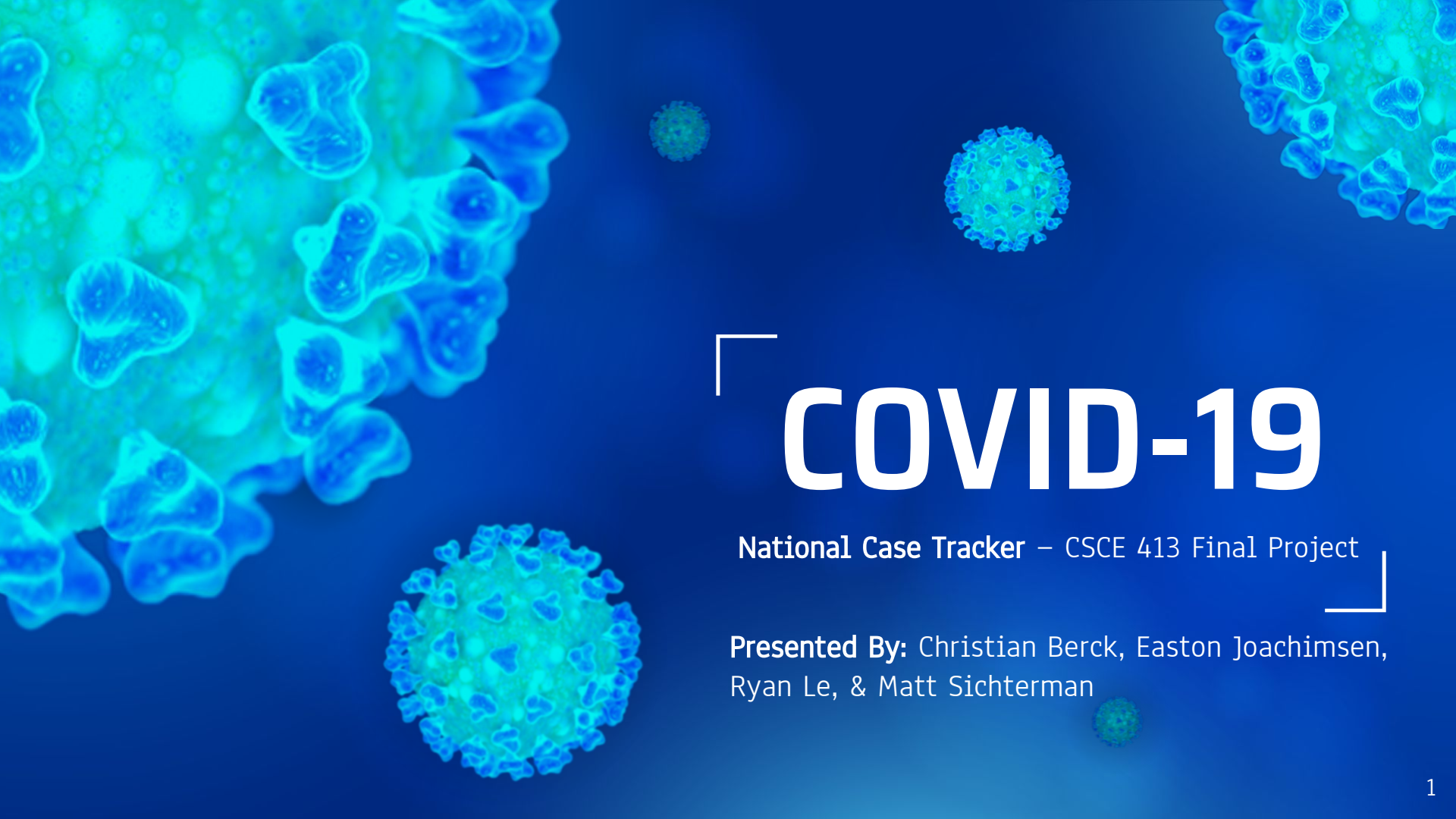


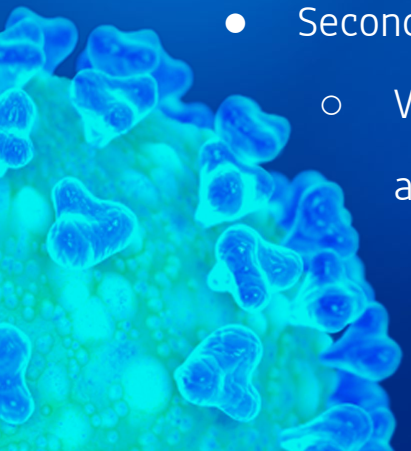
COVID-19

National Case Tracker – CSCE 413 Final Project

Presented By: Christian Berck, Easton Joachimsen, Ryan Le, & Matt Sichterman

GOALS

- Primary Goals
  - Visualize COVID cases as they rise throughout the US
  - Analyze trends and showcase which states have been most susceptible to the coronavirus
  - Determine areas most impacted by the pandemic
- Secondary Goals
  - Visualize potential relationships between state governors' party affiliations and COVID cases over time



01.

DATA SET

Discussion of the John-Hopkins COVID-19 data and other supplemental data that we utilized

02.

DATABASE DESIGN

Breakdown of our database model and decisions made

03.

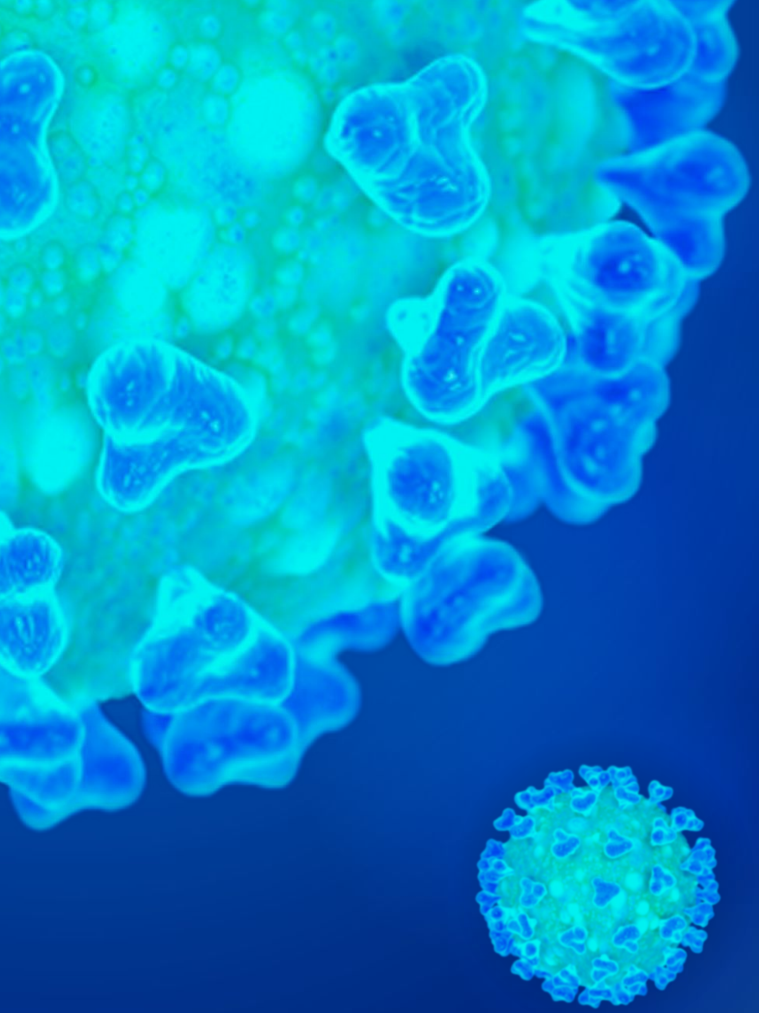
FUNCTIONALITY

Functions to give users insight into our data set

04.

IMPLEMENTATION

How and why we implemented an end-to-end web application



01.

DATA SET

Discussion of the John-Hopkins COVID-19 data and other supplemental data that we utilized

DATA SET OVERVIEW

- Utilized the data from https://data.world/associatedpress/johns-hopkins-coronavirus-case-tracker

- Data included time series data of cases and deaths by states, county level confirmed cases, and time series data of cases and deaths by county
    - Leveraged the the time series data of cases and deaths by state
    - Only leveraged general information about counties

- Limited the time frame of our data between **July 1, 2020** and **November 3, 2020** in order to have a more manageable data set.

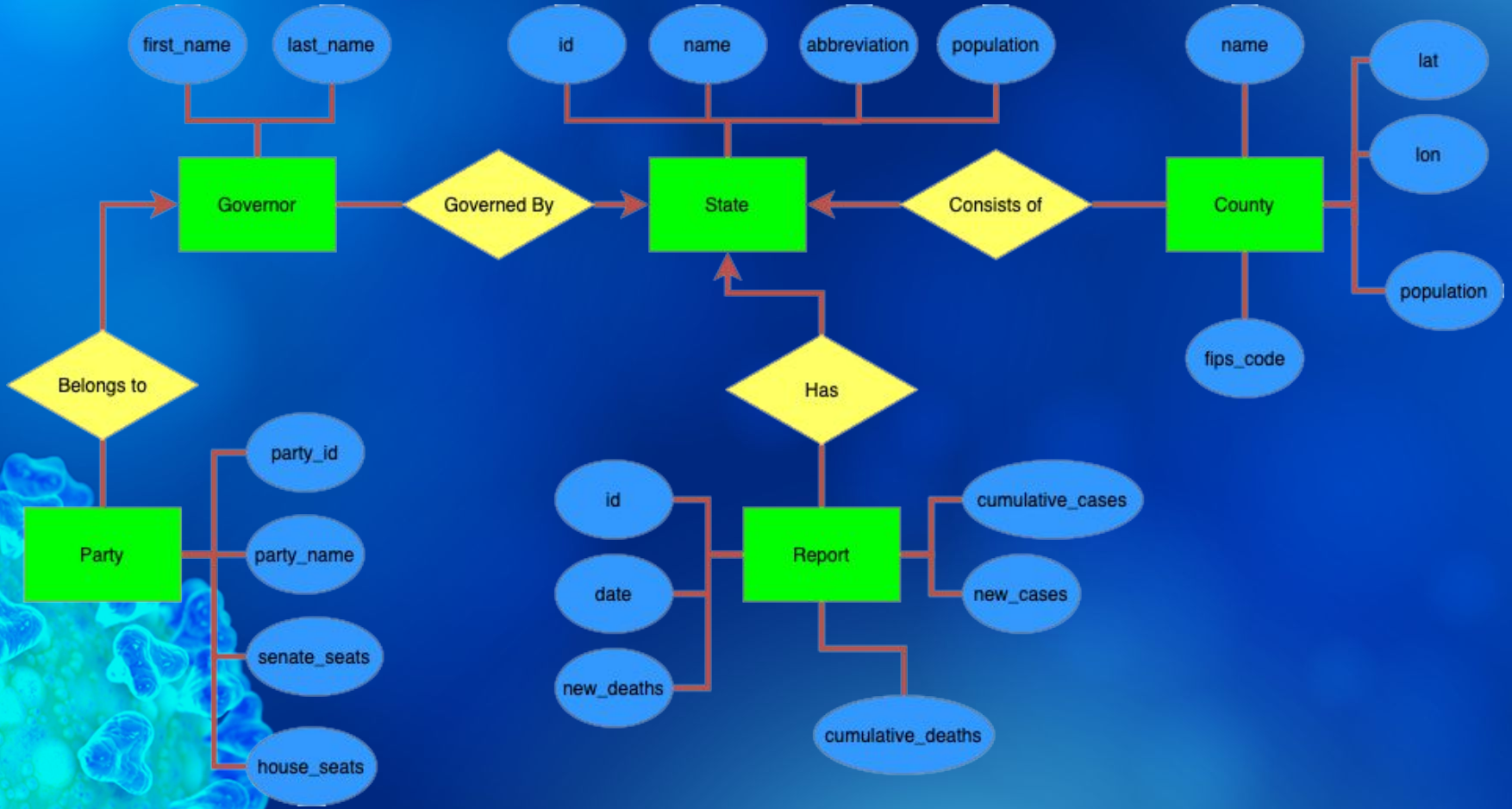
- Supplemented our COVID-19 data with governor information per state and their political parties

02.

DATABASE DESIGN

Breakdown of our database model and decisions made

ER DIAGRAM


first_name
last_name
id
name
abbreviation
population
name
lat
lon
population
fips_code
Governor
Governed By
State
Consists of
County
Belongs to
Has
party_id
party_name
senate_seats
house_seats
Party
id
date
new_deaths
Report
cumulative_cases
new_cases
cumulative_deaths

DESIGN DECISIONS

- Needed to separate our data into groups of tables rather than a mega table as depicted in the spreadsheet
- There were clear groups of data including States, Counties, Reports, and Governors, all relating back to the root State table
- Normalized the data, reduced data reduncies, and created these relationships between our four tables
- To continue normalizing the data, we broke our governors table into two tables: Governor and Party.
    - This normalizes the data and allows us to avoid storing duplicate information for each governor (political party, senate seats, and house seats).
- Thus, our five tables: **State**, **County**, **Report**, **Governor**, and **Party**

03.

FUNCTIONALITY

Three functions to give users insight into our data set

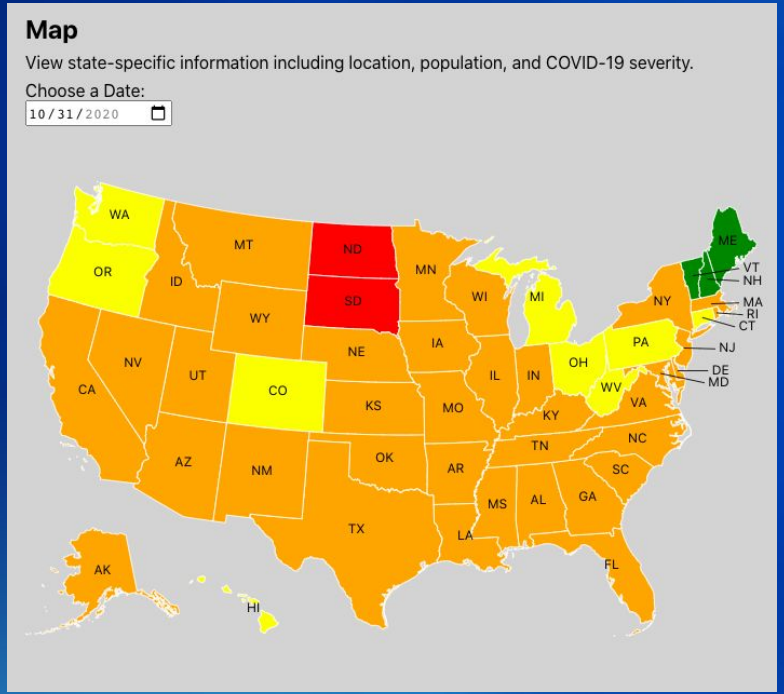
1. Given a date, get state information to populate the map and later calculate the amount of covid cases for each states' population (as a percentage)

Query:

```
SELECT states.id, states.name, states.population,
states.abbreviation, reports.cumulative_cases
FROM states
INNER JOIN reports ON states.id = reports.state_id
WHERE reports.date = '2020-10-31'
```

Description:

Returns data for each state used to visualize a national view of COVID-19 severity for states based on their percentage of population with reported cases for the selected date.



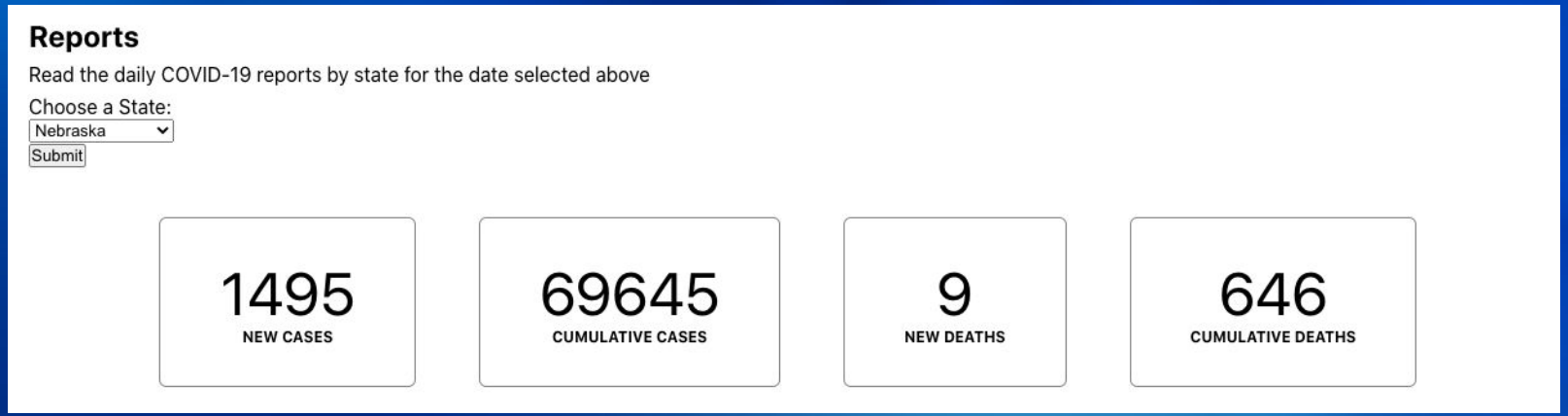
2. Given a state and a date, get the report including COVID-19 cases and deaths

Query:

SELECT cumulative_cases, new_cases, cumulative_deaths, new_deaths
FROM reports
WHERE state_id = ***27*** and date = '***2020-10-31***'

Description:

Returns the report data for the selected state and date, and is then visualized numerically in the web application (uses the same date as selected in the map section).



3. Given a political party, get the amount of covid cases for the population (as a percentage)

Query:

```
SELECT (
        SELECT CAST(SUM(cumulative_cases) AS FLOAT)
        FROM reports
        WHERE date = '2020-10-31'
        AND state_id IN (SELECT state_id FROM governors WHERE party_id = 1)
) / (
        SELECT CAST(SUM(population) AS FLOAT)
        FROM states
        WHERE id IN (SELECT state_id FROM governors WHERE party_id = 1)
) * 100
```

Description:
Returns a percentage of COVID-19 cases for the population of states with governors who are the selected political party. Also uses the date selected in the Map section.

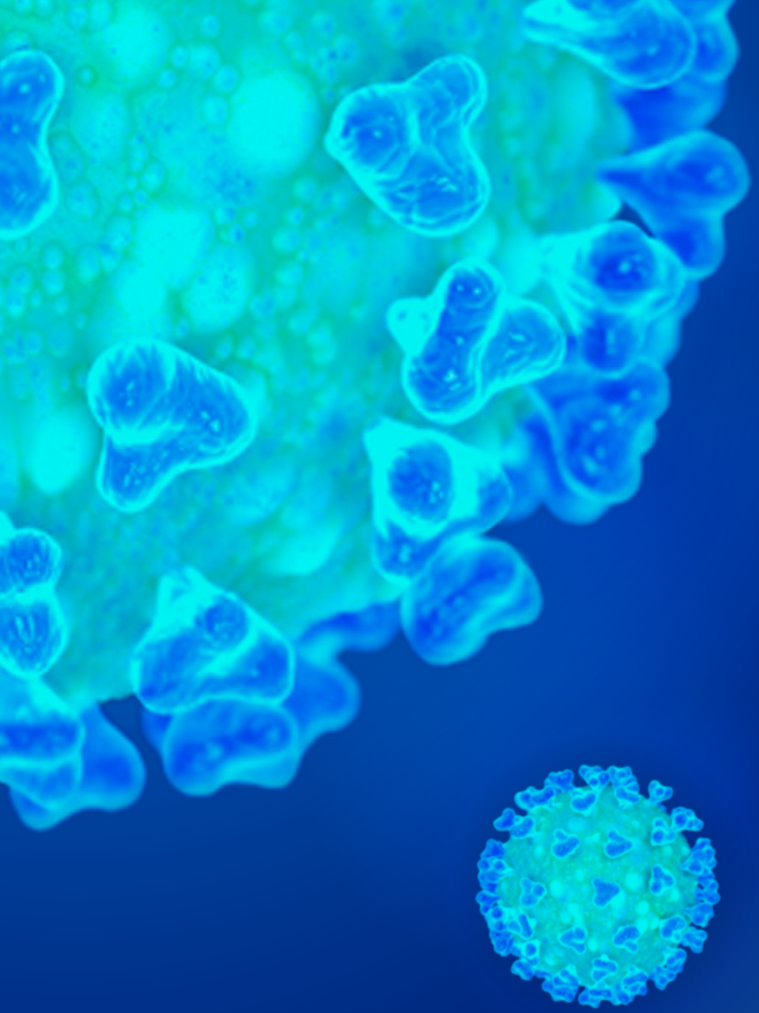
4. Given a state and a date, what are the average cases per county (not shown in our web app)

Query:

```
SELECT (
        SELECT cumulative_cases
        FROM reports
        WHERE state_id = 27
        AND date = '2020-10-31'
) / (
        SELECT COUNT(*)
        FROM counties
        WHERE state_id = 27
)
```

Description:

In order to utilize the counties table, this query takes a state and a date and computes the (approximate) amount of cases per county. This query was not implemented in the web application, but serves as a proof of concept for future work.



04.

IMPLEMENTATION

https://csce-413.msich.dev/

How and why we implemented an end-to-end web application

Overview

- Utilize modern technologies to showcase topics learned in class
  - React frontend
  - Node.js (via Express.js) API
  - PostgreSQL database
- Leverage COVID-19 time series data in order to show trends over time
- Use various queries to give our users the ability to control visualizations and gain meaningful insights

PostgreSQL

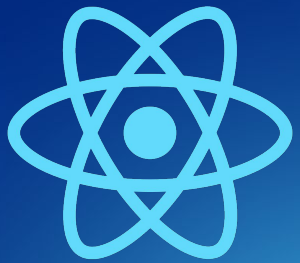
- PostgreSQL is a free and open source relational database management tool.
- After normalizing the data, we loaded each CSV file as a separate data table and established relationships and foreign keys between the tables.
- From there, we integrated our database to our React application using server-side technologies such as Express and Node.js as an API layer between our front-end application and our PostgreSQL database.

Express.js

- Express is a backend web application framework for Node.js that allows for building APIs.
- Our team used Express to build the API to communicate between our React front-end and our PostgreSQL database.

React

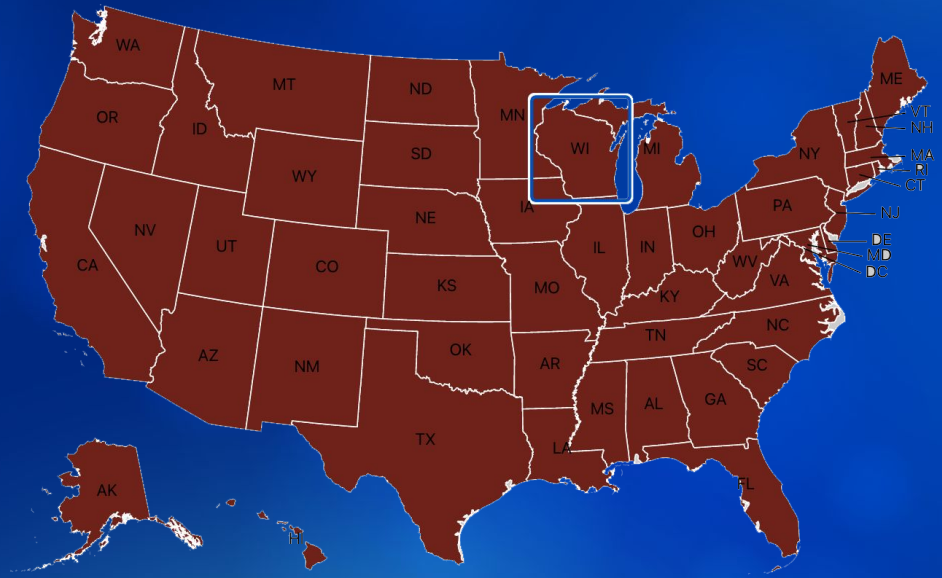
- React is an open-source front-end library for building user interfaces in JavaScript that has become the industry standard.
- React made it easier for our time to quickly package and build our web application, and was a technology that everyone on our team had prior experience with.
- Using our React front-end, users can define their search to be submitted to our API and thus rerender the page based on their selections to show them meaningful COVID insights.



React Simple Maps

https://www.react-simple-maps.io/examples/usa-with-state-labels/

- React Simple Maps is a declarative scalable vector graphic creation library for visualizing map interactions within a React component.
- This library allows us to create an interactive user experience in displaying and analyzing COVID data across the country.



THANKS!

Any questions?

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik