Dwight Look College of
ENGINEERING
TEXAS A&M UNIVERSITY

ECEN 403 Final Presentation
Team 15: Wastewater Modeling

Team Members: Grace S , Ayaan S , Quinlon H , Matthew D

Sponsor: LANL

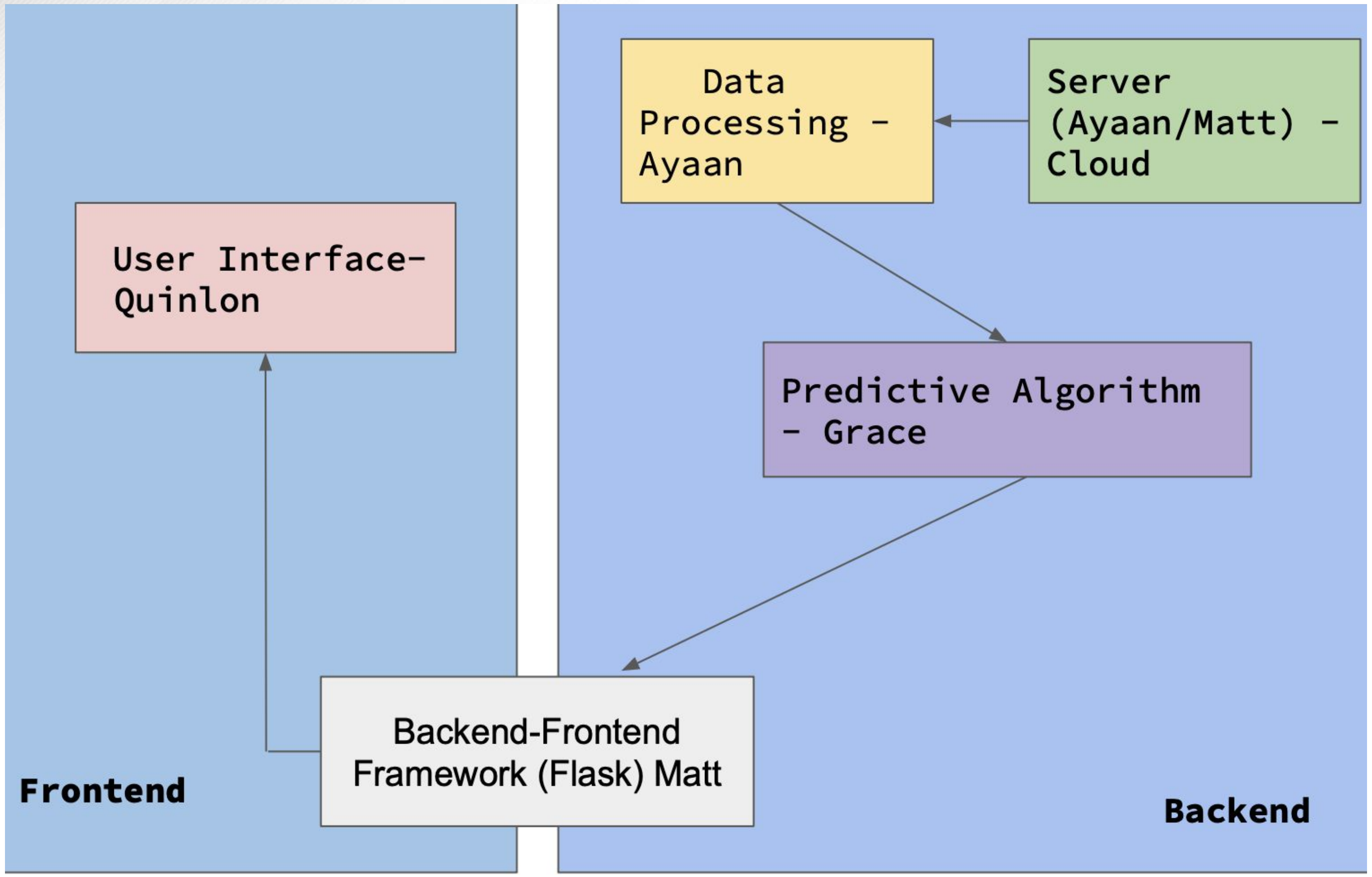
Problem Statement & Proposal

- Los Alamos National Labs
  - Currently limited in ability to make accurate predictions based on COVID-19 case data.
    - **Source Limitations** (gov/agencies)
    - **Privacy Concerns** (Individual data)
    - **Inconsistency** (availability, accuracy)
  - Need a **better method to make COVID-19 case projections** up to 1-2 weeks in advance.
    - Allows LANL to take necessary precautions

Solution Proposal

- "Utilize measurements of Covid-19 in wastewater data to develop a predictive algorithm that predicts case trends up to 1-2 weeks in advance, presented through a web app."

System Overview


Data Processing - Ayaan
Server (Ayaan/Matt) - Cloud
User Interface- Quinlon
Predictive Algorithm - Grace
Backend-Frontend Framework (Flask) Matt
Frontend
Backend

Dwight Look College of ENGINEERING TEXAS A&M UNIVERSITY

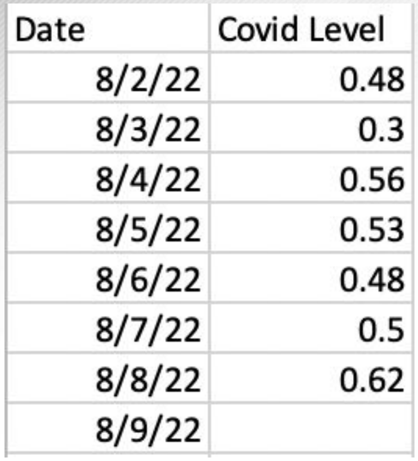
Data Processing/Analytics

Semester Accomplishments

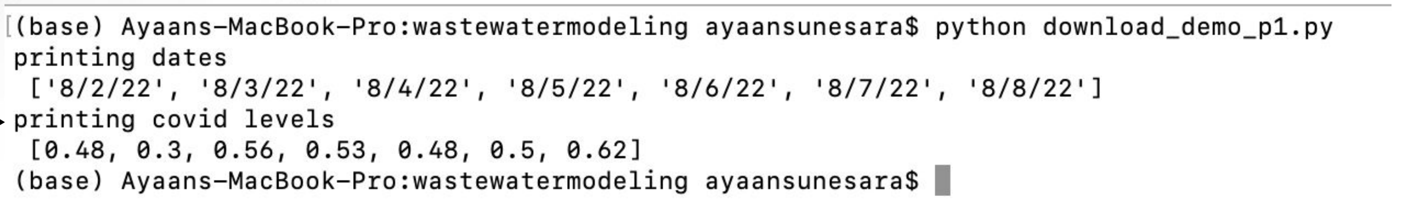
- AWS Server
  - Set up using S3 bucket and fully connected to the front end of the website
  - Save and upload files when pushed from front end
  - Download data files from the server via the history tab
  - Check if only a csv or xlsx file is uploaded, else throws an error

- Data Parsing
  - Separate dates and covid levels from csv files into two arrays to graph
  - Able to skip over data that aren’t filled in efficiently
  - Format the predictive algorithm output as JS objects to be graphed by the backend graphing system

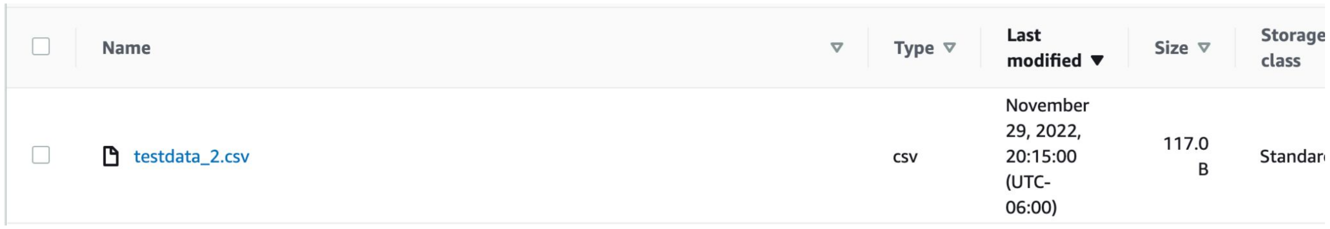
Data Processing/Analytics



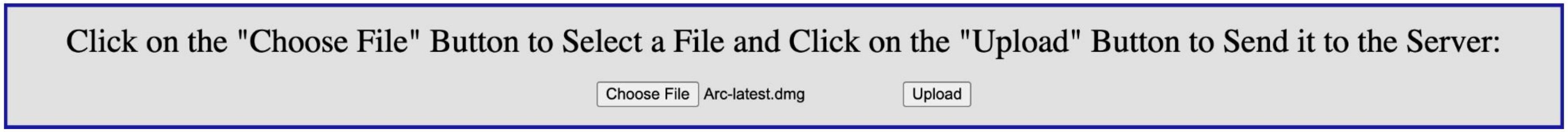
Date	Covid Level
8/2/22	0.48
8/3/22	0.3
8/4/22	0.56
8/5/22	0.53
8/6/22	0.48
8/7/22	0.5
8/8/22	0.62
8/9/22	



```
(base) Ayaans-MacBook-Pro:wastewatermodeling ayaansunesara$ python download_demo_p1.py
printing dates
 ['8/2/22', '8/3/22', '8/4/22', '8/5/22', '8/6/22', '8/7/22', '8/8/22']
printing covid levels
 [0.48, 0.3, 0.56, 0.53, 0.48, 0.5, 0.62]
(base) Ayaans-MacBook-Pro:wastewatermodeling ayaansunesara$
```

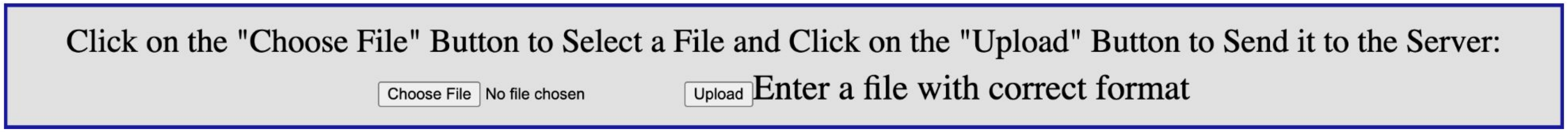


	Name	Type	Last modified	Size	Storage class
☐	testdata_2.csv	csv	November 29, 2022, 20:15:00 (UTC-06:00)	117.0 B	Standar



Click on the "Choose File" Button to Select a File and Click on the "Upload" Button to Send it to the Server:

Choose File Arc-latest.dmg Upload



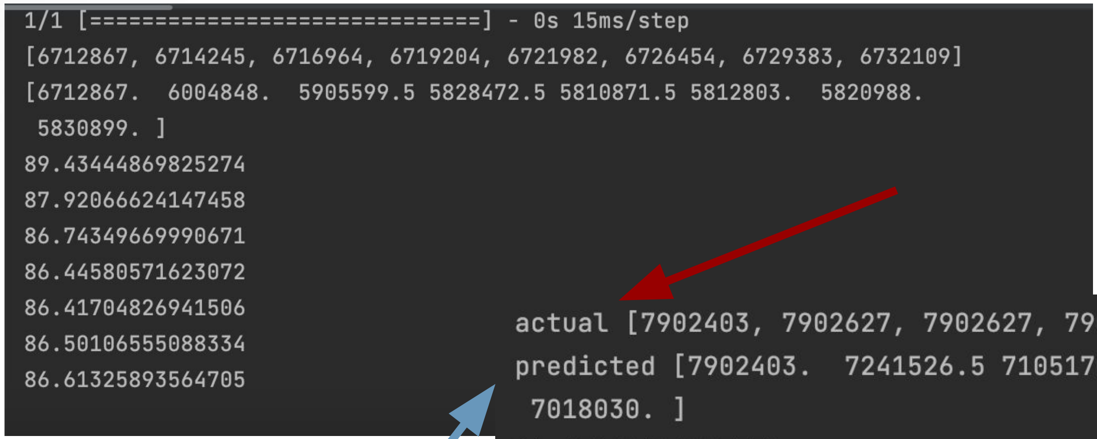
Click on the "Choose File" Button to Select a File and Click on the "Upload" Button to Send it to the Server:

Choose File No file chosen Upload Enter a file with correct format

Predictive Algorithm

Semester Accomplishments

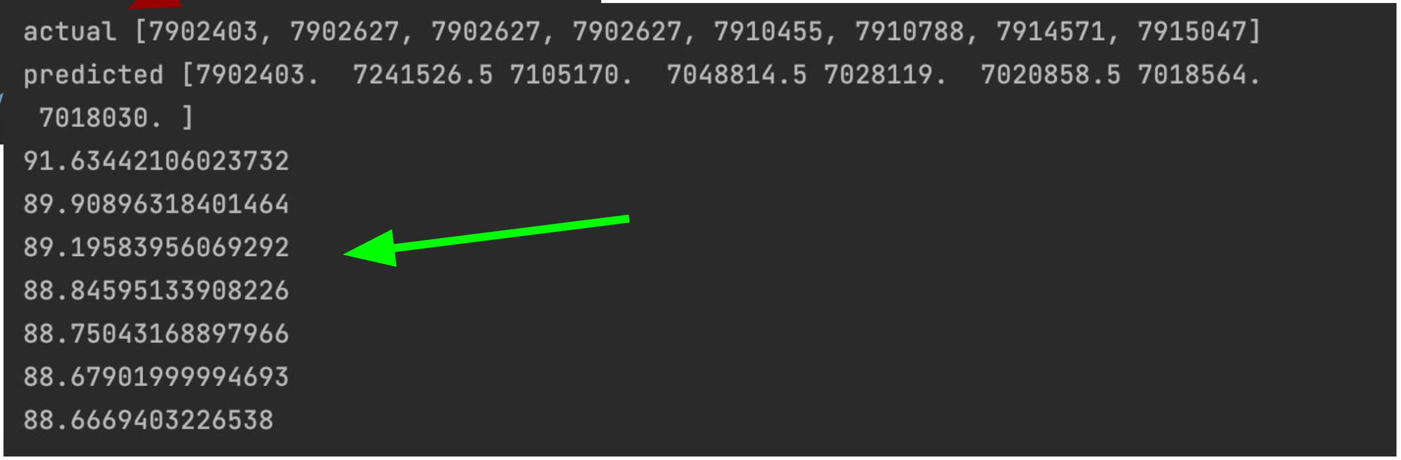
- Operating RNN/LSTM that can forecast predictions
- Built of using the covid cases in Texas supplied by CovidAct API but algorithm was tested on the approved wastewater data
- For validation used an updated csv file and withheld the last week of cases and predicted for that week & calculated percent accuracy



```
1/1 [==============================] - 0s 15ms/step
[6712867, 6714245, 6716964, 6719204, 6721982, 6726454, 6729383, 6732109]
[6712867.  6004848.  5905599.5 5828472.5 5810871.5 5812803.  5820988.
 5830899. ]
89.43444869825274
87.92066624147458
86.74349669990671
86.44580571623072
86.41704826941506
86.50106555088334
86.61325893564705
```

4/30-5/6

11/11-11/17

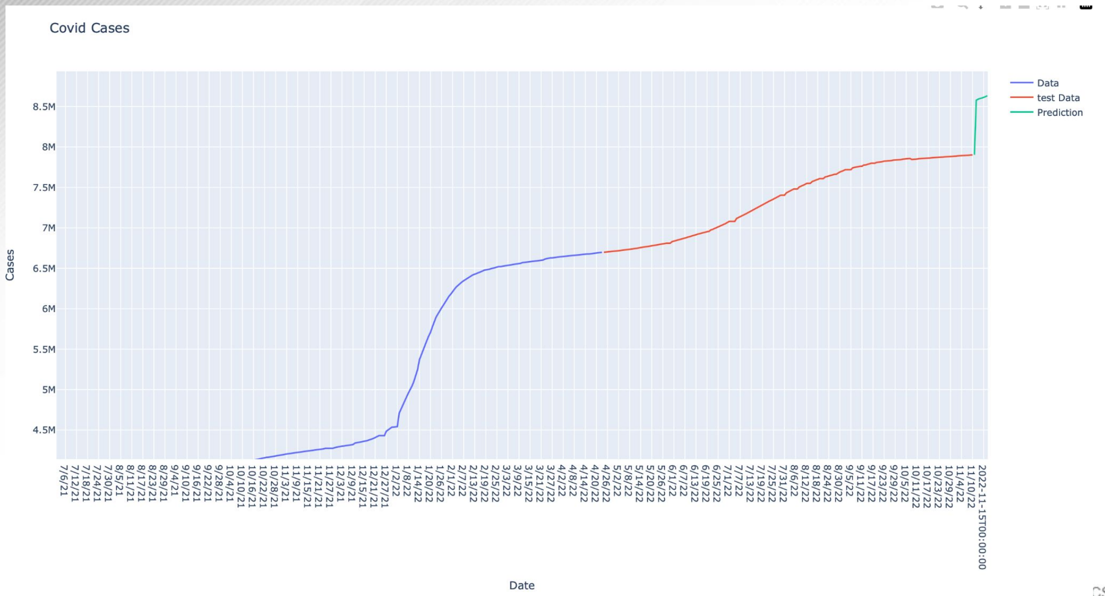


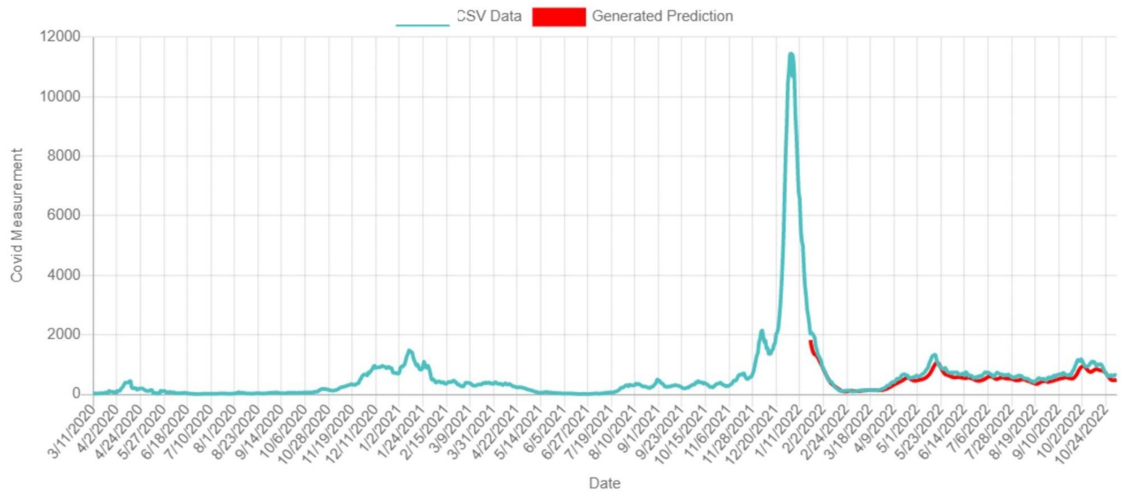
```
actual [7902403, 7902627, 7902627, 7902627, 7910455, 7910788, 7914571, 7915047]
predicted [7902403.  7241526.5 7105170.  7048814.5 7028119.  7020858.5 7018564.
 7018030. ]
91.63442106023732
89.90896318401464
89.19583956069292
88.84595133908226
88.75043168897966
88.67901999994693
88.6669403226538
```

Predictive Algorithm

Validation/Results

- Does work with processed wastewater data



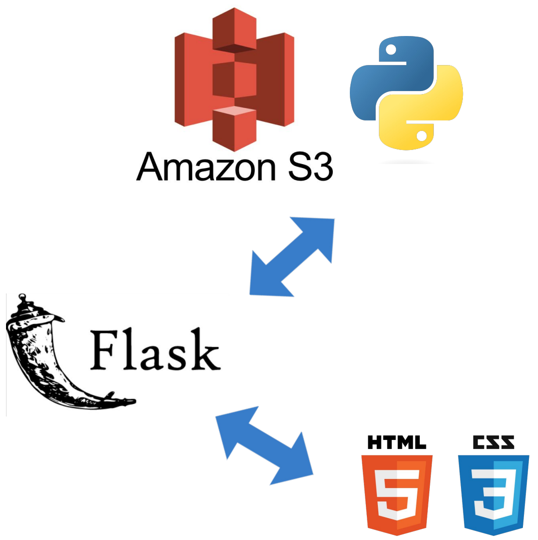


Frontend/Backend Integration

Semester Accomplishments - Matthew D.

Fully integrated Flask microframework

- **Routing** - ensures all user requests from UI perform correct operations through connections to correct Python procedures and servers.
    - Ex: Functionality behind buttons/tabs
- **Templating -** ensures the backend data results/variables can be utilized and displayed to the UI when accessed



Dynamic Chart Generation (Chart.js)

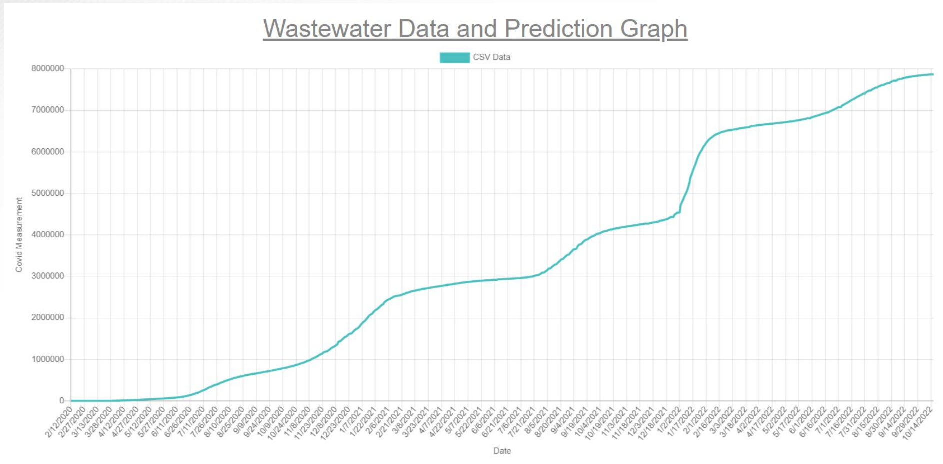
- Utilizes Chart.js to display a graph of the user submitted data and prediction model results (csv upload).
    - **Initial Generation**
        - Takes in user data objects (Python)
        - routes to HTML through Flask
        - utilizes Javascript/Chart.js/Jinja to display data at correct position.
    - **Update w/ Prediction**
        - Takes output of Pred. Algo, regenerates w/ additional line on graph for prediction data.

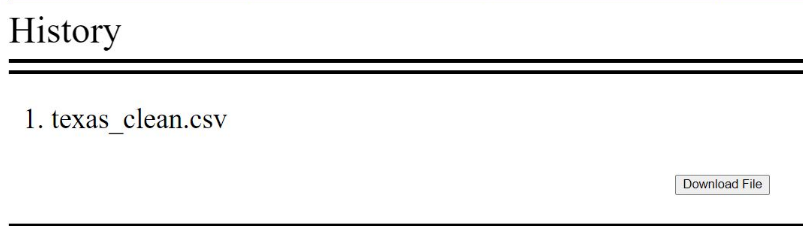
Publicly Hosted - Heroku

- Utilized virtual environments to maintain version control of packages
- Currently running with certain limitations in memory and RAM

Frontend/Backend Integration

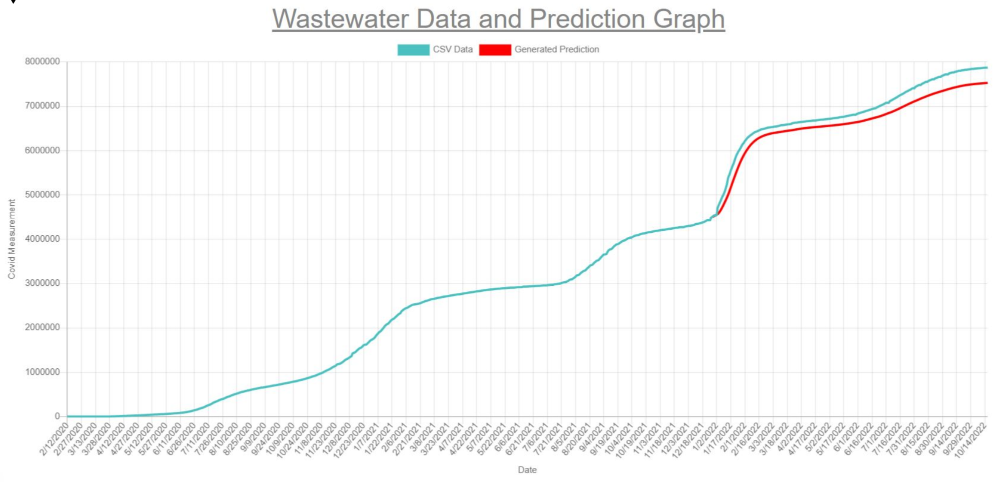
Validation/Results - images below verify that all connection components of routing/templating, upload/generate buttons, Graph.js, and dynamic updating of webpage data (Graph, history) are functioning correctly.

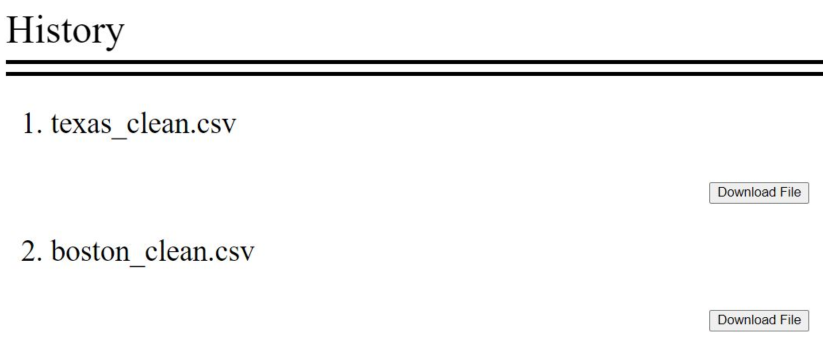




History

1. texas_clean.csv





History

1. texas_clean.csv

2. boston_clean.csv

Dwight Look College of ENGINEERING TEXAS A&M UNIVERSITY

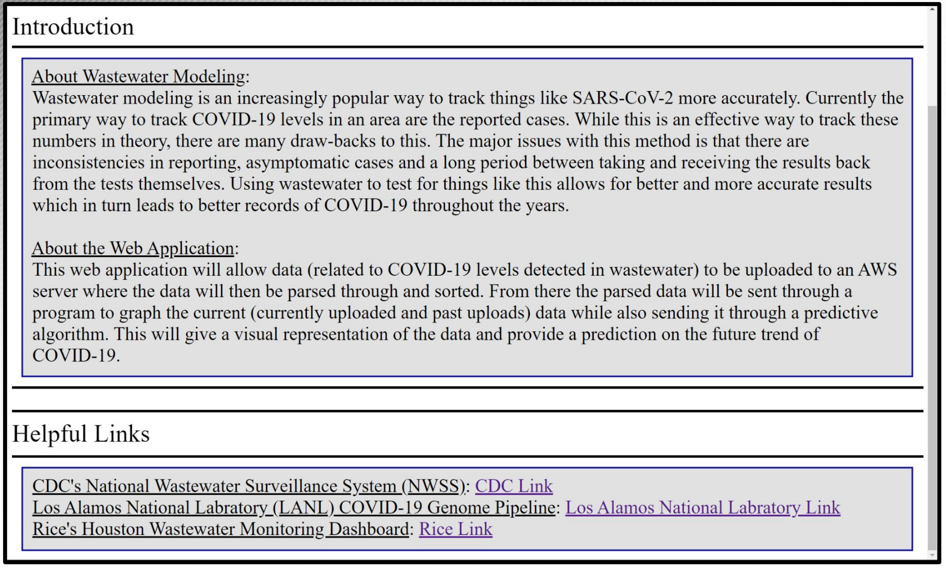
User Interface Design

Semester Accomplishments

- Title Block - This is the overall structure of the website (title, names and tab structure) and remains the same in each tab. (Tested and Validated)
- Functional Tabs - Allows the user to traverse web application.
    - Tabs include Home, File Upload, Graphs & Data, History, Github link and TAMU link. (Tested and Validated)
- File Upload Button - Button that you can be clicked on to select a file from device that then sends file to the AWS server. (Tested and Validated)
- Graph - Graph and prediction that is generated after a file is uploaded. (Tested and Validated)

Dwight Look College of ENGINEERING TEXAS A&M UNIVERSITY

User Interface Design



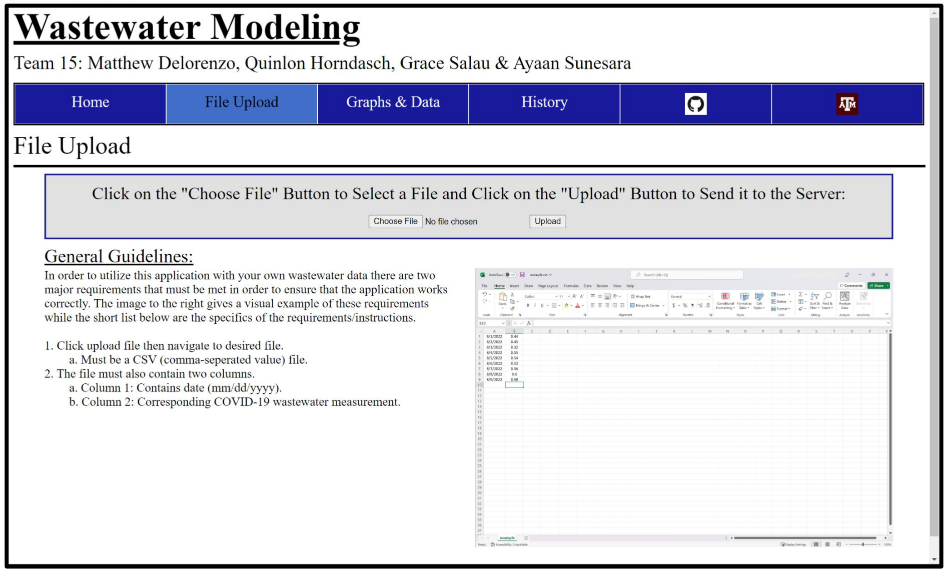
Introduction

About Wastewater Modeling:
Wastewater modeling is an increasingly popular way to track things like SARS-CoV-2 more accurately. Currently the primary way to track COVID-19 levels in an area are the reported cases. While this is an effective way to track these numbers in theory, there are many draw-backs to this. The major issues with this method is that there are inconsistencies in reporting, asymptomatic cases and a long period between taking and receiving the results back from the tests themselves. Using wastewater to test for things like this allows for better and more accurate results which in turn leads to better records of COVID-19 throughout the years.

About the Web Application:
This web application will allow data (related to COVID-19 levels detected in wastewater) to be uploaded to an AWS server where the data will then be parsed through and sorted. From there the parsed data will be sent through a program to graph the current (currently uploaded and past uploads) data while also sending it through a predictive algorithm. This will give a visual representation of the data and provide a prediction on the future trend of COVID-19.

Helpful Links

CDC's National Wastewater Surveillance System (NWSS): CDC Link
Los Alamos National Labratory (LANL) COVID-19 Genome Pipeline: Los Alamos National Labratory Link
Rice's Houston Wastewater Monitoring Dashboard: Rice Link



Wastewater Modeling

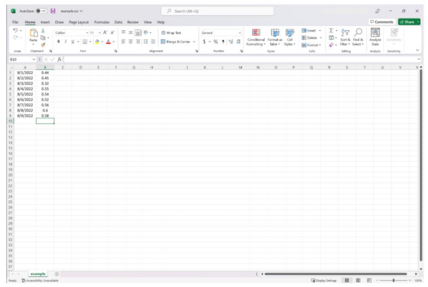
Team 15: Matthew Delorenzo, Quinlon Horndasch, Grace Salau & Ayaan Sunesara

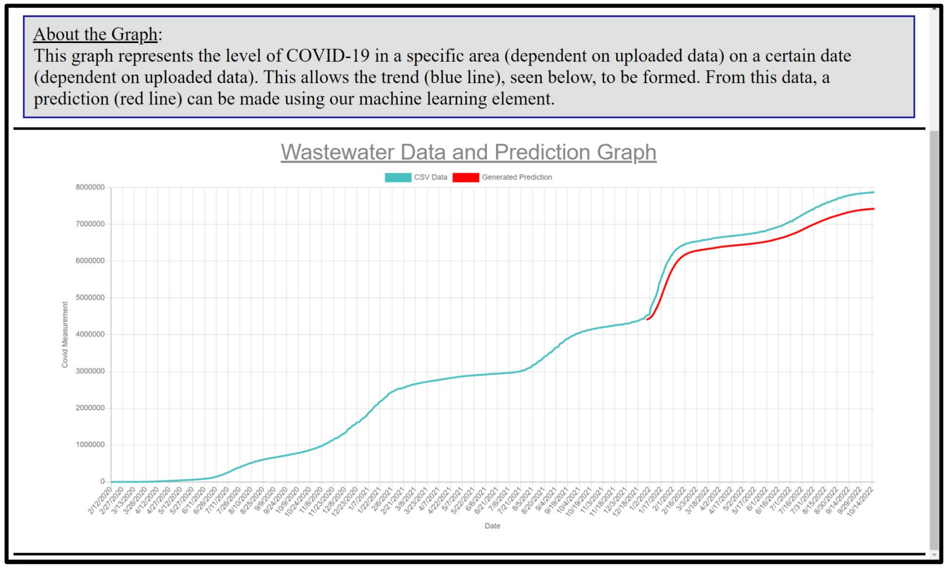
File Upload

Click on the "Choose File" Button to Select a File and Click on the "Upload" Button to Send it to the Server:

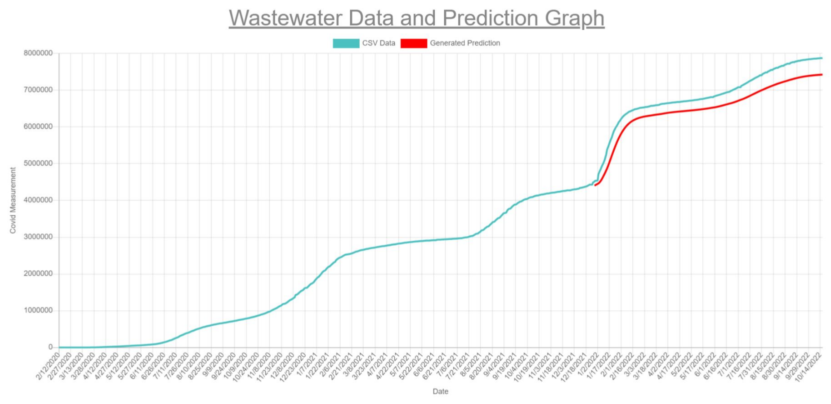
General Guidelines:
In order to utilize this application with your own wastewater data there are two major requirements that must be met in order to ensure that the application works correctly. The image to the right gives a visual example of these requirements while the short list below are the specifics of the requirements/instructions.

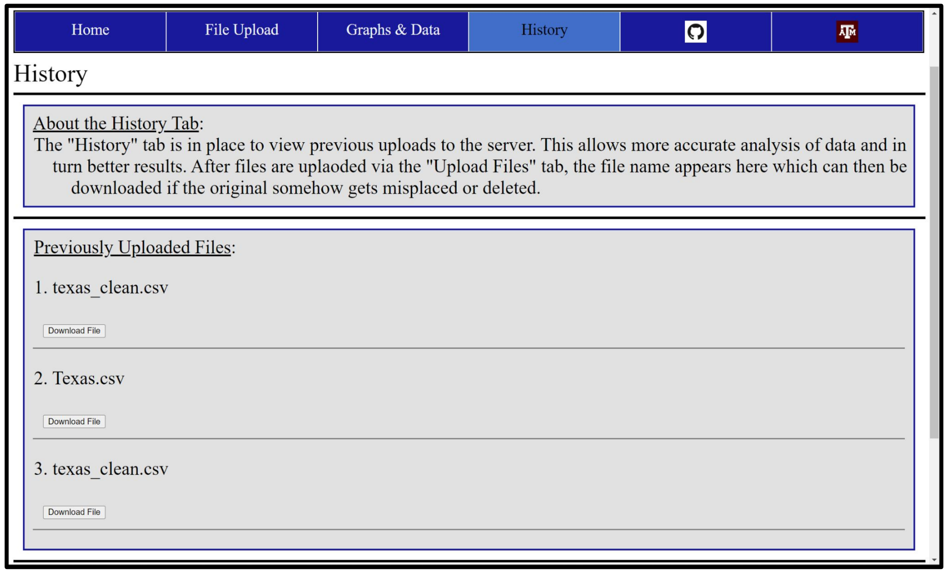
1. Click upload file then navigate to desired file.
   a. Must be a CSV (comma-seperated value) file.
2. The file must also contain two columns.
   a. Column 1: Contains date (mm/dd/yyyy).
   b. Column 2: Corresponding COVID-19 wastewater measurement.





About the Graph:
This graph represents the level of COVID-19 in a specific area (dependent on uploaded data) on a certain date (dependent on uploaded data). This allows the trend (blue line), seen below, to be formed. From this data, a prediction (red line) can be made using our machine learning element.





History

About the History Tab:
The "History" tab is in place to view previous uploads to the server. This allows more accurate analysis of data and in turn better results. After files are uplaoded via the "Upload Files" tab, the file name appears here which can then be downloaded if the original somehow gets misplaced or deleted.

Previously Uploaded Files:

1. texas_clean.csv

2. Texas.csv

3. texas_clean.csv

Dwight Look College of ENGINEERING TEXAS A&M UNIVERSITY

Validation Plan

Validation Plan					
FSR tie in	Test name	Success Criteria	Methodology	Status	Responsible Engineers
3.2.1.1	Testing Functionality of Predictive algorithm and having a 50% completed algorithm	Runs without error, takes in csv file data of various sizes, and outputs sample graph	Testing corner cases as well as fundamental cases in the algorithm with a data set/data that's reliable	passed	Grace
3.2.1.1	85% completed algorithm with intermediate testing	Outputted predictive trend has required accuracy as well as runs without error	Using a dataset that I have deemed as reliable and running the algorithm and validating the trend against what the trend was	Passed	Grace
3.2.1.1	100% completed algorithm testing wastewater data and validation comparison	Outputted predictive trend runs with using covid related dataset of any size	Using a dataset that i have deemed as reliable and running the algorithm and validating the trend against what the trend was	Passed	Grace
3.2.1.1	Fully functional algorithm outputs past and future data with 85% accuracy	Outputted predictive trend has an accuracy of 85% & outputs graph with necessary data	Using a dataset and inputting more training and testing data and running the algorithm and comparing the outputted trend against what happened	Passed	Grace

Validation Plan

3.2.1.3	Testing functionality of tabs in the web application	Each tab correctly directs user to the specified part of the website (ie. Home tab brings user to the homepage)	Open the web application and click on each of the tabs to ensure the tabs correctly direct the user	Tested	Quinlon
3.2.1.3	File upload function uploads data to AWS server	The user uploads the data (CSV) and the data gets properly uploaded onto the AWS server	Using the file upload button, a dataset will be uploaded. Next the AWS server will be checked to ensure the data was stored correctly	Teested	Quinlon
3.2.1.3	Graphs properly show the uploaded data trend and predictive algorithms prediction	The uploaded data is properly shown visually through graphs along with the predictive trend within the web application	Open the web application, click on the upload files tab and upload a dataset that has a known trend then click on the graphs and data tab to ensure the correct data is presented	Tested	Quinlon
3.2.1.3	History tab shows past uploads, graphs and datasets	When history tab is clicked and loaded, the previously uploaded datasets and associated graphs are visible	Open the web application, click on the upload files tab and upload a multiple datasets then click on the history tab to ensure that there are multiple entries and that the data is viewable	Tested	Quinlon

Validation Plan

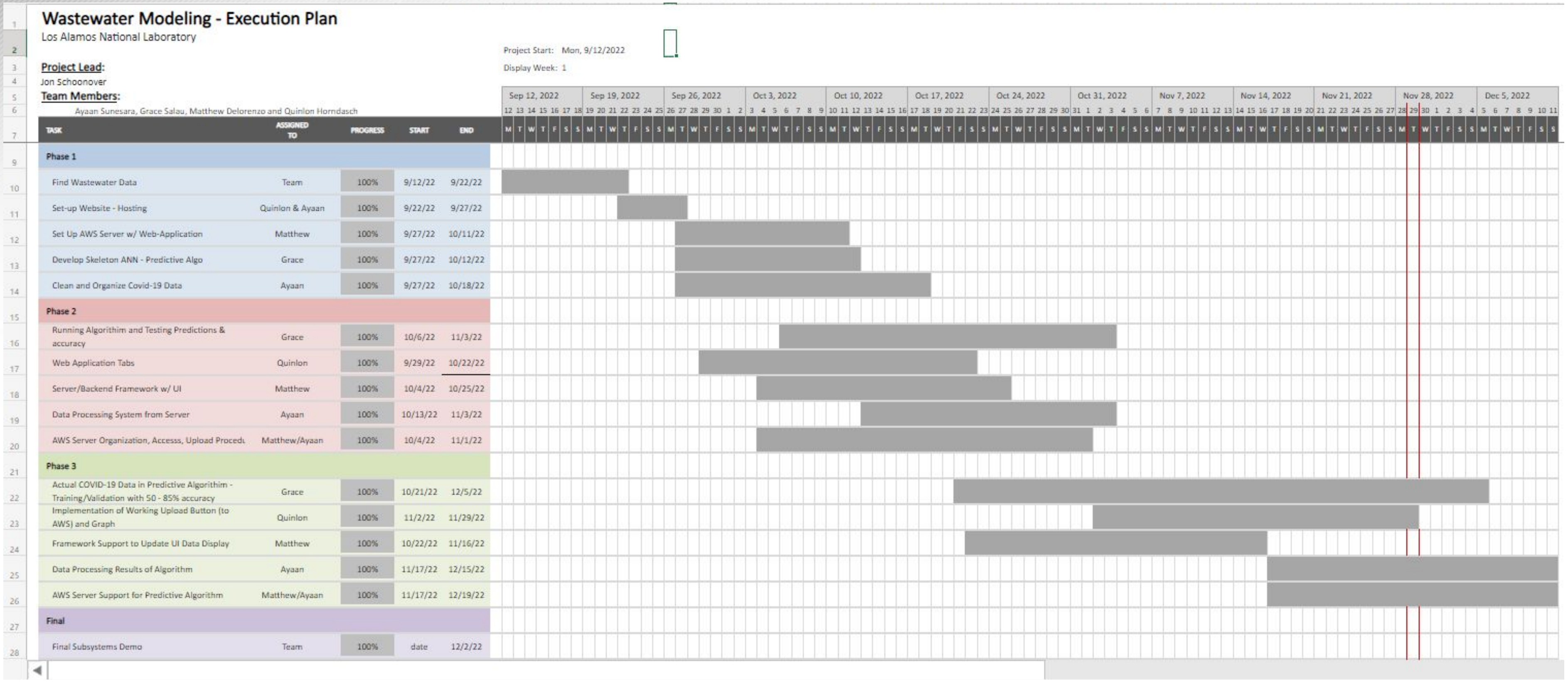
8.2.2.3	Able to connect the web app backend data to the frontend.	A variable declared in a Python file is able to be displayed on the UI of the web-app.	Utilize Flask templating to pass the variable from the Python file into a predefined portion of HTML code - test if correct value is then displayed	Validated	Matthew
8.2.2.4	Data values from the server are able to be displayed.	A variable derived from the AWS Server is able to be displayed on the UI of the web-app	Find the variable stored in AWS from the server, utilize Flask templating and routing in the file to pass the variable to a predefined HTML code portion	Validated	Matthew
8.2.2.4	Able to push large data constructs (graphs/projections) to UI.	A larger chunk of data (such as an array variable for a graph) is able to be displayed to the UI	Utilize Flask to take the data structure through templating/routing, and loop through the values to display them in the HTML side	Validated	Matthew
8.2.2.4	Can refresh presented data with an updated version when specified.	When a button is pushed on the UI, the backend is able to push the requested data to the frontend	When a button is pushed, triggers backend data processing to update, Flask utilized as before to push new data to the UI.	Validated	Matthew
8.2.2.4	Can transfer user input data from the UI to the server.	User is able to input data in the UI that is able to be input in the data server.	Use Flask/HTML to grab the UI data, place it in the Python code, then deposit it into the server data repository correctly.	Validated	Matthew

Dwight Look College of ENGINEERING TEXAS A&M UNIVERSITY

Validation Plan

3.2.2.4	Downloading Data from Example Files to Analyze	Able to grab data from the excel sheets and create an array out of it from the server	Use an example data file to represent the waste water data LANL will give to figure out if AWS Quicksight can read data values from excel sheet	Validated	Ayaan
3.2.2.4	Parse Useful Data and Save as New File	Remove unnecessary data from the array created to have the data and covid values only	Be able to parse the data grabbed from the excel sheets and be able to distinguish required data to save separately. (only covid and date data can stay)	Validated	Ayaan
3.2.2.4	Create Graphs and Save it on AWS Server	Plot the trend on a graph and export it back to AWS server to use	Using matplotlib library in Python to plot the x and y values, creating a graph showing a trendline	Validated	Ayaan
3.2.2.3	Check uploaded file to be the correct format that can be parsed	Error message is thrown in the upload file tab when an incorrect file type is uploaded and the file should not be saved on the server	Parsing the input file name and the extension of the file to check if the extension is either a csv or xlsx file. If the file doesn’t match the extension, it will not be saved in the server and an error message will show up on the front end	Validated	Ayaan

Execution Plan



Wastewater Modeling - Execution Plan

Los Alamos National Laboratory

Project Lead:
Jon Schoonover

Team Members:
Ayaan Sunesara, Grace Salau, Matthew Delorenzo and Quinlon Horndasch

Project Start: Mon, 9/12/2022
Display Week: 1

Sep 12, 2022 | Sep 19, 2022 | Sep 26, 2022 | Oct 3, 2022 | Oct 10, 2022 | Oct 17, 2022 | Oct 24, 2022 | Oct 31, 2022 | Nov 7, 2022 | Nov 14, 2022 | Nov 21, 2022 | Nov 28, 2022 | Dec 5, 2022

TASK	ASSIGNED TO	PROGRESS	START	END
Phase 1				
Find Wastewater Data	Team	100%	9/12/22	9/22/22
Set-up Website - Hosting	Quinlon & Ayaan	100%	9/22/22	9/27/22
Set Up AWS Server w/ Web-Application	Matthew	100%	9/27/22	10/11/22
Develop Skeleton ANN - Predictive Algo	Grace	100%	9/27/22	10/12/22
Clean and Organize Covid-19 Data	Ayaan	100%	9/27/22	10/18/22
Phase 2				
Running Algorithim and Testing Predictions & accuracy	Grace	100%	10/6/22	11/3/22
Web Application Tabs	Quinlon	100%	9/29/22	10/22/22
Server/Backend Framework w/ UI	Matthew	100%	10/4/22	10/25/22
Data Processing System from Server	Ayaan	100%	10/13/22	11/3/22
AWS Server Organization, Accesss, Upload Procedu	Matthew/Ayaan	100%	10/4/22	11/1/22
Phase 3				
Actual COVID-19 Data in Predictive Algorithim - Training/Validation with 50 - 85% accuracy	Grace	100%	10/21/22	12/5/22
Implementation of Working Upload Button (to AWS) and Graph	Quinlon	100%	11/2/22	11/29/22
Framework Support to Update UI Data Display	Matthew	100%	10/22/22	11/16/22
Data Processing Results of Algorithm	Ayaan	100%	11/17/22	12/15/22
AWS Server Support for Predictive Algorithm	Matthew/Ayaan	100%	11/17/22	12/19/22
Final				
Final Subsystems Demo	Team	100%	date	12/2/22

Post Demo - 404 Plans

Data Processing/Server (404):

- Parsing and processing inputs and outputs of files for different predictive algorithms
- Organize AWS server so files saved for a period of time delete
- Continue to clean and parse new data we receive such as from Los Alamos

Frontend/Backend Integration (404):

- **Continue expanding graph displays as backend developments continue**
  - (new models, prediction lines, visuals)
- **Ensure correct routing with additional user requests, posts**
  - Buttons, arguments, specifications.
  - More user customizability in prediction generation.
- **Maintain publicly hosted webpage**
  - Currently sufficient with Heroku
  - Research potential alternatives if needed

Post Demo - 404 Plans

Predictive Algorithm:

- Integrating my code all in one cohesive program for backend, validate that it works and displays as expected on the frontend (403/404)
- Adding **additional Models (naive model)** (404)
- Utilize **data from Los Alamos** to tailor algorithm/models (404)

User Interface:

- Specify location of download in description. (403)
- Develop Error Handling - ex: (CSV formatting), pressing the “Generate Prediction” button multiple times. (403)
- Allow for different viruses to be tested. (404)