Mission: The UMBC Covid-19 Tracker - Provide Covid-19 related case and vaccine information for public safety and knowledge

DID: Software Test Report (STR)

PREPARED by: Quinque Equos

(

[Alex Wilson](mailto:alexanw1@umbc.edu)

Safia Shah

[Israel Morocho](mailto:cx59298@umbc.edu)

[Duncan Taylor](mailto:ak58697@umbc.edu)

[Tom Tennant](mailto:tent1@umbc.edu)

[David Kravets](mailto:kravets1@umbc.edu)

)

DATE: APRIL-2021

REVISION LOG

| Revision | By | Description | Date |
| --- | --- | --- | --- |
| A | Team | Initial Version | 04/29/2021 |
| B | Team | Updated All sections | 05/04/21 |

Table Of Contents

[**Scope**](#_nvr0dbhej8y5) **5**

[1.1 Identification](#_pzekma8ucbql) 5

[1.2 System Overview](#_pzekma8ucbql) 5

[1.3 Document Overview](#_pzekma8ucbql) 7

[**Referenced Documents**](#_gcxumdb42sbr) **7**

[**3. Overview of Test Results**](#_73reuv77zqwb) **7**

[3.1 Overall assessment of the software tested.](#_2c8tmn8kbymg) 7

[3.2 Impact of Test Requirements](#_2c8tmn8kbymg) 8

[3.3 Recommended Improvements](#_2c8tmn8kbymg) 8

[**4. Detailed Test Results**](#_4wyaaxdivf07) **8**

[4.2.6 Problems Encountered](#_74gy5xeko4me) 12

[4.2.7 Deviations from test cases/procedures.](#_mmokx0h6ryzl) 12

[4.3 Database Tests](#_mmgi6xayll8) 12

[4.3.1 Summary Of Test Results](#_ghyml6gi02ep) 13

[4.3.2 Problems encountered.](#_k3x8it9kzhr9) 15

[4.3.3 Deviations from test cases/procedures.](#_5kgck23tskm0) 15

[**5. Test Log**](#_4wyaaxdivf07) **16**

[**6. Notes**](#_bope32zeucft) **16**

# 

List of Tables

Table 1: Possible functions for the CDT

Table 2: Outcomes for database testing

List of Figures

Figure 1: Overview of system

Figure 2: Flow of Execution

Figure 3: Interfaces necessary for the CDT

# Scope

## 1.1 Identification

This Software Development Plan (STP) pertains to the development of the UMBC Covid-19 Data Tracker application. The intended audience includes the engineers tasked with developing the application and the test conductors responsible for the verification and acceptance of the application.

## 1.2 System Overview

The Covid-19 Data Tracker (CDT) web applications main purpose is to be informative about the covid-19 virus and its currently available vaccines. The CDT web application will be designed to display a heat map of a selected state and, if applicable, the counties within and their covid data. This web application does not expand outside of U.S data. The data will be queried from the Broadstreet covid data sets in order to generate an organized display for the user selected areas.

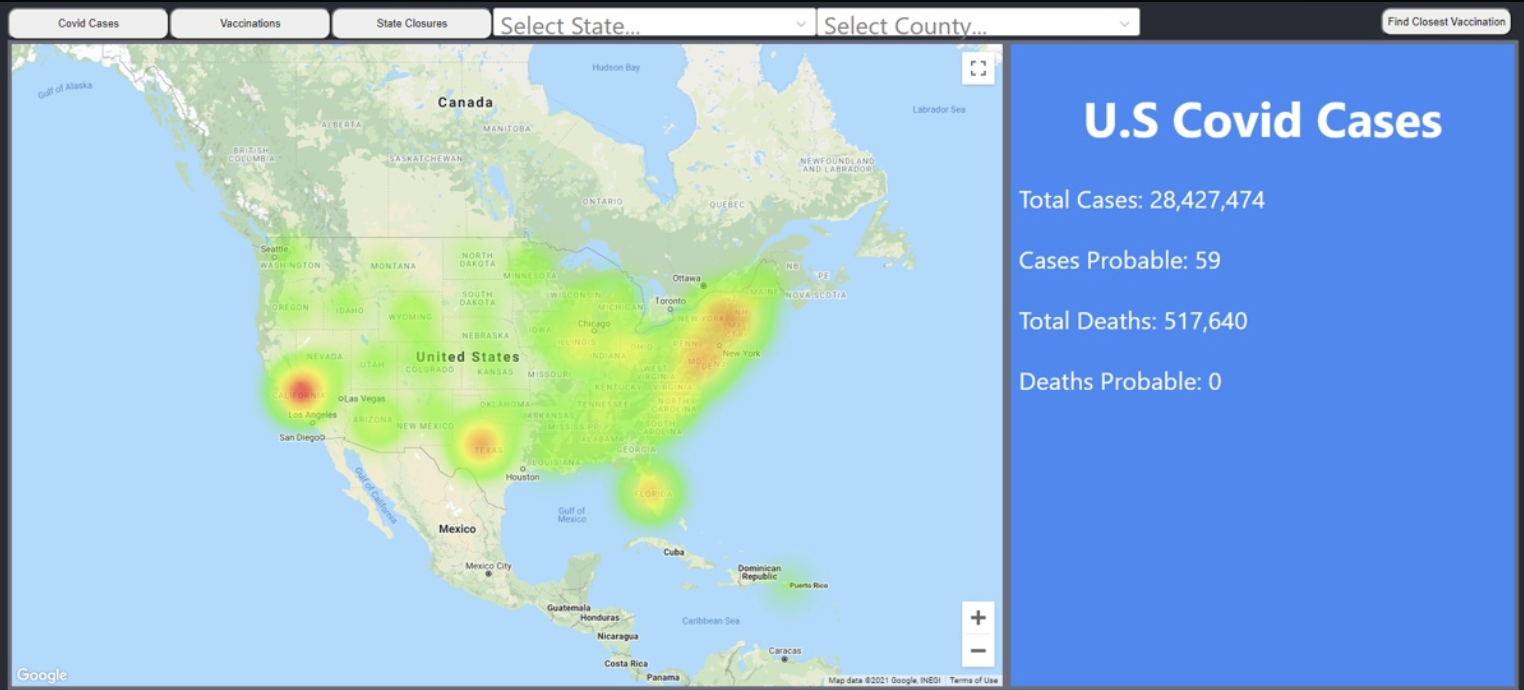


Figure 1: Overall visual of UMBC Covid Data Tracker

Below are the possible functions of the UMBC CDT:

\*Functions are listed as tabs displayed on the front-end of the UMBC CDT

| Function | Description |
| --- | --- |
| Covid Cases | Allows a heatmap display of the user selected state and county for their respective covid case related data. This data includes: deaths, confirmed cases, possible cases. Numerical data is displayed to the right of the heatmap with a statics column |
| Vaccinations | Allows a heatmap display of vaccination data. Numerical data is displayed to the right of the heatmap with a statics column |
| State Closures | Allows a display in the statistical column of the state closure policies currently intact. This does not expand into individual counties within a state |
| Find Closest Vaccination | Link to <https://www.vaccines.gov/search/> where the user is redirected to an area to search their closest vaccine site. |

Table 1: Possible Functions



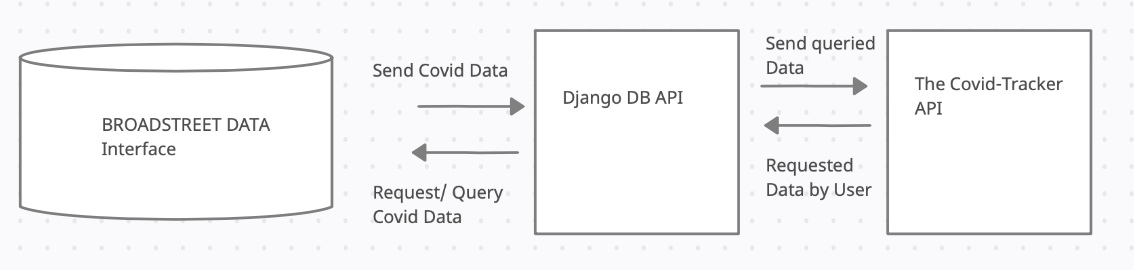
Figure 2: Flow of Execution

Figure 3: The UMBC Covid-19 Data tracker consists of 3 interfaces

## 1.3 Document Overview

Section 1 describes the scope of the project. Section 2 provides referenced documents. An overview of results and detailed test results are included in Sections 3 and 4, respectively. Section 5 presents the test logs.

# Referenced Documents

* 1. Software Test Report Template

<https://alt-5954eb74c7df4.blackboard.com/bbcswebdav/pid-4725370-dt-content-rid-47159292_1/courses/CMSC447_11379_SP2021/STR.htm?one_hash=F48EE2899AE76FF9372983C2F2A0FE44&f_hash=B35CC09BBEB2F1A34C76E8C8610C3B0A>

* 1. Quinque Equos Covid-19 Covid Data Tracker Software Requirement Specification (SRS)

<https://docs.google.com/document/d/1kl3riti_Gm3MA275jgtFUKiSZ86POZsVcov-uhFyYSE/edit?usp=sharing>

* 1. Quinque Equos Covid-19 Covid Data Tracker Software Test Plan (STP)

<https://docs.google.com/document/d/1eA-tAFdtG9A6wZ_7dmnY53vQAzZAQktPk2egJ-gasE8/edit?usp=sharing>

* 1. Quinque Equos Covid-19 Covid Data Tracker Software Development Description (SDD)

<https://docs.google.com/document/d/1CThzQMoRKqLOV1wme_l3mFHD8wpiCP1nipAAEGqZ6CA/edit?usp=sharing>

* 1. Quinque Equos Covid-19 Covid Data Tracker Software Development Plan (SDP)

<https://docs.google.com/document/d/1-Jzi5aSdVpQueVgdMcXMtLGFlsWKKXbrv6YH79iCcEw/edit?usp=sharing>

# 3. Overview of Test Results

This document shall describe the results of the actions of the UMBC Covid-19 Data Tracker Software Test Plan with the requirements defined in the Software Requirements Specification (SRS).

## 3.1 Overall assessment of the software tested.

Through the action of the tests set forth it is noted that the software is performing as designed and shall be recommended for further integration. The Covid-19 tracker was successful in that it managed to quickly and readily provide up to date covid data to users accessing the front end.

## 3.2 Impact of Test Requirements

The tests set forth were only considering the operation of the software under a one user load. We are uncertain of the dependability of our product when released to a broader audience or hosted to the public. Thorough vulnerability analysis has yet to be performed. So far, we have proven that our software runs smoothly under a variety of common use cases.

## 3.3 Recommended Improvements

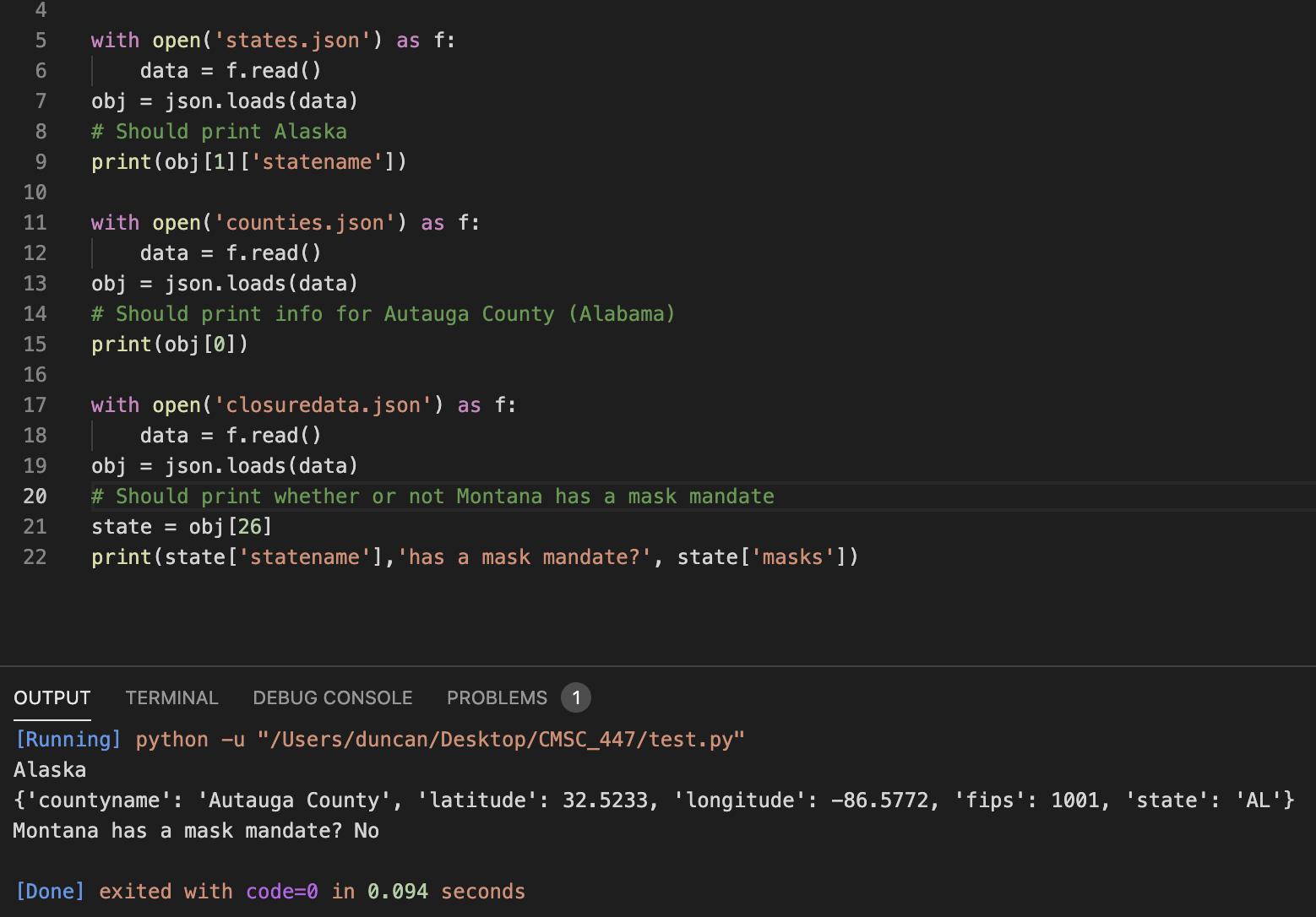
More thorough test cases for backend security should be performed. These tests could include penetration testing, system robustness, protection from DDos and etc. Frontend tests should be performed to certify that no combination of user entries results in a crash or overloaded query.

# 4. Detailed Test Results

The results of the tests performed against the UMBC Covid-19 Data Tracker web application are described within this section.

**4.1.1 API Testing**

The following tests were performed on the API documents to ensure the states, counties, and closure data is accessible



**4.1.2 Summary of Test Results**

The testing file accesses the JSON files associated with the database data and prints the associated value of the provided key for the states file, an entire dictionary for the counties, and creates a sentence with closure data regarding a specific state. For each of these JSON files the data is stored as an array of dictionaries. The output was as expected.

**4.1.3 Problems Encountered**

Initially the open system call was not working to access the JSON files. My solution to investigate was to import the os library for python to see my current working directory (CWD). The CWD was pointing to the wrong directory so I reset my CWD to access the JSON files with a function from Python’s os package.

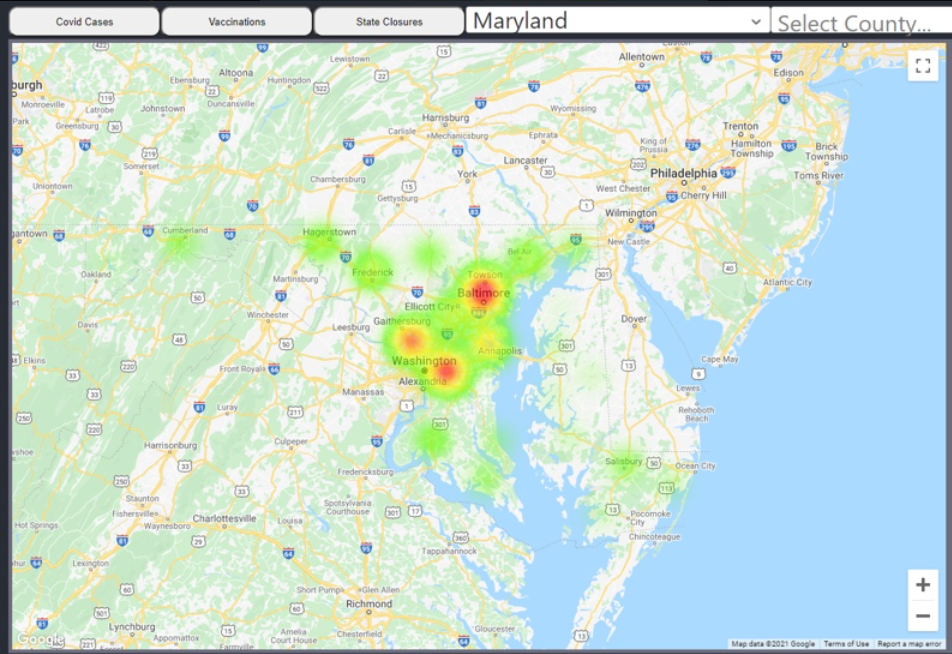
**4.2.1 Front End Testing**

**4.2.2 Search Box**

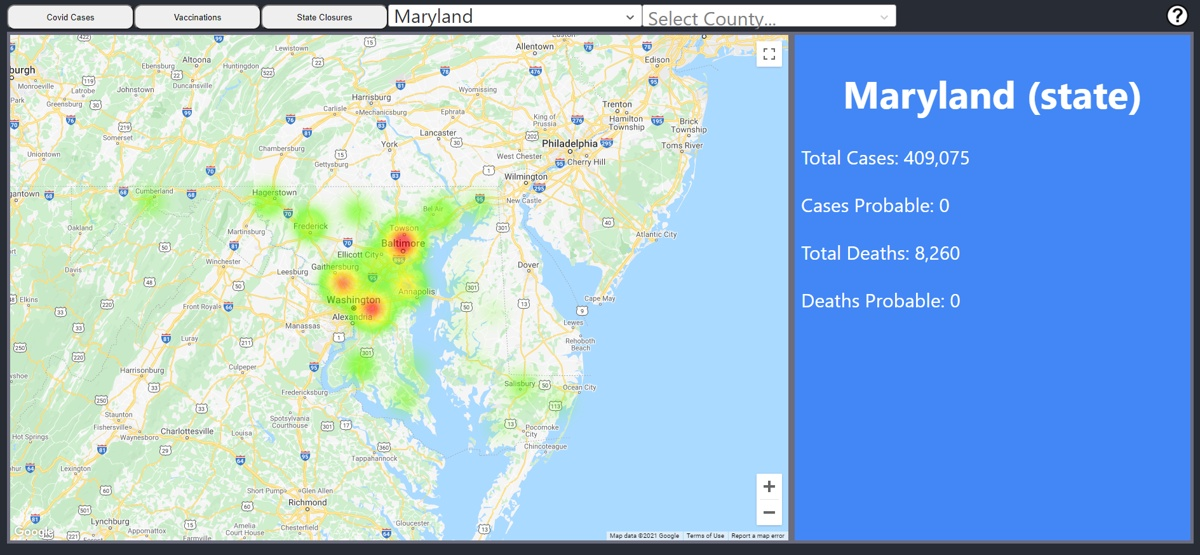
Our goal is for users to be able to use the drop down search boxes for States and Counties to search for the locations they wish to gain information on. To test these boxes we would click the drop down tab and see if we are able to access any state or county we desire.

**4.2.3 GUI Testing**

Our GUI testing is split into two main parts. Our first will be the user’s interaction with the map. Our goal is to have the map move and zoom into the location the user selects given the state and/or county from the search boxes. In order to test each location we created a script program that would run through each state and county in alphabetical order. As the script ran we would view if the current location that the script was on matched the current location the map was displayed on. In the screenshot below we provide an example of one of the areas. Here you can see the State box has Maryland, and we can also see that our map’s view is currently displayed correctly on Maryland as well.

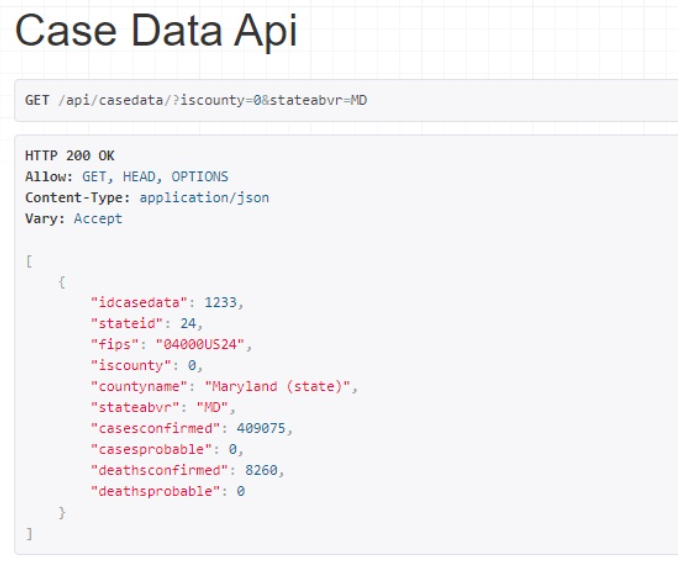


The next part of our GUI requires us to test the visual aspect of our heatmap. The goal is to have every state to have several heat points which would represent the counties of that state. To conduct our test we were able to reuse our previous script as we would need to visit each State and see if the number of heat points visually seen match the number of counties the state should have. We have provided a screenshot below as an example, in which we can see that the State Maryland which has 23 counties, matches our map with 23 heat points each referring to the counties in Maryland.



**4.2.4 Statistic Display Column**

Our statistical display column is meant to display the information of the location the user has chosen. Our goal is to have the correct display for each location seen within this column. To do so we were able to once again re-use our previous script as we would be visiting every state and county. As we visit each location we would check that the statistic display column is showing the correct data for the location by comparing it with the information in our database. Here we have a screen shot in which we can see the case data for Maryland on the left with a screen shot of the case data for Maryland seen in our database.



**4.2.5 Button Functionality**

Our program will have three main buttons known as Covid data, Vaccine data, and Closure data. Our goal is for users to be able to switch between the types of data for the locations they desire and have the different types of data be displayed on the statistical display column. To test the button’s functionality was simple, once we were able to confirm that the correct data would be displayed in the statistical column, we were able to test the functionality of the buttons by switching between them and confirming that the statistical display column did indeed output the correct type of data, or if no data was present, to indicate so.

### 4.2.6 Problems Encountered

One issue we encountered was that for some locations if we were to look at a location which had at least one county and then switch to a location that had none such as D.C. we would receive an error. This error came about the program attempting to search for county data when there would be none.

This occurred during our GUI testing phase, and would appear for any areas which had no counties. The error luckly did not cause any other problems and got be fixed by refreshing the page which would then allow the user to look at information of that location which had no counties.

To solve this problem we implemented a check system which would allow the program to check if any counties exist before attempting to retrieve their information, thus eliminating the error. This only took about two attempts to perfect as it was a simple fix.

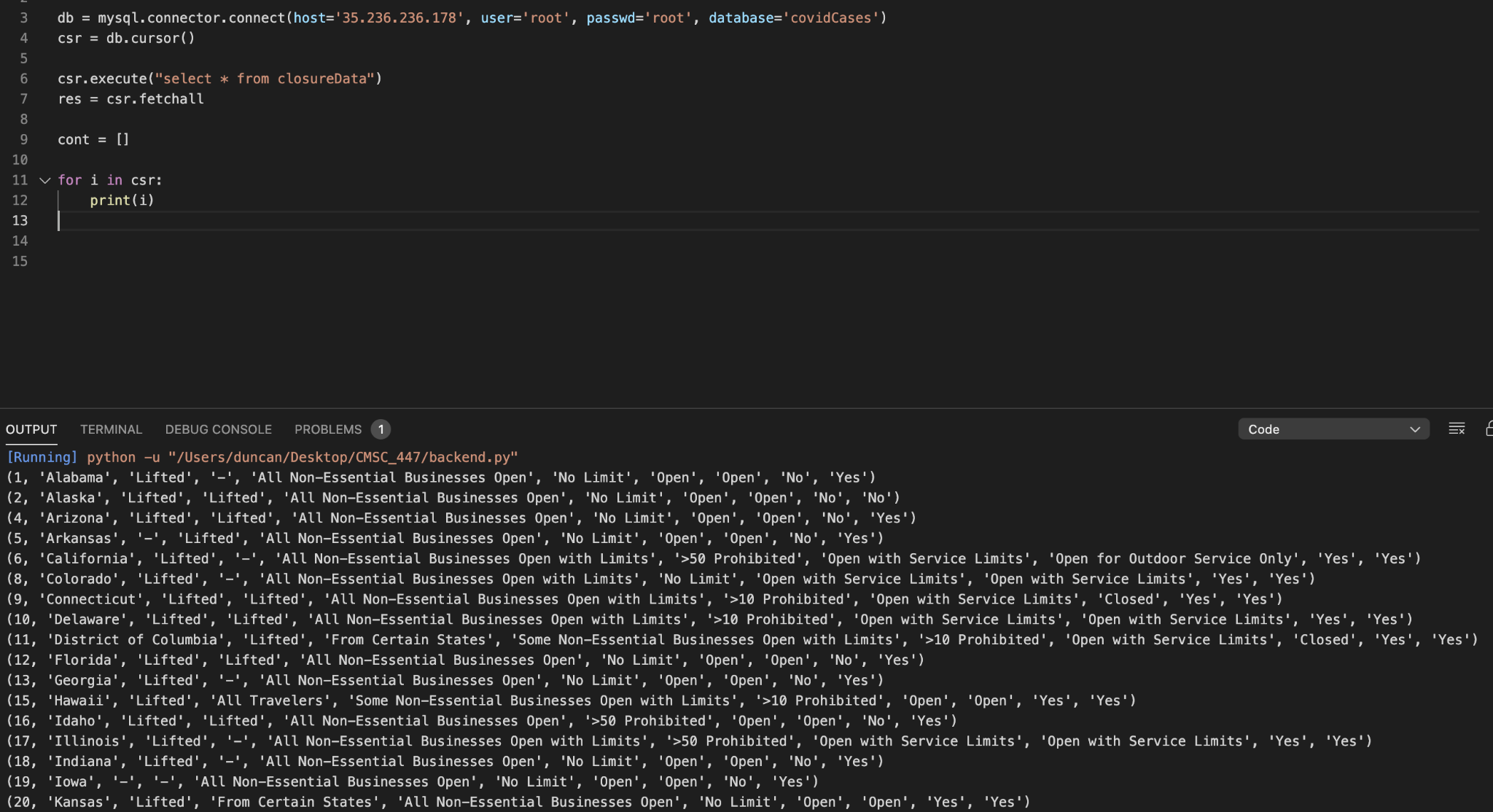
Excluding the first issue the Covid Data Tracker’s front-end returned all correct results and was working as intended.

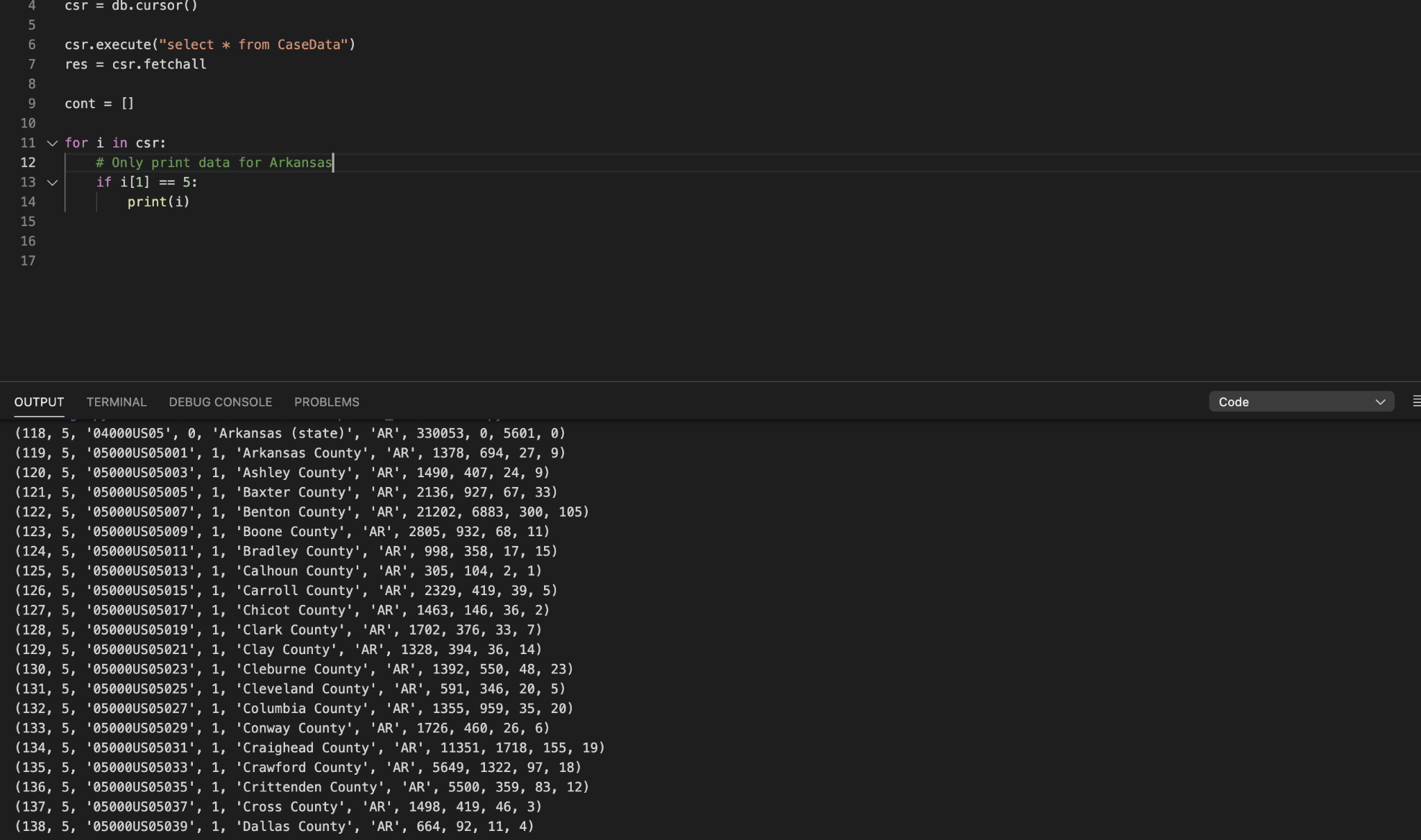
### 4.2.7 **Deviations from test cases/procedures.**

The testing designed in the STP was followed as written and did not deviate.

## 4.3 Database Tests

These tests were the tests performed on the database to ensure correct and efficient results were obtained. Many different tests were administered on each table.





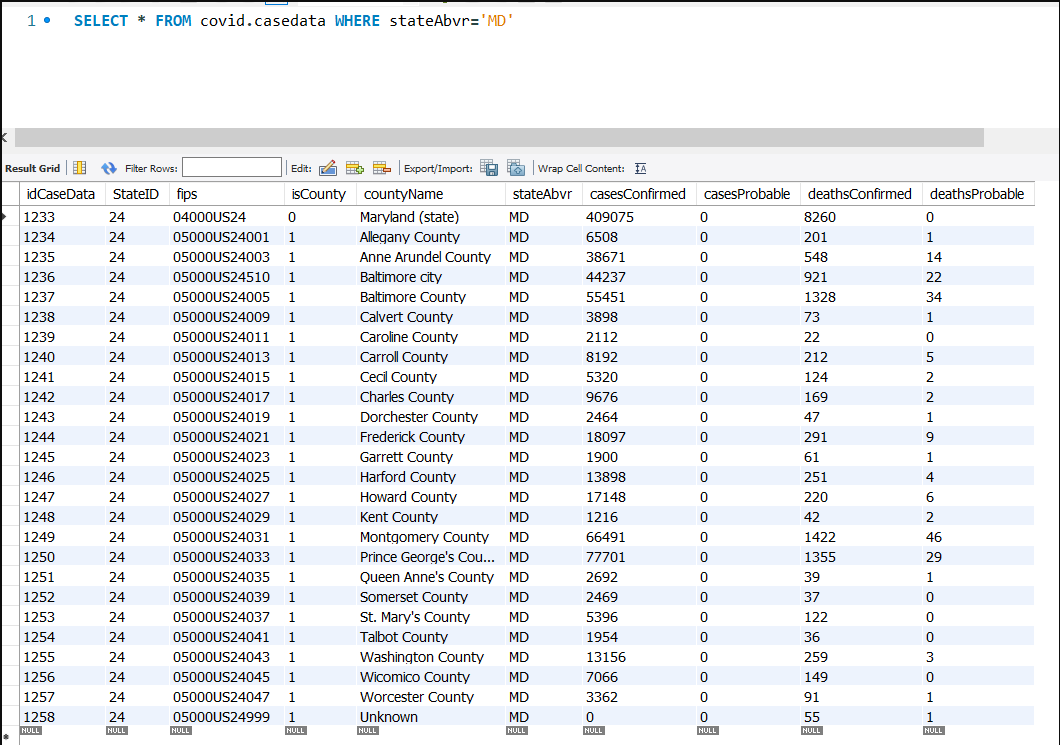
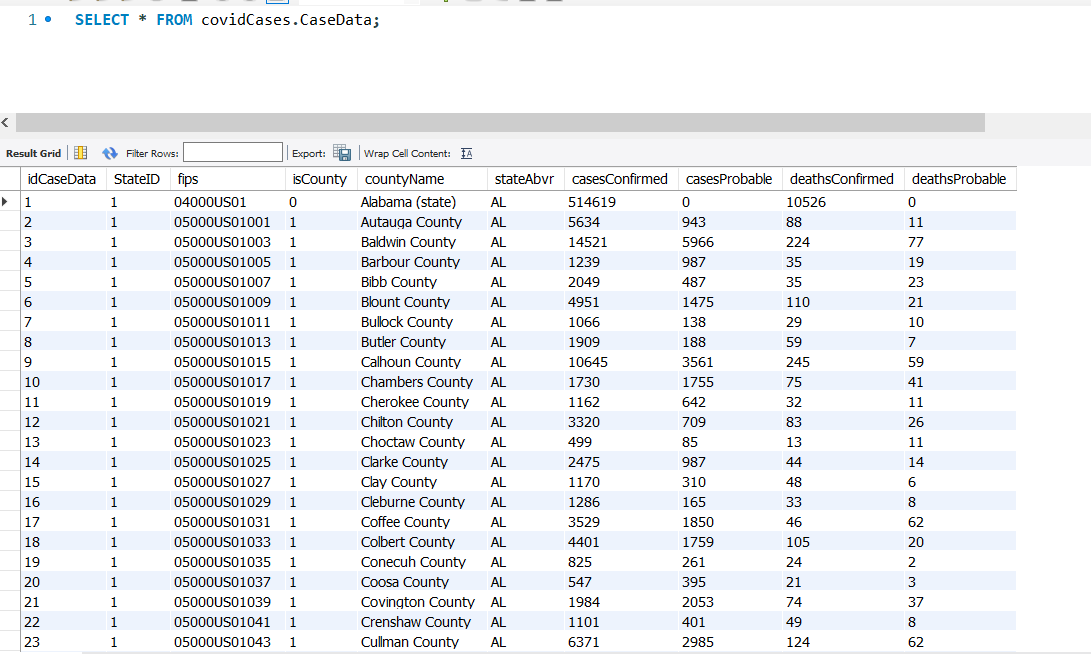


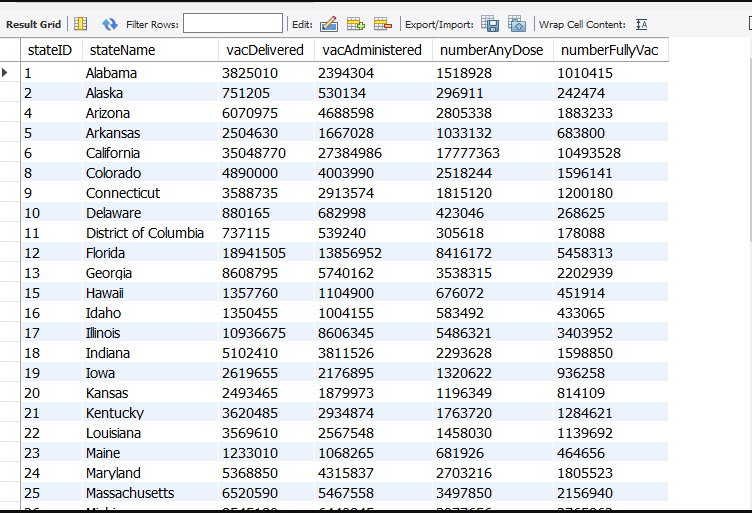
### 4.3.1 Summary Of Test Results

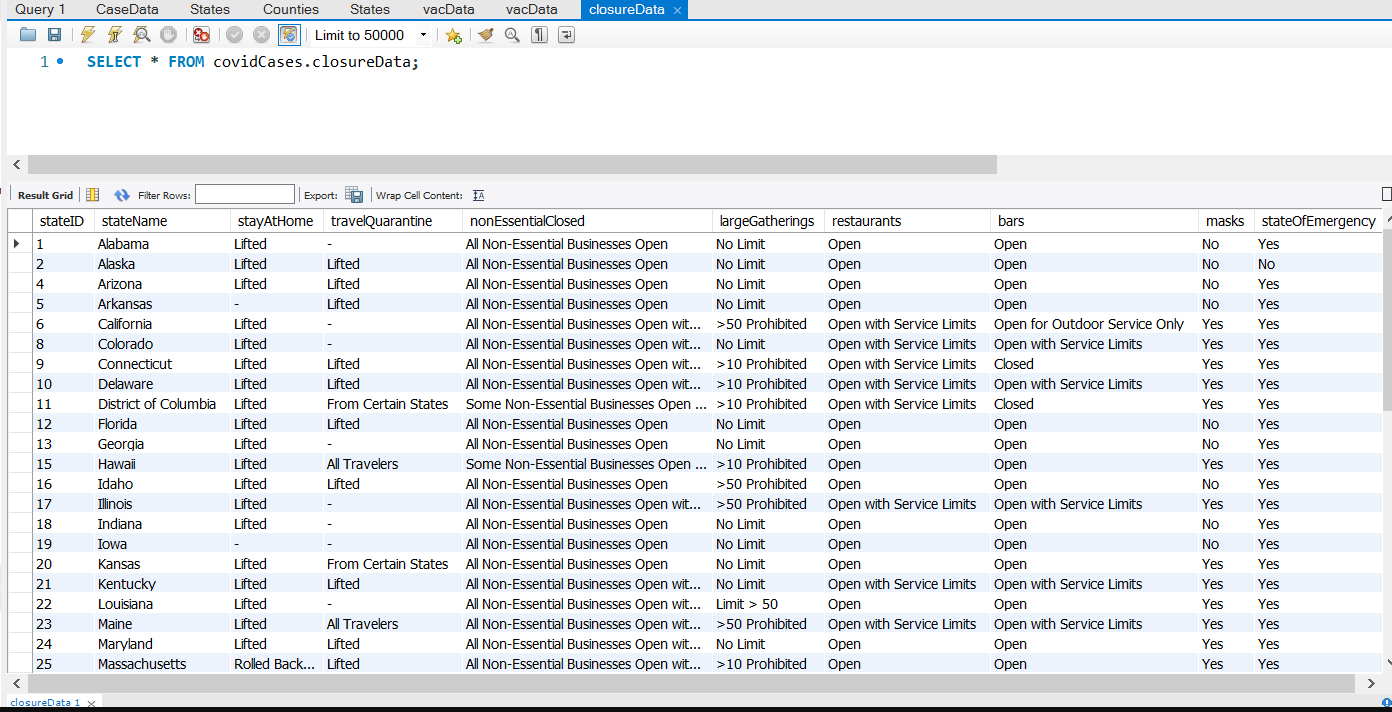
Tests were performed on each individual table inside of our database. These tables include Covid Cases, Vaccination Data, and Closure Date. For each table, a SELECT \* was used to return the data present in each column and row. This was enough to ensure that data was returned properly for two out of three tables. The one this was not sufficient for was the Covid Cases table. This table has over 3000 rows and needs to be closer examined. We then queried the Covid Cases table by state for multiple states to ensure data was as expected.

Table 2

| Test | Result |
| --- | --- |
| Covid Cases, all | As Expected |
| Covid Cases, by state | As Expected |
| Vaccination Data | As Expected |
| Closure Data | As Expected |







### 4.3.2 Problems encountered.

The Covid Data Tracker’s database returned all correct results and was working as intended.

### 4.3.3 Deviations from test cases/procedures.

The testing designed in the STP was followed as written and did not deviate.

# 5. Test Log

# 6. Notes