CS595-Termination Project
Jithin Jose
B00815334

Covid 19 Dashboard

Introduction

Coronavirus disease 2019 (COVID-19), also known as the coronavirus, or COVID, is a contagious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The first known case was identified in Wuhan, China, in December 2019. The disease has since spread worldwide, leading to an ongoing pandemic.

This project is aimed at developing a dashboard web application which displays statistics related to Covid-19.

Accurate, timely data is a key tool in states' efforts to understand and respond to the impact of the coronavirus (COVID-19) outbreak. There have also been increasing calls to further break down COVID-10 data into subcategories (such as by gender, age, and race and ethnicity) in order to track the impact of the disease on specific populations. These dashboards are designed to organize complex data in an easy-to-digest visual format, allowing the audience to easily interpret key trends and patterns at a glance

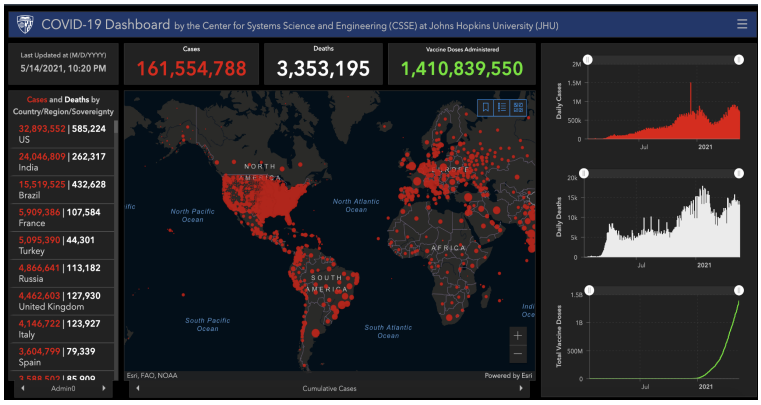
Literature Survey

Many countries have developed their own covid-19 dashboards. Considering the US itself, there were 32 states with public-facing COVID-19 data dashboards as of April 6, 2020. States are reporting a wide number (ranging from 4 to 13) and type of indicators in their dashboards, most of which are updated at least daily. Many states are also starting to show trends in these data points over time. The most common indicators reported on a state dashboard include:

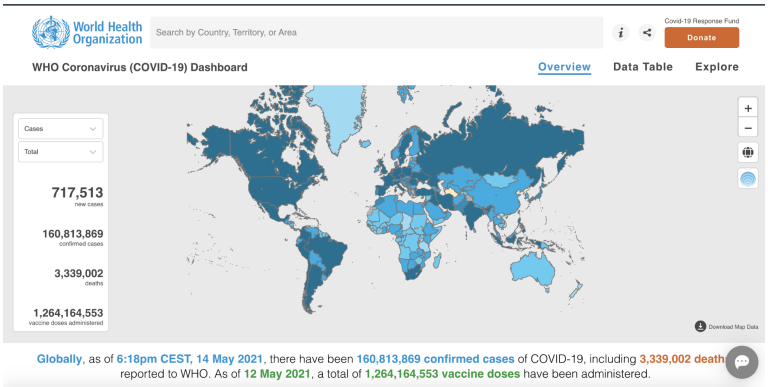
- Number of total cases
- Number of total deaths
- Number of cases by county
- Map of cases by county
- Number of tests completed
- Number of cases by age group
- Number of cases by gender
- Number of deaths by county
- Number of hospitalizations

Below are a few images of some covid dashboards:

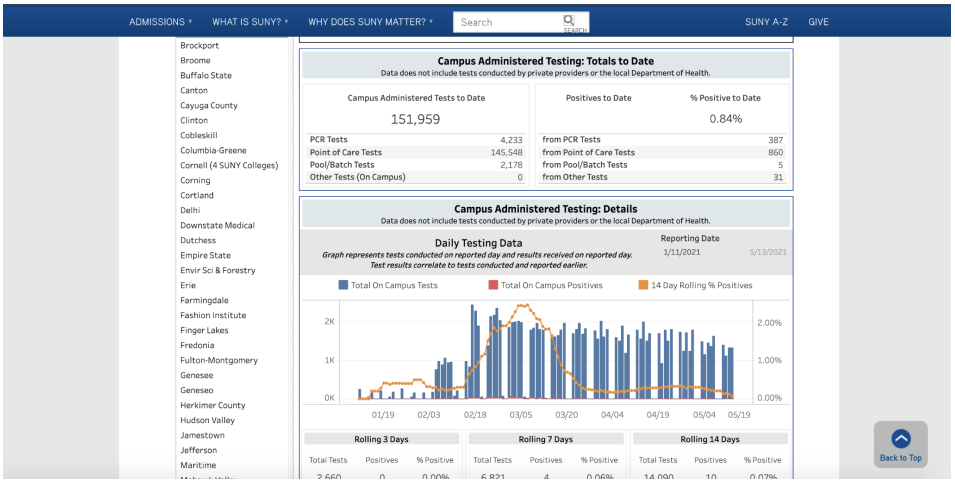
John Hopkins University:



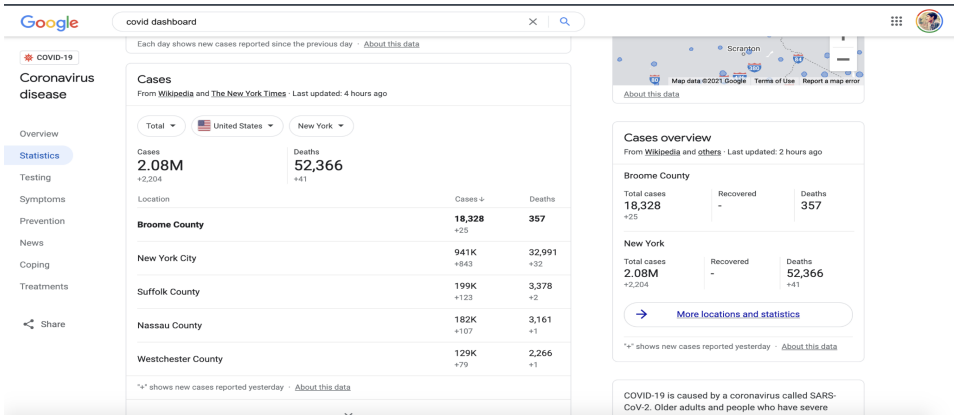
World Health Organization (WHO):



State University of New York:



Google:



Dataset

The dataset being used is operated by the Johns Hopkins University, Center for Systems Science and Engineering (JHU CSSE). Also, Supported by ESRI Living Atlas Team and the Johns Hopkins University Applied Physics Lab (JHU APL). The dataset is updated daily and updates occur between 04:45 and 05:15 GMT. This dataset has the following attributes:

- **FIPS**: US only. Federal Information Processing Standards code that uniquely identifies counties within the USA.
- **Admin2**: County name. US only.
- **Province_State**: Province, state or dependency name.
- **Country_Region**: Country, region or sovereignty name. The names of locations included on the Website correspond with the official designations used by the U.S. Department of State.

- **Last Update**: MM/DD/YYYY HH:mm:ss (24 hour format, in UTC).
- **Lat and Long**: Dot locations on the dashboard. All points (except for Australia) shown on the map are based on geographic centroids, and are not representative of a specific address, building or any location at a spatial scale finer than a province/state. Australian dots are located at the centroid of the largest city in each state.
- **Confirmed**: Counts include confirmed and probable (where reported).
- **Deaths**: Counts include confirmed and probable (where reported).
- **Recovered**: Recovered cases are estimates based on local media reports, and state and local reporting when available, and therefore may be substantially lower than the true number. US state-level recovered cases are from COVID Tracking Project.
- **Active**: Active cases = total cases - total recovered - total deaths.
- **Incident_Rate**: Incidence Rate = cases per 100,000 persons.
- **Case_Fatality_Ratio (%)**: Case-Fatality Ratio (%) = Number recorded deaths / Number cases.

Implementation

This project has the following tech stack:

- Spring MVC
- MySQL
- Hibernate
- Python
- Thymeleaf
- HTML/CSS/Jquery

MySQL:

MySQL is being used to store the records from the dataset in the database. All attributes are being stored except for Incident Rate, Case Fatality Ratio, FIPS and Admin2.

Hibernate:

Hibernate is used to update the database in case of any changes to the database schema. It keeps the database and the entity class in sync.

Python:

Python script is responsible for the visualization. The database records are converted into a dataframe. **Pandas** library is used for the cleaning of data.

Data aggregation is done by applying summation on “confirmed”, “deaths” ,”active”and “recovered” columns and grouping the summed values by “country”. The aggregated data is then stored in a separate dataframe.

This data is used to calculate the total number of cases at the global level and country level. For the global level, I just applied the sum() method to the 4 columns mentioned above.

For the top 10 countries, I used the nlargest() method provided by pandas. This method takes an n-value and a column name.

All of the above data is then represented visually using plots provided by the **Plotly** library. The 3 plots being used are: Scattergeo plot, Indicator plot and Bar plot.

Spring MVC:

The Spring Web MVC framework provides Model-View-Controller (MVC) architecture and ready components that can be used to develop flexible and loosely coupled web applications. The MVC pattern results in separating the different aspects of the application (input logic, business logic, and UI logic), while providing a loose coupling between these elements.

- The Model encapsulates the application data and in general they will consist of POJO.
- The View is responsible for rendering the model data and in general it generates HTML output that the client's browser can interpret. This is where **Thymeleaf** comes in as a template-engine.
- The Controller is responsible for processing user requests and building an appropriate model and passes it to the view for rendering.

This project has the following classes:

Corona: This is the entity class that has all the attributes, and its setters and getters.

CoronaController: This class acts as the controller for all endpoints and returns HTML templates for all endpoints. It's got methods that pass the database records to the HTML template.

CoronaService: This class has methods that accesses the database and populates it. It updates the database with the latest records from the dataset daily at a scheduled time.

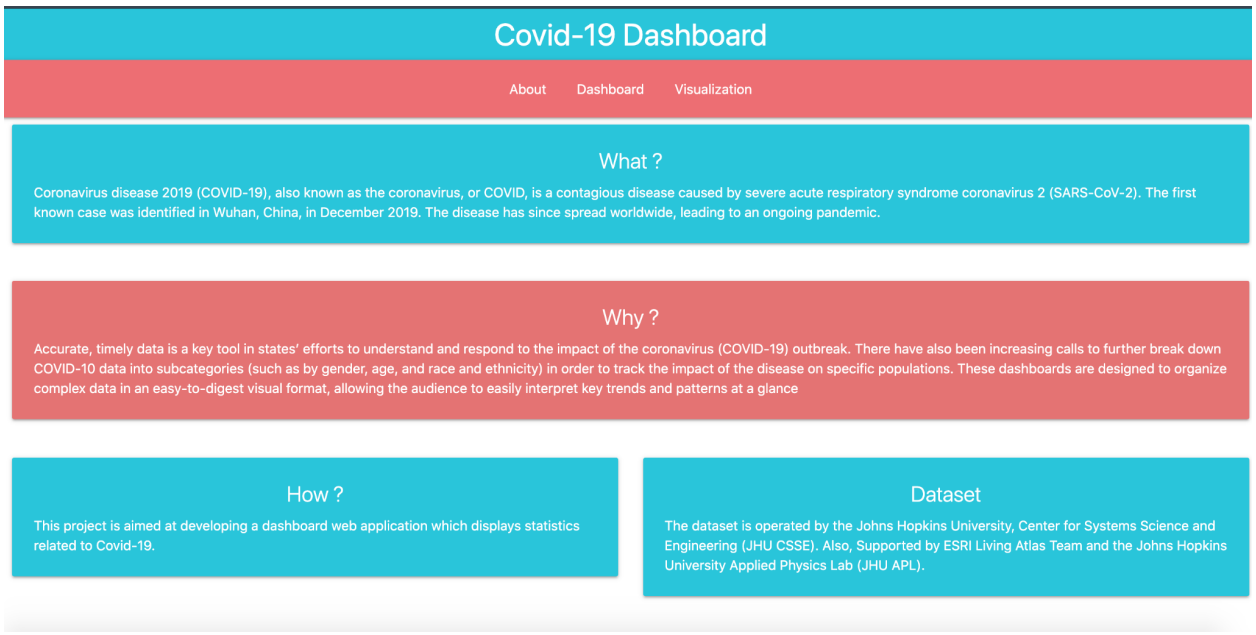
CoronaRepository: This class provides the mechanism for storage, retrieval, search, update and delete operation on objects.

HTML/CSS/Jquery:

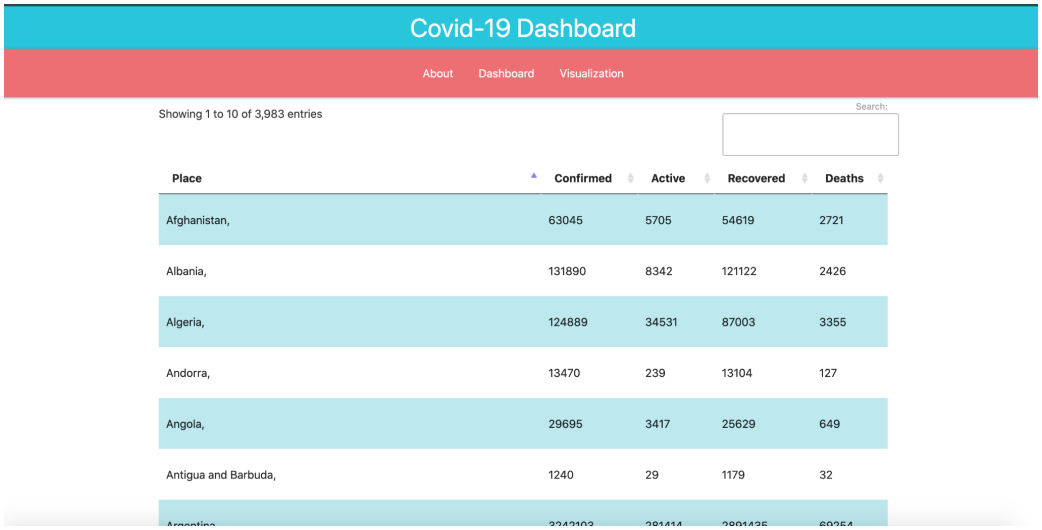
Used for designing the frontend. Used Materialize stylesheet for css. Also, I used DataTables which is a plug-in for the jQuery Javascript library. It provides features like pagination, filtering etc.

Results

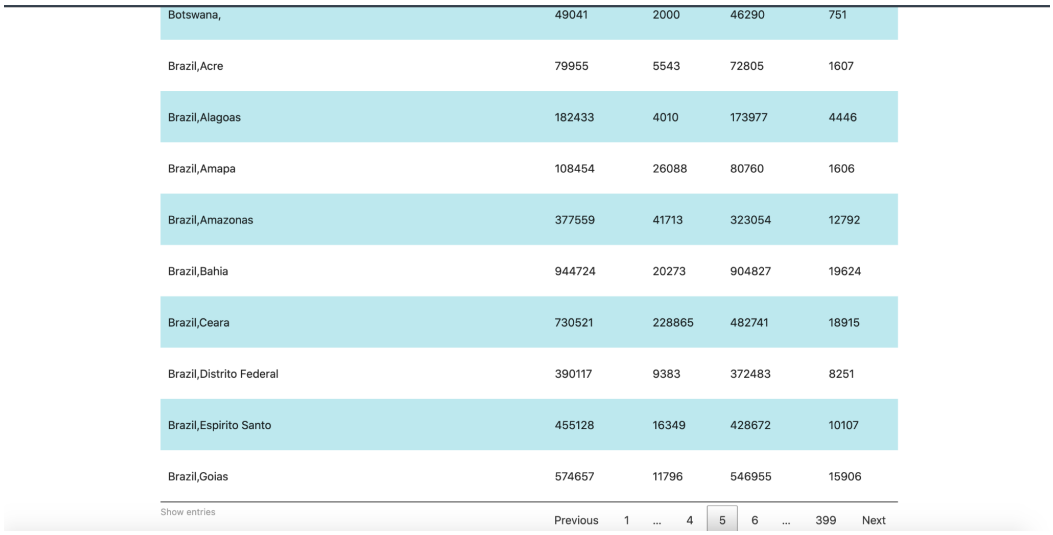
About:



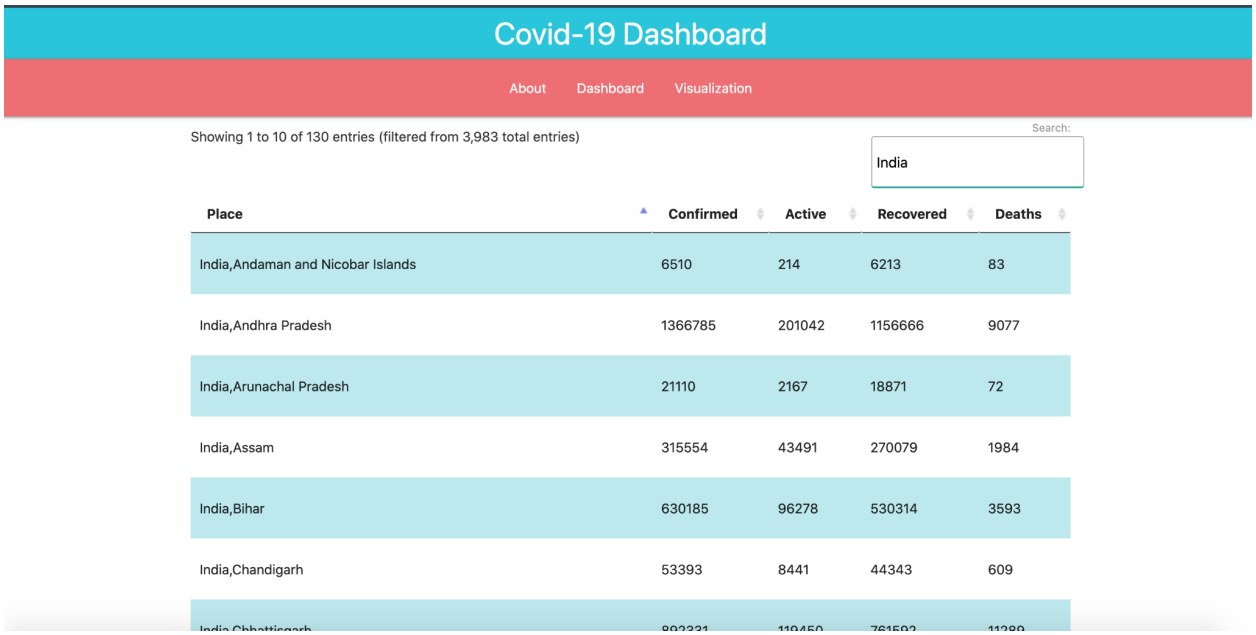
Dashboard:



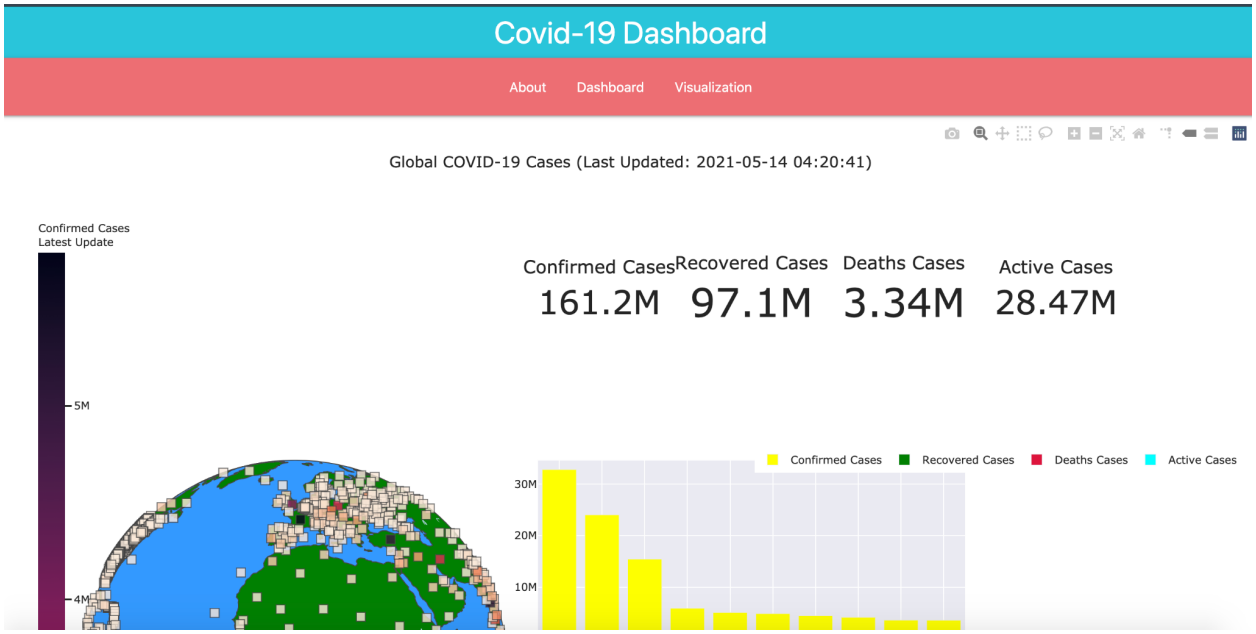
Pagination:

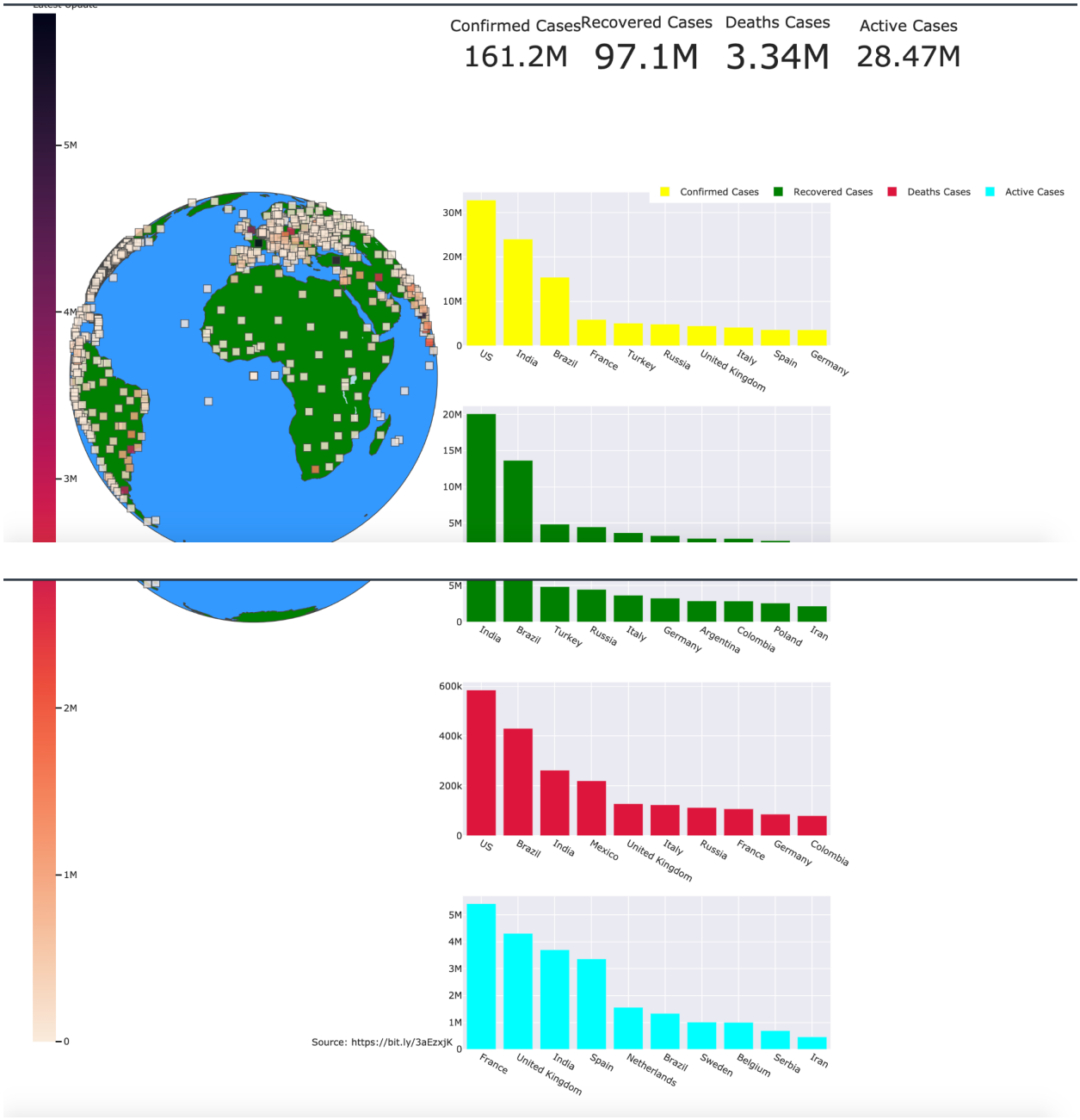


Filtering records for India:



Visualization:




Confirmed Cases
161.2M
Recovered Cases
97.1M
Deaths Cases
3.34M
Active Cases
28.47M
Confirmed Cases
Recovered Cases
Deaths Cases
Active Cases
US
India
Brazil
France
Turkey
Russia
United Kingdom
Italy
Spain
Germany
India
Brazil
Turkey
Russia
Italy
Germany
Argentina
Colombia
Poland
Iran
US
Brazil
India
Mexico
United Kingdom
Italy
Russia
France
Germany
Colombia
France
United Kingdom
India
Spain
Netherlands
Brazil
Sweden
Belgium
Serbia
Iran
Source: https://bit.ly/3aEzxjK

References

1) **Dataset** CSSEGISandData/COVID-19: Novel Coronavirus (COVID-19) Cases, provided by JHU CSSE
2) Building A Real-time Dashboard Using Python Plotly Library And Web Service
3) State COVID-19 Data Dashboards
4) Coronavirus (COVID-19)