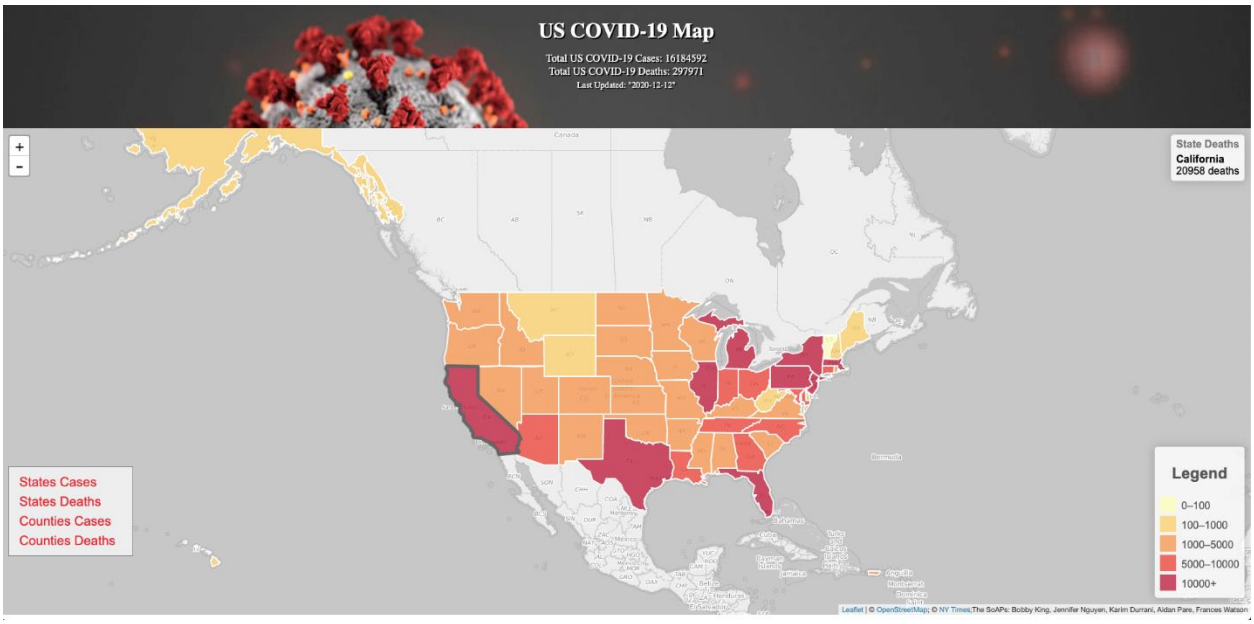
Administrator Manual



1. Introduction

This product is a software that displays a map of the United States color-coded by COVID-19 cases and deaths. The interface provides the end user the ability to select the granularity of data they want to see (e.g., county, state) and the ability to view current data. The product will continually pull new COVID-19 statistics for its database and will update the graphical user interface accordingly to accommodate the new data as it comes in.

2. System Overview

a. Background

The product is a combination of MySQL, JavaScript, and HTML/CSS files that cooperate in order to display COVID-19 statistical data on an interactive choropleth map. In that vein, for the product to run effectively, a MySQL server and a node.js server must be maintained and active. With the servers running, the

databases can have automated updates and a map with up-to-date information displayed. The product was designed such that it is able to be run on a single computer alone, so it should not be necessary to allocate a bulk of differing resources to get our program to run.

b. Hardware and Software Requirements

Software:

i. MySQL (with at least MySQL Server and MySQL workbench)

ii. Node.js (with the following dependencies: ajax, body-parser, convert-csv-to-json, cors, cron, csv, express, fast-csv, fs, jquery, jquery-csv, leaflet, mysql, node-fetch, temp)

Hardware (minimum requirements shown):

i. CPU: Intel Core or Xeon 3GHz (or Dual Core 2GHz) or equal AMD CPU

ii. Cores: Single (Dual/Quad Core is recommended)

iii. RAM: 4 GB (6 GB recommended)

iv. Graphic Accelerators: nVidia or ATI with support of OpenGL 1.5 or higher

v. Display Resolution: 1280×1024 is recommended, 1024×768 is minimum.

vi. OS: macOS, Windows, or Linux

3. Administrative Procedures

a. Installation

i. Download, install, and setup the MySQL server and database. You can use the following link to access the community edition: https://dev.mysql.com/downloads/

ii. Download, install, and setup Node.js. You can use the following link to find a downloadable package: https://nodejs.org/en/download/

iii. (Optional) If you do not have a linux command line available, you may find that downloading Git Bash makes pulling files and running commands much easier. You can use the following link to download the program: https://git-scm.com/downloads

iv. Clone the following github repository that includes the necessary files: https://github.com/Kutoux/SoAPs

v. Navigate via command line to the directory where the github repository was cloned and run "npm install". Running this command will install npm and download the necessary dependencies (see *2.b Software Requirements*).

vi. (Optional, but highly recommended) Run the commands on different terminals: "node express/js/cronjob.js" and "node express/js/cronjobGeo.js" to start automatic updates (will update the SQL database at 2:30 a.m. EST and the geojson files at 3:30 a.m. EST daily)

vii. On a separate terminal, run the command "npm start" to start the map website server

viii. To view the map, open any internet browser and visit: localhost:3000.

b. Routine Tasks

i. Should you decide to not run the cronjob script to automate database updates (*see 3.av Installation*) , you will need to run the update script with the file provided in the cloned repository. You can update manually by running the following lines of code (please run them on separate lines): “node express/rest_api/generateDB.js”, “node express/js/generateGeo.js”.

c. Periodic Administration

i. As a best practice, you should backup the database biweekly/monthly. On linux systems you can perform a sqldump via the command line with the following command: “sudo mysqldump -u [user] -p [database_name] > [filename].sql” where [user] is the username and password (if necessary), [database_name] is the path and filename of the database, and [filename] is the desired name of the backup file.

d. User Support

i. Issues with the MySQL database may be answered at: https://www.mysql.com/about/faq/

ii. Issues with the Node.js may be answered at: https://nodejs.org/en/docs/guides/

iii. Should the above resources not resolve issues, you may contact customer support at jennifn1@umbc.edu. Customer service is available Monday-Friday 9 am to 5 pm. Please allow customer service 1-3 business days to get back to you.

4. Troubleshooting

a. Dealing with Error Messages and Failures

i. When doing a manual update, if the output does not look like the images below, the data's csv files may either be corrupted or ill formatted.

```
null OkPacket {
  fieldCount: 0,
  affectedRows: 0,
  insertId: 0,
  serverStatus: 2,
  warningCount: 0,
  message: '',
  protocol41: true,
  changedRows: 0
}
null OkPacket {
  fieldCount: 0,
  affectedRows: 826402,
  insertId: 0,
  serverStatus: 2050,
  warningCount: 0,
  message: '+Records: 826402  Duplicates: 0  Warnings: 0',
  protocol41: true,
  changedRows: 0
}
```

```
(base) Bobbys-Air:SoAPs king$ node express/js/generateGeo.js
[]


Closed
Closed
Closed
Closed
```

b. Known Bugs and Limitations

i. The default port number that the node.js server runs on is 3000. Currently, we do not support changing that port number if the user desires so.

ii. We do not currently have support for displaying prison data on our maps, due to the inconsistent formatting of the prison data csv files. Consequentially, the prison geojson dataset does not created, but the data is still viewable via the MySQL database.

iii. If you see an "undefined" value on the top right info box while scrolling over the map on the webpage, it is possible that the geojson is lagging in its information retrieval process. Refreshing the page should fix this error.

iv. Currently, users can only view the most up to date info. We do not support viewing previous dates' data at this time.