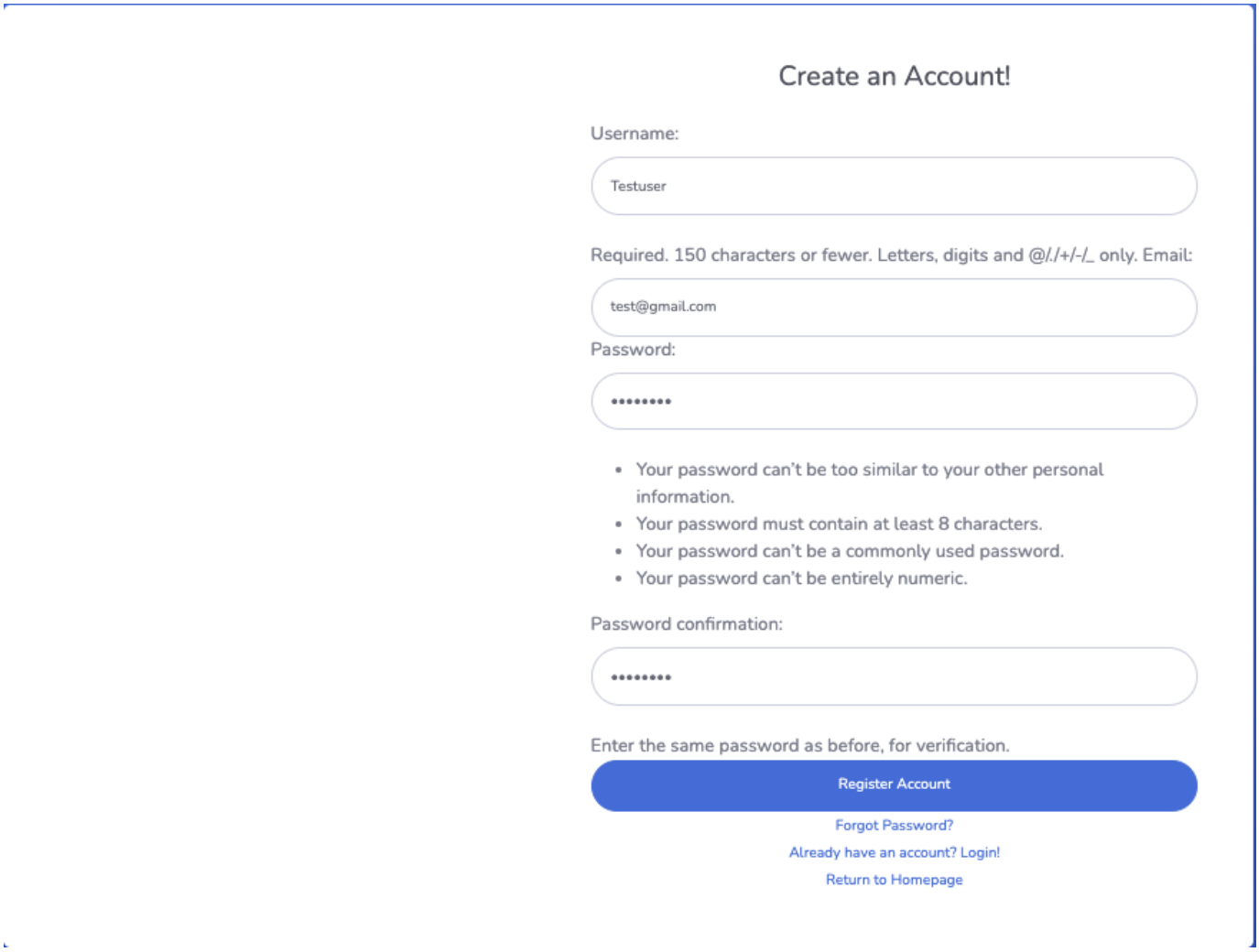
1.2. Motivation

During the COVID-19 pandemic, our advisor noticed the lack of interactive and customizable COVID-19 statistics websites/dashboards. The majority of dashboards have predefined datasets, graphs, and maps. No dashboards allowed the user to perform their own statistical analysis or create their own plots. No dashboards allowed users to upload additional data to use in conjunction with the sites existing data. Lastly, few dashboards have layered plots, and none allow the user to create their own layered plots. This project presents a customizable solution to this problem.

2. Creating an Account

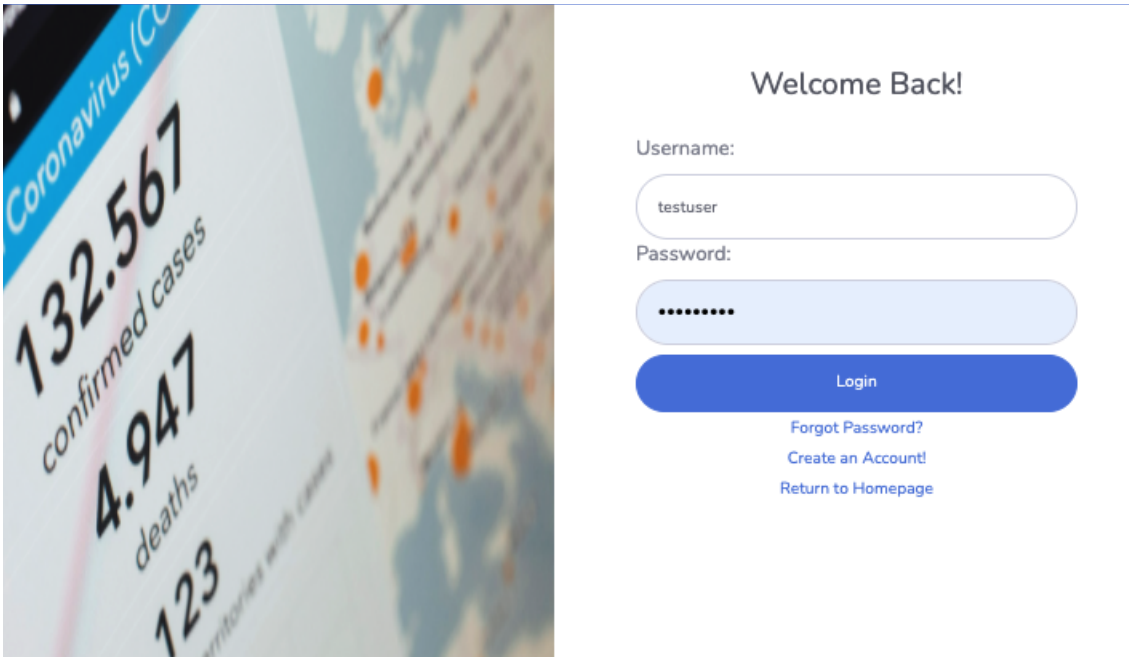
2.1. Register

To get started with data customization, a user must create an account. To make an account, a user must navigate to the “Register” tab in the top right corner of the website. On this page, they can then enter their username and password information.



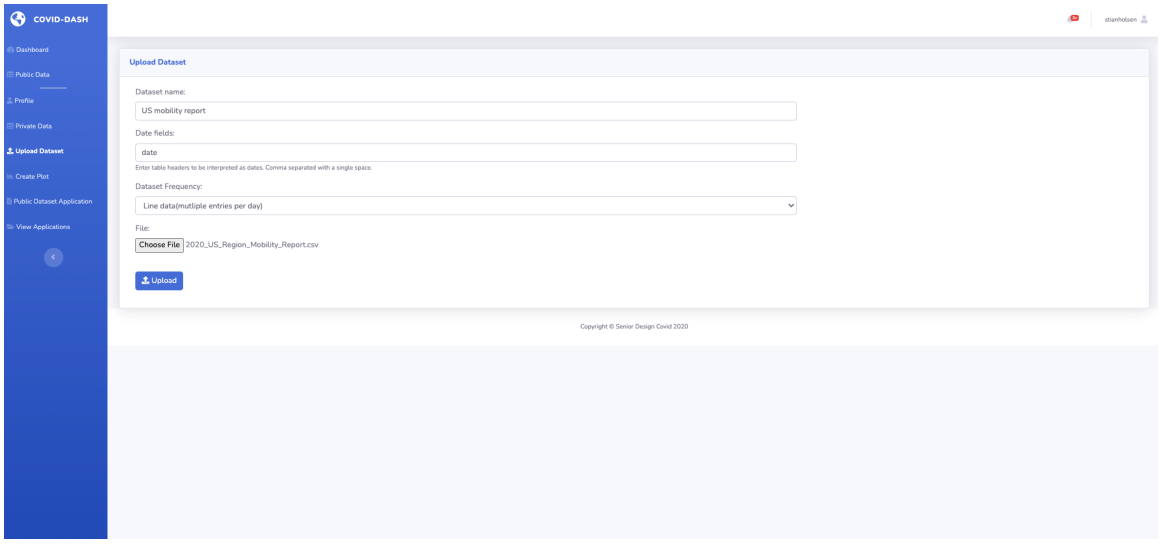
2.2. Login

The next step is to login. Users can click “Login“ on the top right corner of the website, from here, enter the username and password that was just created and you will be brought to the user dashboard.



3. Uploading a Dataset

Once logged in, a user can upload a dataset by first clicking the “Upload Dataset” tab on the sidebar to the left. From here, the user will be prompted to enter the dataset name, the date fields of the dataset, and a dataset frequency. Then users can upload the dataset file by selecting “Choose File” and then finally selecting “Upload.”



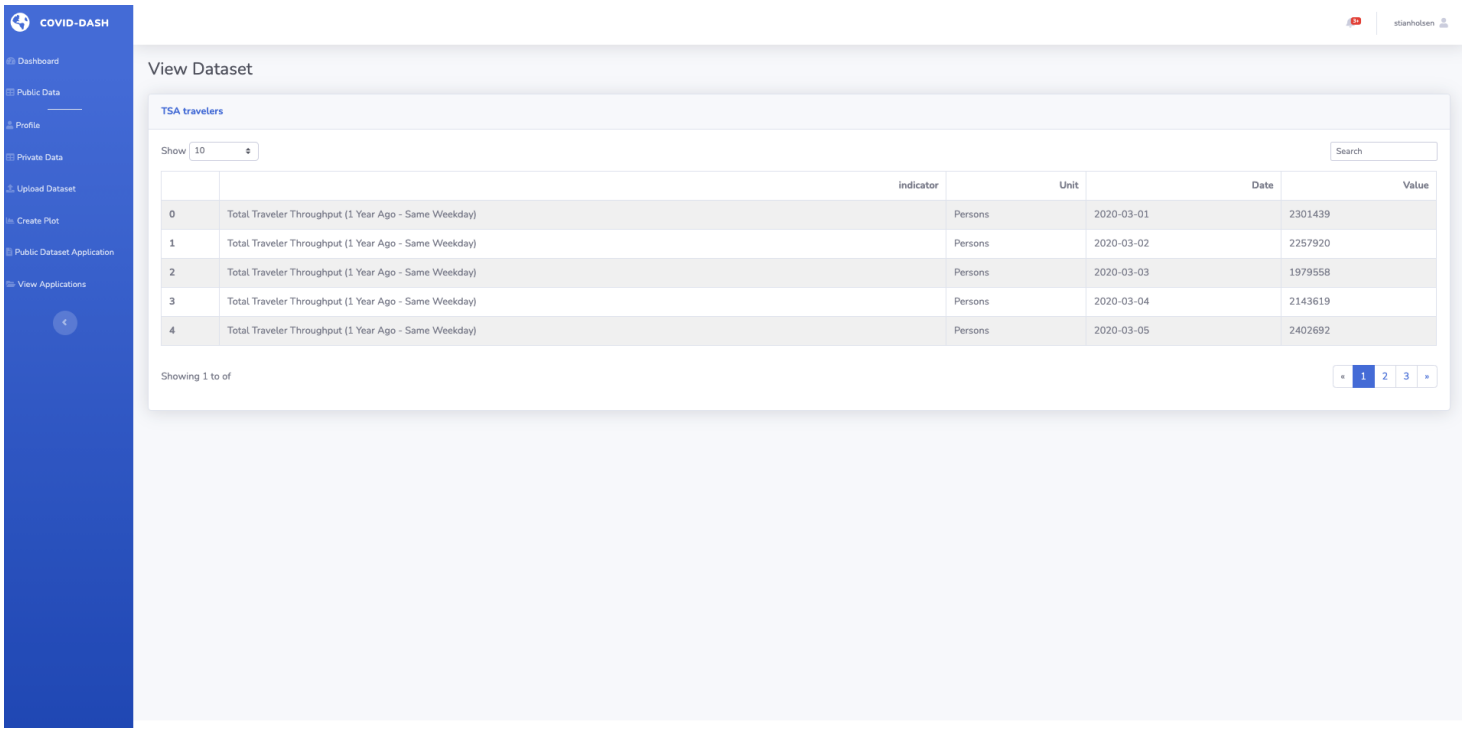
4. Viewing Data

4.1. Private Data

Once the user successfully uploads a dataset, the dataset can be found in the "Private Data" tab on the sidebar to the left. In the "Private Datasets" table on this page, a list of the user's uploaded datasets can be found. Click on the dataset of choice to be brought to the "View Dataset" page and from here a table of the dataset is displayed.

4.2. Public Data

If a user would like to make uploaded data public, they have the choice of uploading it as shared data, or submitting an application to get it approved by the admin, making it curated data. To make data public, users will select the "Public Dataset Application" tab on the sidebar to the left. From there, the user will be prompted to select their dataset, enter the dataset source, and enter a description of the source. If the user wants to have the dataset reviewed by the admin to become curated, they can click the "request review" button. The data will then be sent to the admin for review to be either approved or denied to become curated. If they do not want the dataset to be reviewed by the admin, they will not click the "Request Review" button and then click "Submit" to have the data uploaded as shared public data.



5. Create Plot

5.1. Select Plot

Users can create custom plots first by selecting the “Create Plot” tab on the sidebar to the left. This will bring you to the “Create Plot page.” In the “Select Plot” box, the user will first be prompted to enter the plot name alone with the plot type. For plot type, users can choose between trends over time, proportion among categories, relationships between possible factors and situations, and distributions over Florida counties.

5.2. Select Dataset and Variables

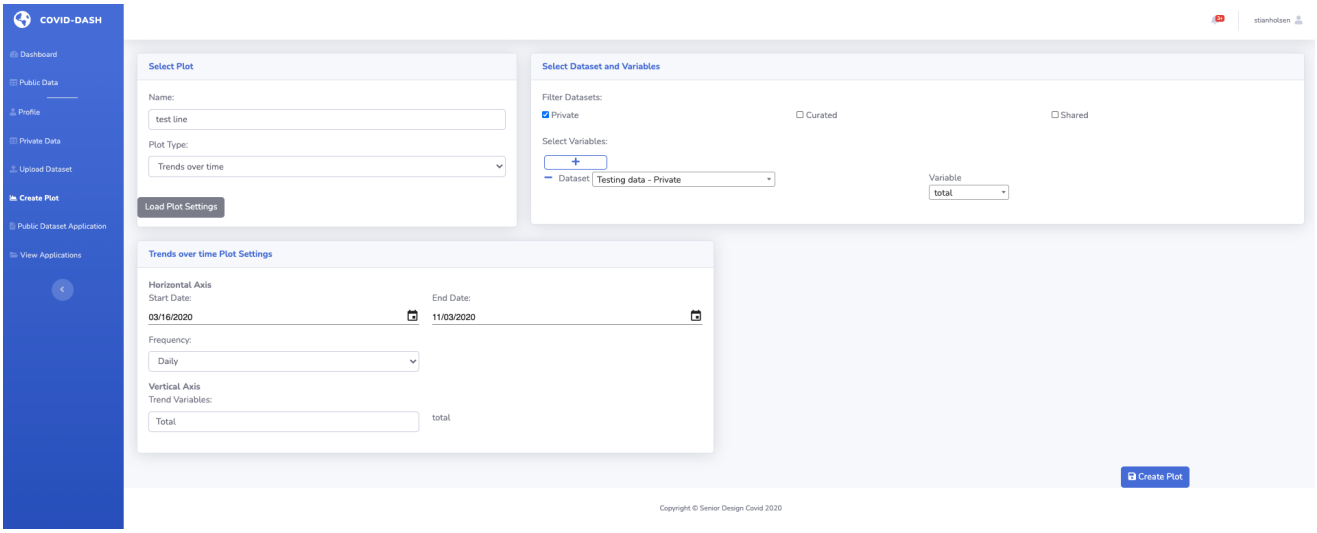
In the “Select Dataset and Variables” box, users will be prompted to choose which dataset and variables to use. Firstly, there is a filter for the types of data: shared, curated, and private, where users can filter the data type they want. From here, select the plus sign button to open up the options for datasets and variables. Under “Dataset” the user can select the dataset of their choice from the drop down list. Under “Variable” the user can select the variable they want to look at from the drop down list. To add more datasets and variables, the plus sign button can be selected again and more dataset and variable drop downs will be added for the user to select.

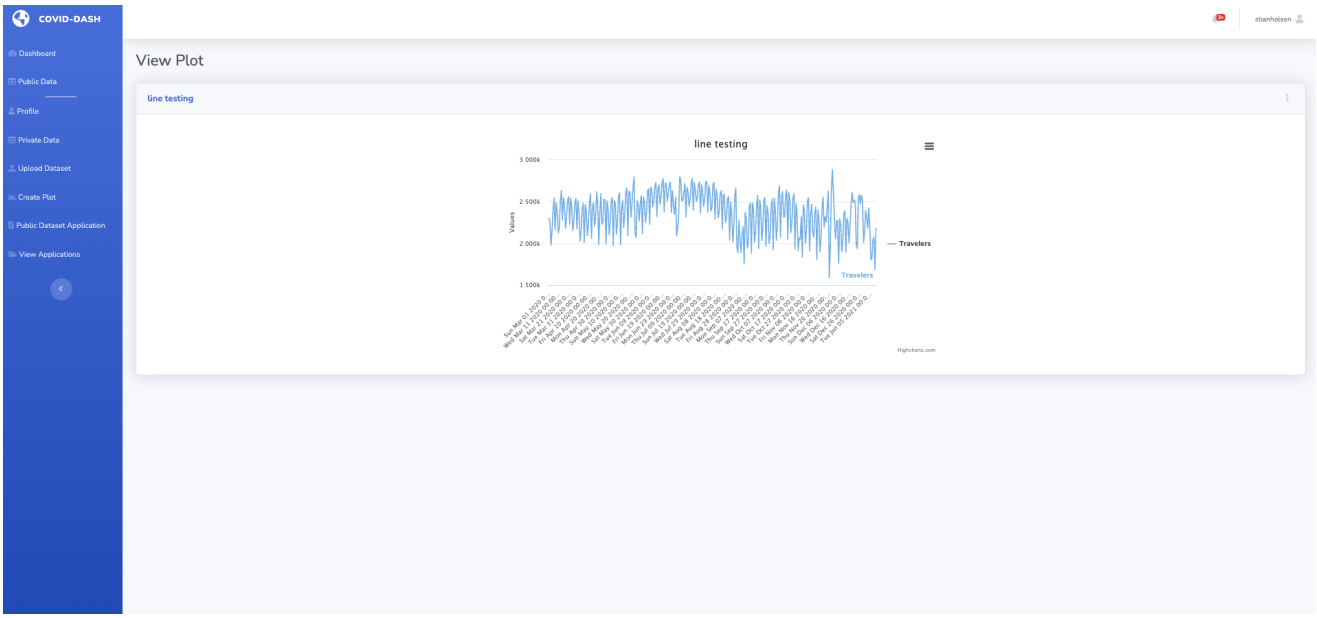
5.3. Plot Settings

Once the above actions are completed, select the “Load Plot Settings” button underneath the “Select Plot” box. Depending on the selected Plot Type, a plot settings card specific to the selected plot type will appear. Once the plot settings are entered, the user will push the “Create Plot” button in the bottom right corner to create the plot.

5.3.1. Trends over time

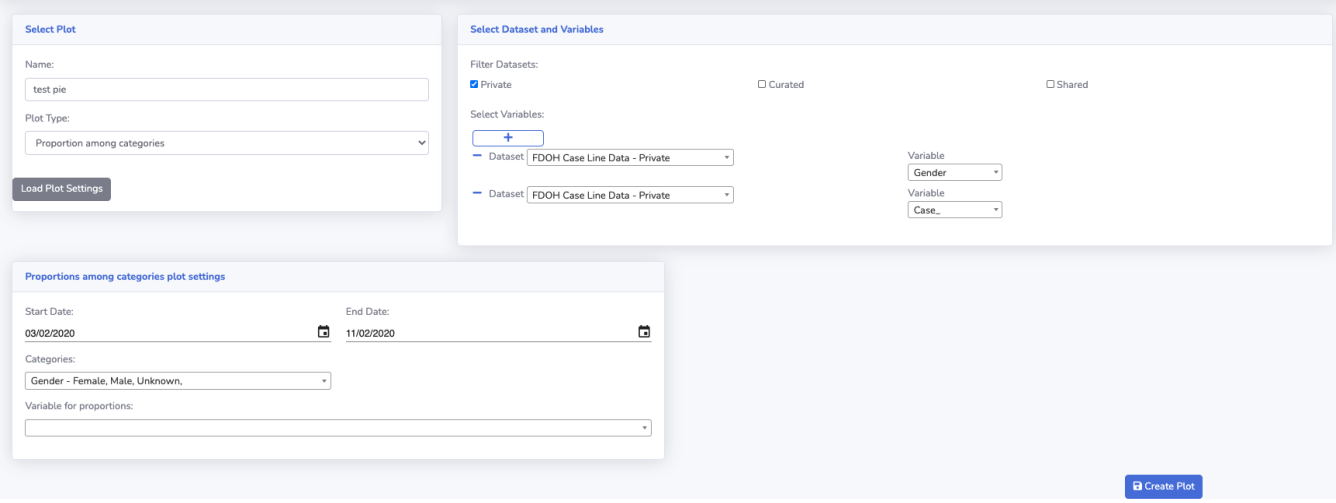
The “Trends over time” card allows users to filter by start date and end date, choose the sampling frequency, and choose the vertical axis labels for the chosen variables. Selecting this option produces what is technically known as a line graph. Multiple lines can be plotted. The date fields of each dataset will be outer joined to produce one shared time series dataset.

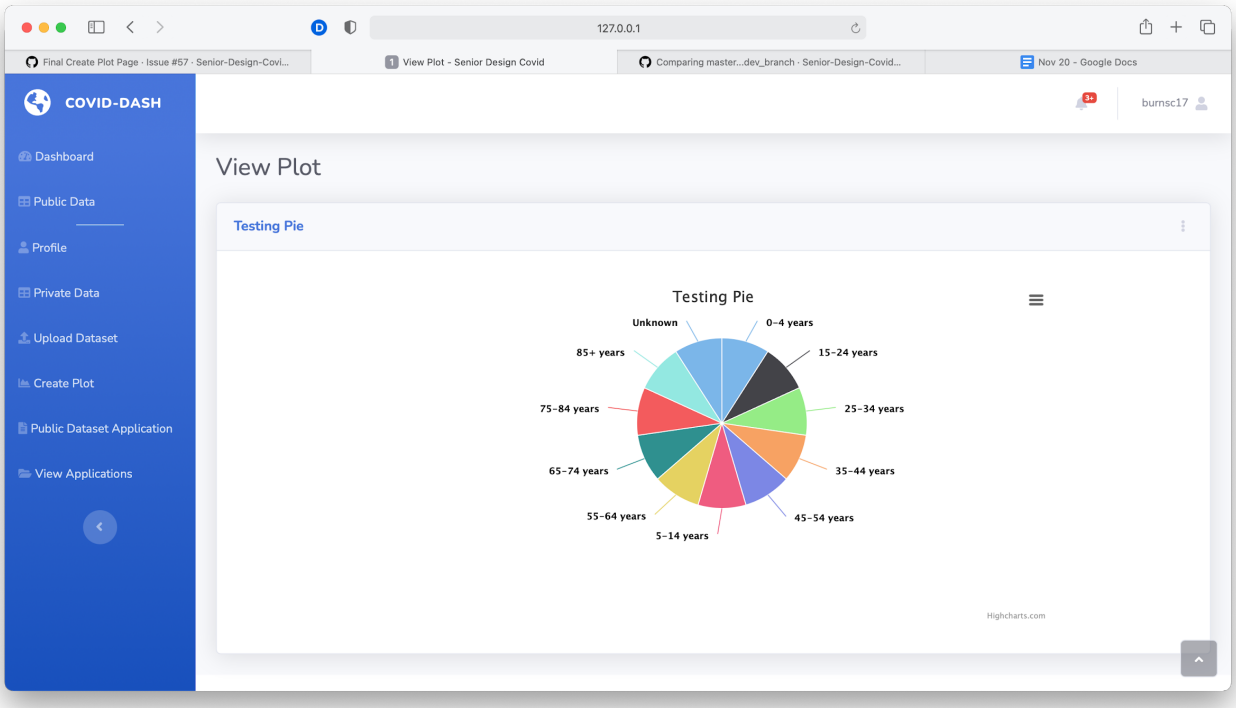




5.3.2. Proportions among categories

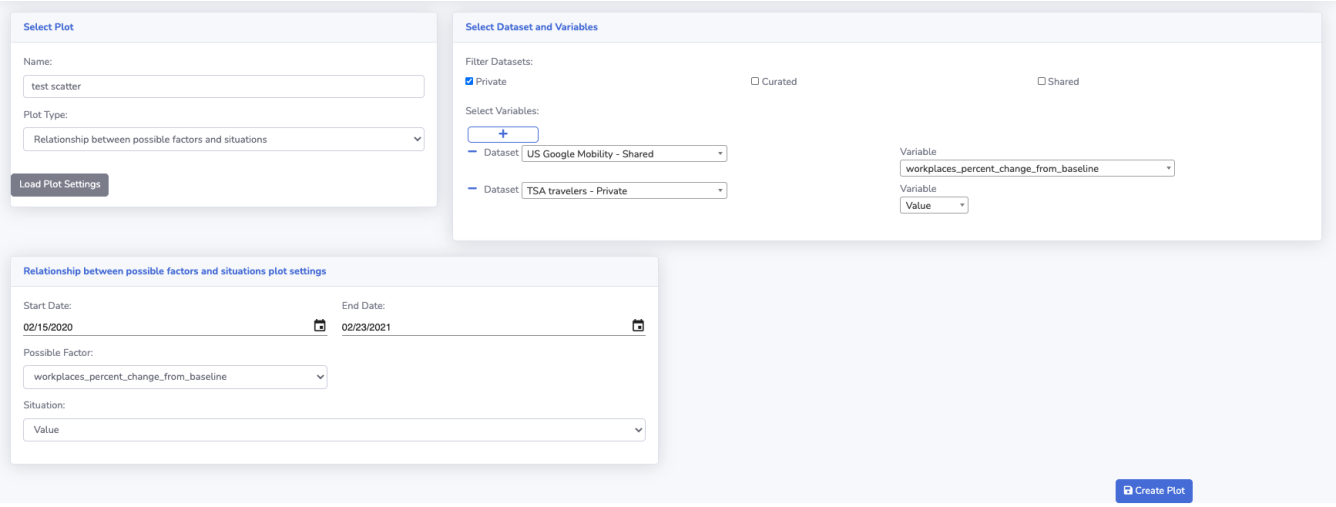
The “Proportions among categories” card allows users to filter by start date and end date, choose a category variable, and choose an aggregate variable. The category variable is how the data will be grouped. For example, to group a dataset by gender, the user should select gender as the category variable. The aggregate variable is what we will count. This card produces what is technically known as a pie chart.

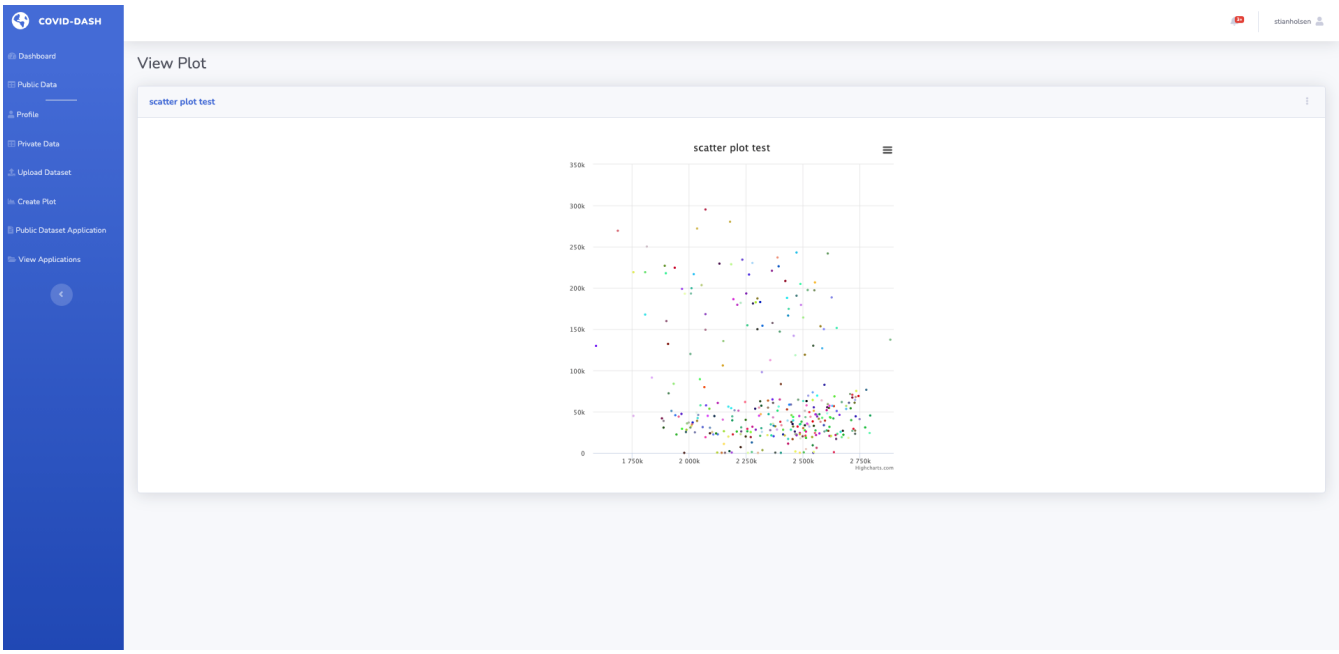




5.3.3. Relationship between possible factors and situations

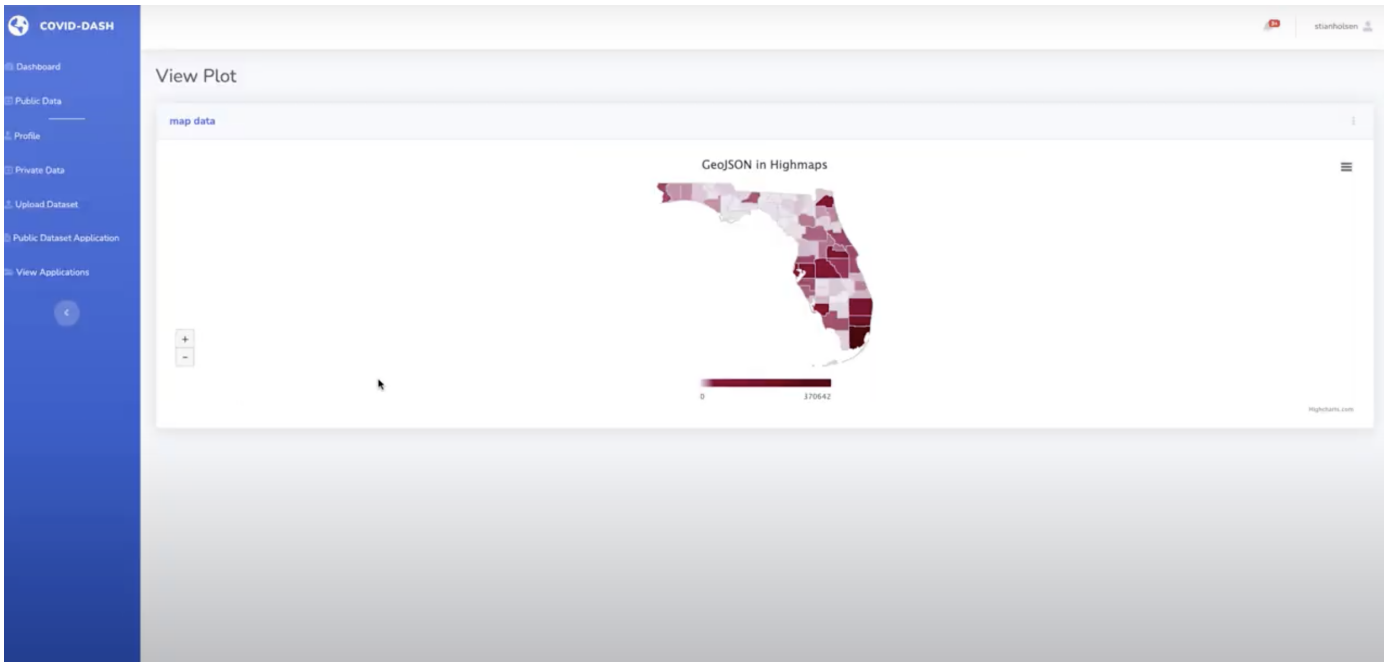
The “Relationship between possible factors and situations” card allows the user to filter the datasets by start date and end date, select a possible factor variable, and select a situation variable. The factor variable will display on the horizontal axis of the plot. The situation variable will display on the vertical axis of the plot. This settings card produces what is technically known as a scatter plot. This plot can be used to find correlations between variables.





5.3.4. Distribution across Florida counties

The “Distribution across Florida counties” card allows the user to select start and end date and one variable that will aggregate and group by counties. Datasets need to have county names and state names in order for our system to correctly match them to county codes. This settings card produces what is technically known as a distribution map or heat map.



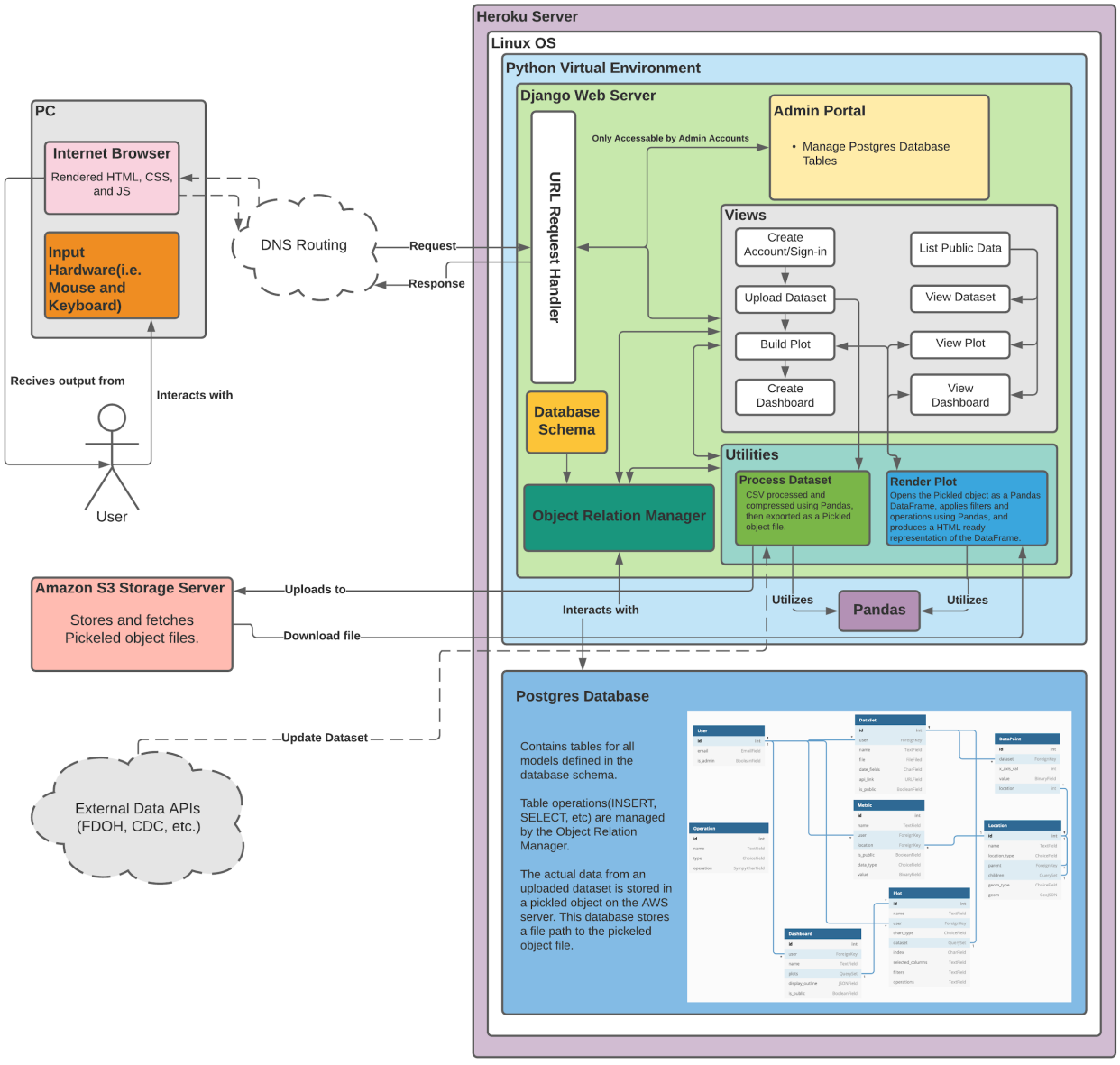
6. View Plot

Once a plot is created from the “Create Plot” page, a user will then want to view it. To do so, they will navigate to the “Private Data” page by selecting the tab on the left sidebar. From here, scroll down to the box labeled “Private Plots.” In this table, users can find their created plots. Click the plot of choice to be taken to the “View Plot” page. The plot will appear on the screen.

7. System Administrators

System administrators have control of the whole system. They can add, modify, and delete datasets, plots, and dashboards, along with reviewing applications for datasets to be made public.

8. System Architecture



9. Developers

Environment Setup

Directory Setup

Create a directory (named whatever you want). This will be used to hold the git repository and the python virtual machine.

cd into your newly created directory.

Python Virtual Machine

We need to create a python virtual machine. This is where we will install all of the necessary libraries to run the Django Project.

If you have python and pip installed, you can use venv.

> MacOS: Do *brew install pyenv* to get python versions other than whats supplied by XCode

To create the VM do, *python3.8 -m venv env*. This will create a VM named env.

You may need to do *pip install venv*.

MacOS: Activate/Deactivate VM
To activate the environment: *source env/bin/activate*

To deactivate at any time: *deactivate*

Windows: Activate/Deactivate VM

Go to the scripts folder and run *activate*.

cd .. back to the folder we created env in.

If the virtual environment activates correctly, you should see the name in (env) in your path.

You can verify your python version using *python -V*.

Geospatial Libraries and PostGIS

MacOS

Install PostgreSQL 12 using Postgres.app.

Install the following libraries using brew:

brew install postgresql
brew install postgis
brew install gdal
brew install libgeoip

Windows

Follow the Django Docs instructions for setting up PostgreSQL, PostGIS, OSGeo4W(GEOS, GDAL, PROJ), and Modifying the Windows Environment. Make sure to get PostgreSQL 12. Stop after you set your Windows Environment.

Clone Project from GitHub

cd to the directory that holds env. If you *ls*, you should get just the directory env

Clone the repository using: *git clone https://github.com/Senior-Design-CovidDash/CovidDashboard.git*

If you *ls* now you should get:

CovidDashboard env3.8

PostgreSQL Setup

1. Create a new database: *createdb django_project*
2. Enter the Postgres shell: *psql django_project*
3. Grant privileges to yourself: *grant all privileges on database django_project to yourusername*
4. Create Extension for PostGIS: *CREATE EXTENSION postgis;*
5. Create Extension for Raster: *CREATE EXTENSION postgis_raster;*
6. Exit psql shell: *\q*

Django Setup

cd into CovidDashboard. This is the git repository. You will start on the master branch. You can confirm this using *git branch* (q to quit).

Install requirements.txt

To install the necessary python packages(including Django), run *pip install -r requirements.txt*. You should see pip install all the packages and dependencies. Hopefully no errors.

The following steps involve creating files that set environment variables. These files have been added to the *.gitignore* so they will not be tracked by git.

MacOS: Create *environment.sh*

Create a new file at the same level as environment.sh.template. Copy and paste the contents of the template into the new file. Enter your username in the file and save.

To execute Django management commands, use *./local.sh <manage_command>*.

Windows: Create *local.bat*

Create a new file at the same level as local.bat.template. Copy and paste the contents of the template into the new file. Enter your username in the file and save.

To execute Django management commands, use *local.bat <manage_command>*.

Running the Local Server

If the installation worked correctly, you should be able to start a local server.

MacOS

./local.sh migrate
./local.sh runserver

Windows

local.bat migrate
local.bat runserver

If this works correctly you should see:

System check identified no issues (0 silenced).
September 29, 2020 - 02:00:40
Django version 3.1, using settings 'config.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
Open a web browser and go to http://127.0.0.1:8000/. There should be a Django Hello World.

Editor Settings

Set your editor to use **4 Spaces** instead of tabs.

Editor Recommendations

We recommend using Sublime Text. It has a good plugin for Django called Djaneiro. Install using the Package Manager in Sublime Text.

Sublime also makes a nice git manager called Sublime Merge.