Link: http://52.47.133.119:5000/

Github: https://github.com/wald3r/coronaPlatform

Video: Part of the uploaded archive

Hosting: EC2 Instance – The first year with Amazon Web Services is for free

Used Libraries: Node.js React, D3, React-Vis, React-Simple-Maps, Python Pandas,

Topic: Coronavirus

Data Source: https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data

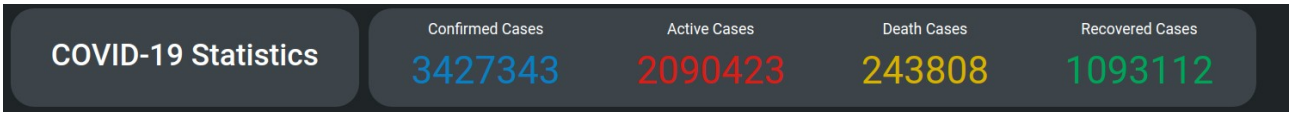
The data gets provided by the Johns Hopkins University. Every day at 4:15 am UTC my application pulls the newest data from the data source. Earlier, or more often is sadly not possible due to the fact that the owners do not update it that often. This leads to the effect that the page is more or less always a bit behind. Especially in the evening.

The data gets abstracted and preprocessed by python scripts. With help from the Pandas library the data gets chopped down to the only necessary bits. In total, three scripts pull the newest data from the data source and generate files, which include total numbers about every single country, summed up numbers for the whole world, and time-series data of every country to evaluate single days.

Interactive Graphs:

Following I describe briefly all parts of my interactive platform.

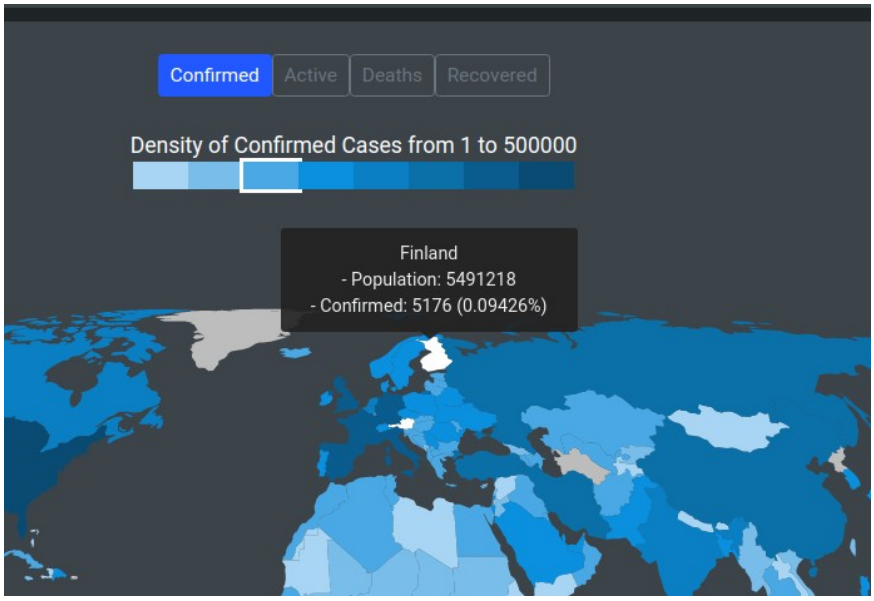
1) At the top of the platform, there can be found the title of the page “COVID-19 Statistics” and the total numbers. As mentioned earlier, the numbers get updated once a day. The numbers give a quick overview of the situation worldwide. The colours are throughout the platform always the same. Blue stands for confirmed cases. Active is red. Deaths are yellow and recovered cases appear in green.



2) The core of the platform is a choropleth map. Choropleth visualizations offer an overview and when it is wished more detailed information. In this case, the map shows the worldwide density of your choosing for each country. With choosing is meant confirmed, active, death or recovered cases (reconfigure interaction). Through hovering over the countries, each country can get highlighted. In addition, the density level above the map gets pointed out as well (elaborate interaction).

While hovering over single countries, a tooltip appears and provides extra information (elaborate interaction). The information includes the name of the country, the number of cases in total numbers and amount of cases in comparison with the population.

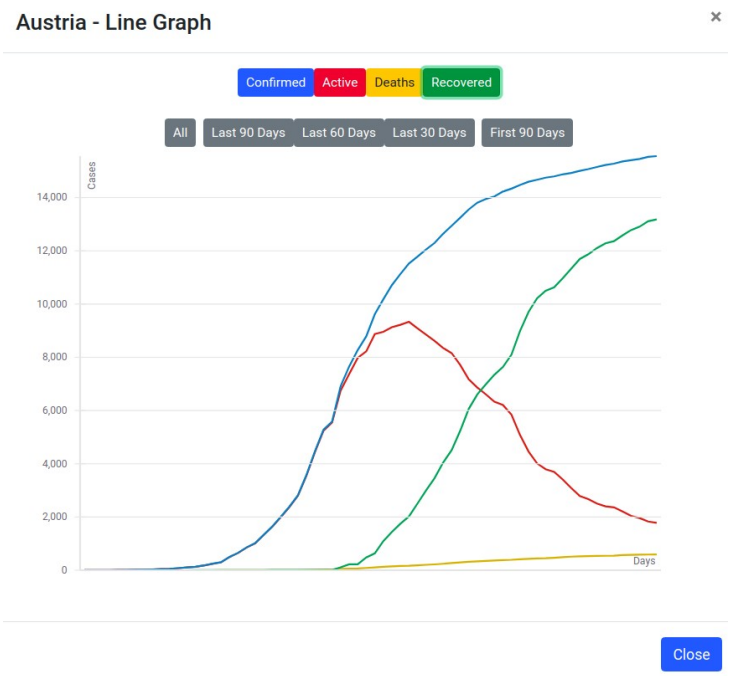
As a little help, a table with all cases per country is located to the left of the choropleth map. I placed it there because the choropleth doesn’t show immediately any numbers per country. With the table, it is possible to get a clearer overview of the worldwide situation.



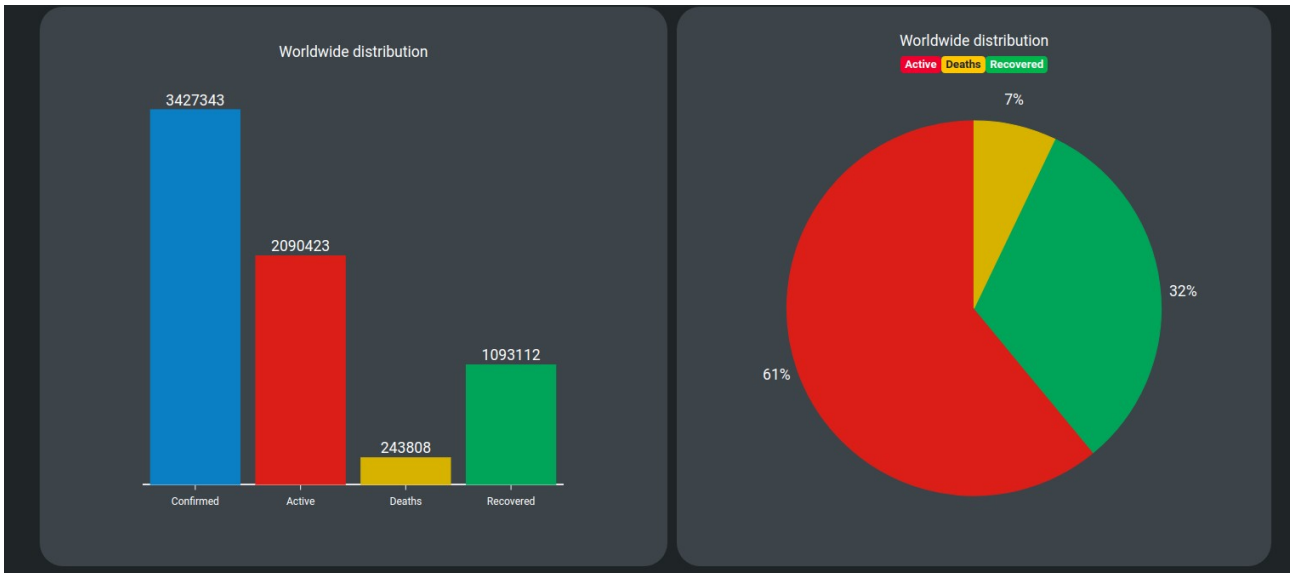
The data source of the choropleth map is a GeoJson file from:https://github.com/zcreativelabs/react-simple-maps/tree/master/topojson-maps

3) With a click on a single country of the choropleth map (explore interaction), an additional graph gets opened. A line graph about the chosen country. The opened line graph shows a time series of all cases since the first case appeared in that country. Via reconfigure interactions it is possible to add or remove certain categories like confirmed, active, deaths and recovered cases. Included are some filter interactions to move the point of focus a little bit. It is possible to move the focus to the last 90 days, last 60 days, last 30 days, first 90 days or all days.

I removed the time series data on the x-axis. The more data gets added, the more the single days along the x-axis get pressed together. At some point, it would not be readable anymore. As a solution, I implemented a crosshair. When a mouse pointer hovers over the line graph, a crosshair appears and blends in the date and all cases numbers of that day (elaborate interaction).



4) A bit below the choropleth map, a bar and pie chart can be found. Both are not interactive but provide an overview of the worldwide distribution of the Coronavirus. All graphs are on this platform are designed with trying to keep a good Data Ink Ratio. Like here, I removed all frames and only display what is necessary. This gives users the possibility to extract information easier.

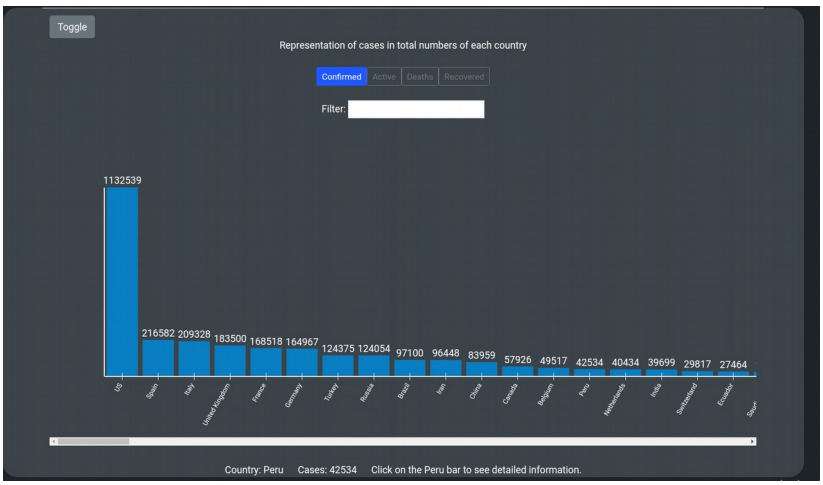


5) On the bottom, a bar chart can be found. The chart shows all cases of every single country. The order is from the biggest to the smallest amount of cases. As with the choropleth and line graph, the bar chart can show all confirmed, active, deaths or recovered cases (reconfigure interaction).

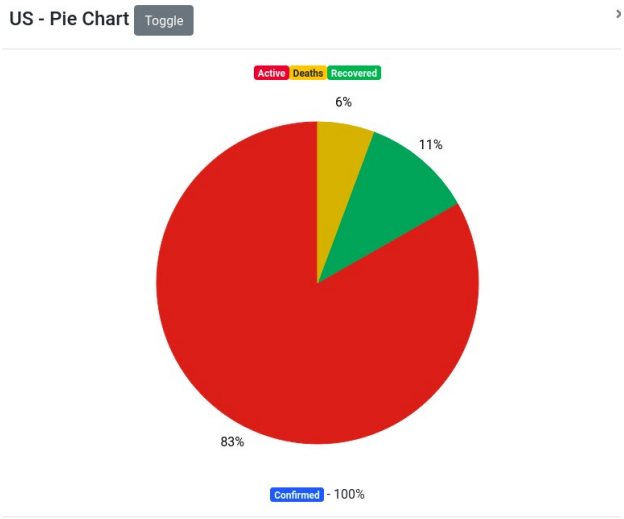
As one of many additions, there is a filter interaction function. It is possible to filter two specific countries, which makes it possible to compare those two countries directly.

Another addition is that via a toggle button (reconfigure interaction), it is possible to change the display of numbers from total numbers to a comparison with the local population.

When a mouse pointer hovers over a single bar, the country gets pointed out below the bar chart. A highlighted bar would have been preferred but React-Vis is stubborn here.



6) Through a click on one of the bars (explore interaction), it is possible to get additional information about a country. A window pops up and shows a pie chart, which displays the number of cases in %. With a click on the toggle button (reconfigure interaction) the display changes to total numbers.



Lessons Learned:

1) I have never heard of the file format GeoJson before. Somehow I managed it all my life without it. Through trying to implement a choropleth map, I stumbled over it and I was immediately able to implement my map with this file format. It is straightforward and includes all the important information.

2) A valuable lesson I learned was the visualization of data on phones. To make it accessible on phones was never part of my original plan or the given task. Although, I tried to make my platform responsive but many visualizations are not able to scale. It is partly useable now via a tablet or phone but still far away from being useful. In the future, I will see if there are libraries which are more applicable to such tasks.

3) New was as well to build such a platform at all. Visualizing data in manner, that it is interactive and still keeps a focus on methods like “Data Ink Ratio” or the “Information Seeking Mantra” was not always easy to achieve.
https://github.com/zcreativelabs/react-simple-maps/tree/master/topojson-maps

Conclusion:

Overall, my platform provides all the necessary information about the Coronavirus clearly and comprehensively. Although, the generated data worldwide for the Coronavirus is massive and because of that there are much more possibilities to visualize the data. Even with the data from the Johns Hopkins University many other different kinds of graphs would have been possible. But I restricted myself to not flood the page with too many different graphs.

Furthermore, I think it is hard to plan at the beginning the very outcome of of such a project, when you barely know any of the used libraries. Which is why I did not plan much at the beginning. All that was planned was that I wanted to use python for preprocesssing, a platform about Corona and a choropleth map to visualize it. I did all three things at the end and the additional services were added throughout the passed time.