1. Introduction

Introduce your product.

Our product is a web application that uses data tables to display a heat map that shows the frequency of COVID-19 cases in prisons around the United States. Our program uses an interactive map that changes based on what is selected to be displayed. There are three sets of data, county data, state data, and data for prisons. This will change depending on what date the user selects for the map to show. The map is color coded with the darker colors indicating a higher number of COVID-19 cases. The prisons are denoted by a marker, which when hovered over, displays the number of cases. Further data can be shown if the user decides to display the data tables for each state/county/prison. These can be toggled on or off and shows the cases and deaths for each state/county/prison.

2. System Overview

2.1 Background

The system administrator will need to get the required dependencies installed onto all the systems and the utilities needed to run the software as well. They will need to maintain the database by updating or extending the database with new information to keep the information required for the software up-to-date. They will also need to make sure that there is a backup of the software incase of an error or the database gets wiped to keep the day-to-day operations running.

2.2 Hardware and Software Requirements

The hardware required is just a working computer with enough power to run a simple web app. The software needed includes node.js and MySQLWorkbench.

- Processor: 2 gigahertz (GHz) or faster processor or SoC
- RAM: 4 gigabyte (GB) for 32-bit or 8 GB for 64-bit
- Hard disk space: 32 GB for 32-bit OS or 64 GB for 64-bit OS
- Graphics card: DirectX 9 or later with WDDM 1.0 driver
- Display: 800 x 600
- OS: Win10, Linux, MacOS

3. Administrative Procedures

3.1 Installation

(node.js, npm, mysql workbench)

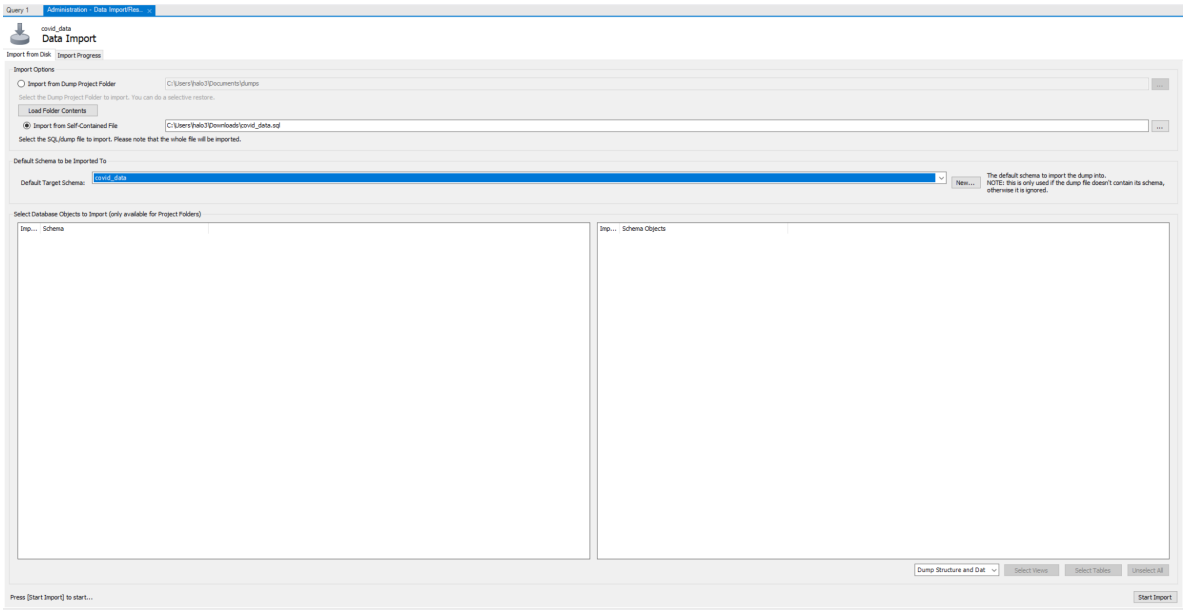
To install the web app, you must first install MySQLWorkbench. After this is installed, you must open up a MySQL80 server instance on MySQLWorkbench. Also install Node.js.

Step 1:

Download the Zip file or clone github directory into your preferred destination folder.
https://github.com/CMSC447/finalproject

Step 2:

Take the sql and import it into mysql workbench to create the database required for the software to show data.



Import the data.

Step 3:

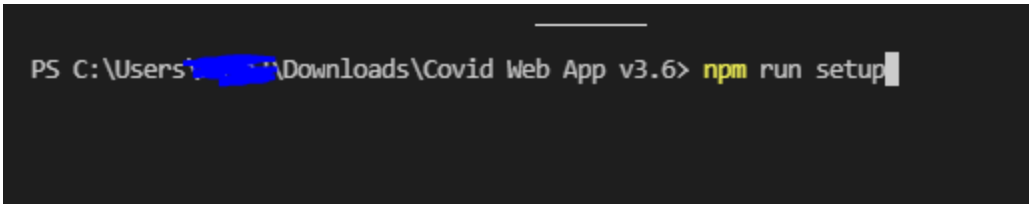
Change the connect name and password in /api/index.js to connect to your database/connection.

```
//connect to the database
var con = mysql.createConnection({
    host     : 'localhost',
    user     : 'root',
    password : 'root', // ENTER PASSWORD HERE
    database : 'covid_data'
});

con.connect();
```

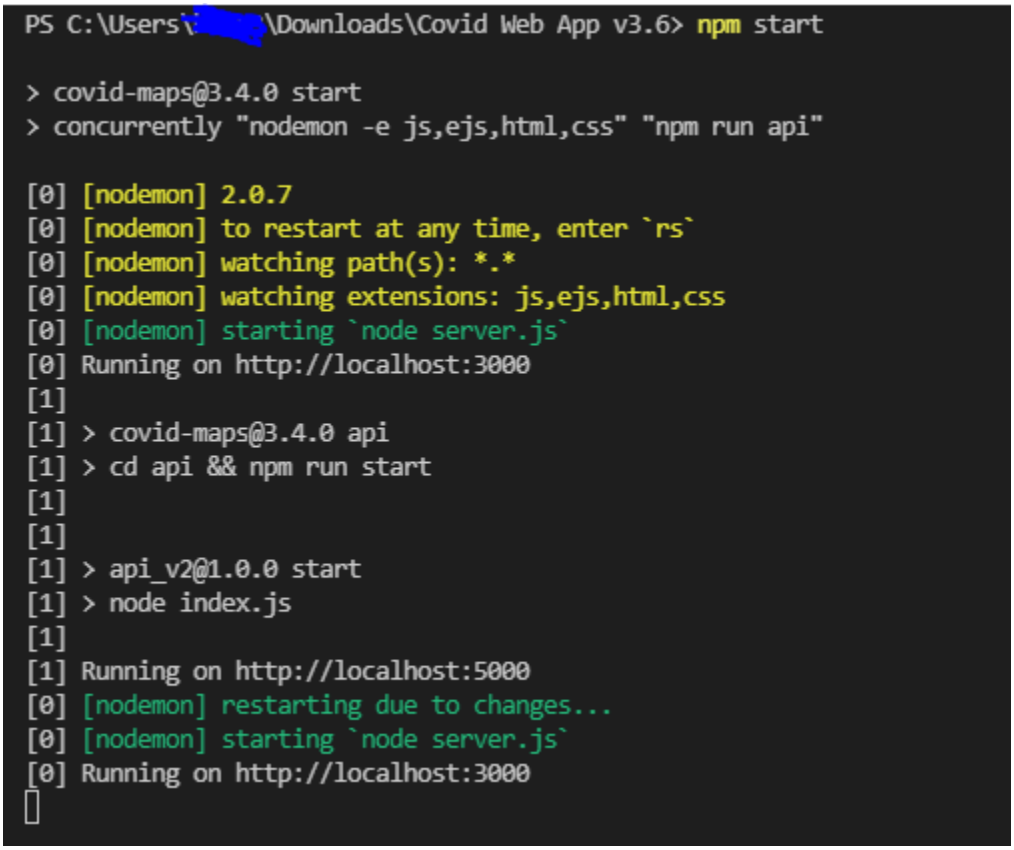
Step 4:

Open a terminal window and in the main directory of the software run the command npm run setup.



Step 5:

After running the setup run npm start to start up the webapp.



Step 6:

Then go to the url http://localhost:3000 to open the webpage.

3.2 General Maintenance

One thing you must do regularly is update the database. You can do this by loading in up-to-date data that matches

the current format of the database. You could back up your database, but it is not necessary for this web application.

4. Troubleshooting

4.1 Dealing with Error Messages and Failures

If an “ER_ACCESS_DENIED_ERROR” error is given with errno code 1045, then most likely you have not opened the index.js file in the api folder and changed it to match the password for the server that you set when creating it in MySQLWorkbench. In the file, the variable has a comment saying “ENTER PASSWORD HERE”. Once the value is changed, save the file and proceed with the installation.

If an “ECONNREFUSED” error is given with errno code -4078, then most likely you will need to check to see that your MySQL workbench database is actively running before starting the server with npm start. This error is thrown when the local server cannot be connected to.

4.2 Known Bugs and Limitations

Browser cant render 3000+ polygons without experiencing severe lag. A limitation is that we were unable to efficiently display the most recent earlier date in the event that there is no data on the selected date because the process required to do that cannot be handled by the local servers. The highlighted borders sometimes also cut across the borders on the map, not matching the actual borders of the map exactly, which may just be a limitation on leaflet. Furthermore, we choose to use the blue bubble markers instead of colored circles due to visual conflict issues of

having colored circles overlaid on colored polygons. Using the blue bubble markers also increased the performance of the web app.