**d18124788 - Liam McKenna**

**dt249 - Internet Application development Assignment**

-  
Please see README.MD for APP for help install/setup instructions.  
If there is an issue with the API timing out, I have the same data in JSON in the database folder for each table.

**Table of Contents:**Overview.  
Goals.  
TECH STACK.  
HOSTING.  
GOOGLE PERFORMANCE AUDIT.  
INITIAL FEATURES.  
DESIGN/ Site Map.

**OVERVIEW:**An overview of the project can be detailed as a full-stack (aka a 3-tiered client/server architecture) website application. The web app will need to present statistical information related to the Covid19 virus in various countries in a front-end UI accessible to the public and store this data on a backend database server.

**goals:**

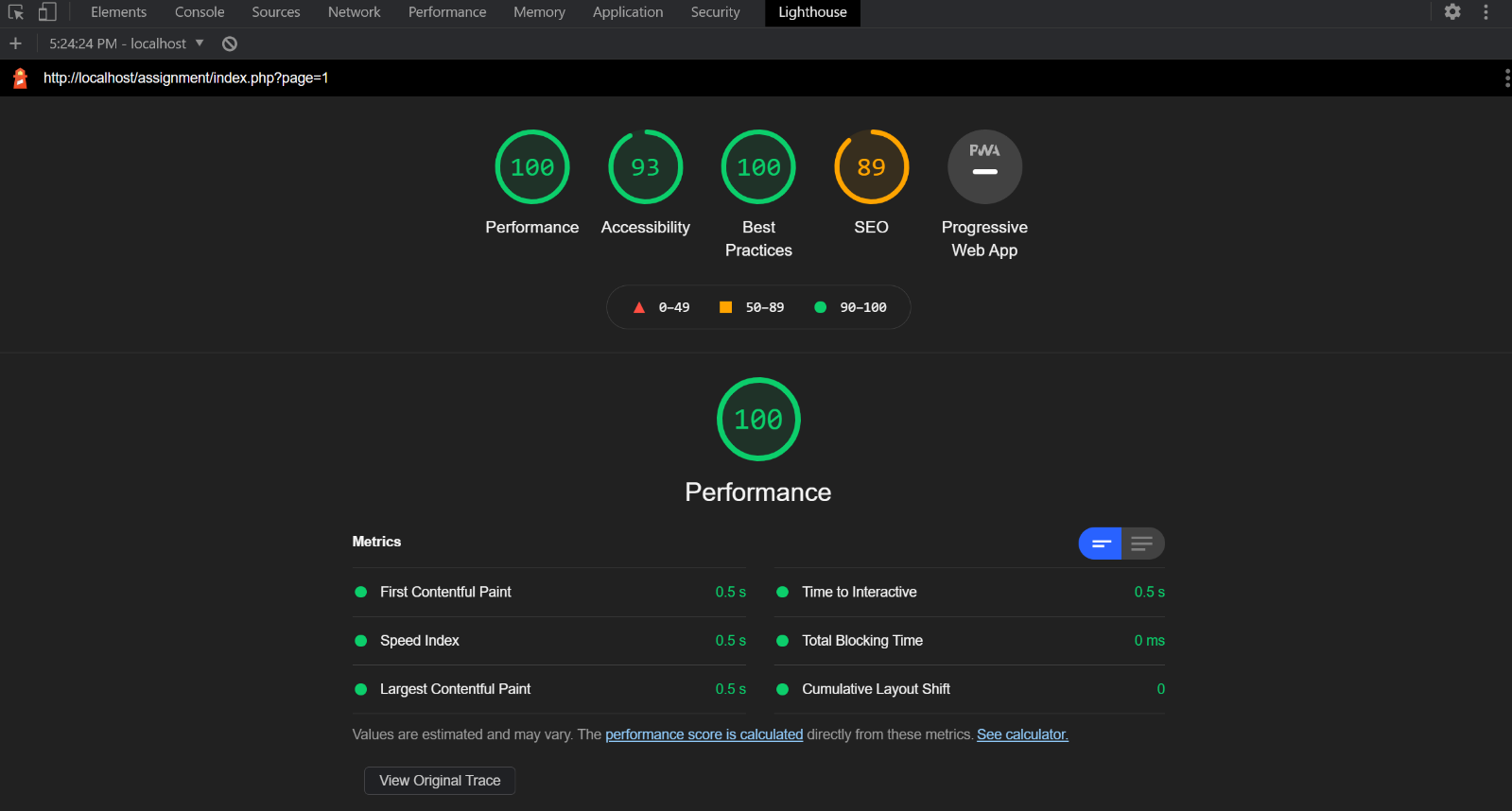
* Build a user-friendly web UI to allow clients to access the information they want and present the covid19 data in a easily readable format.
* Build searching and filtering tools to make it easy to find data.
* Build and host a database to store the data required for the Website UI.
* Initial instruction is to use a dataset provided by the lecturer; however intention is to find a suitable dataset API online that can be used to populate the database.
* Create a user signup and login system to access restricted pages. Initial instruction from lecturer is to use this to facilitate an administrator access to alter functions on the website, however further discussions have found that this may not be suitable. Instead, allowing clients to signup and login to the website to access extended data may be more suitable.
* Ensuring the website is secure from potential malicious attacks such as SQL injection so that user data remains safe and the data cannot be altered or destroyed by a unauthorised party.
* Ensure the web application is mobile friendly across multiple devices.

**Tech Stack:**XAMPP, PHP, HTML, CSS, Javascript, MySQL  
For this project, I intend to keep with the above technologies. Given the module has focused on these It would make sense to apply the knowledge that has been taught.  
  
Setting up of the developer environment will be with Visual Studio Code for the front end and Datagrip for the backend.  
  
I initially wanted to use newer adopted technologies such as Node.Js, React.Js, tailwind, Typescript, Sass and even thought about looking into even higher prototype tech such as Web Assembly with Rust. However, while this could cause a problem by diverging from the source material of the module, my main reason to focus on the classic web stack was to expand my knowledge on how a full stack web application will be developed with technologies like PHP and possibly understand why the industry has started to move away from this kind of tech stack to modern stacks such as MERN/MEAN.

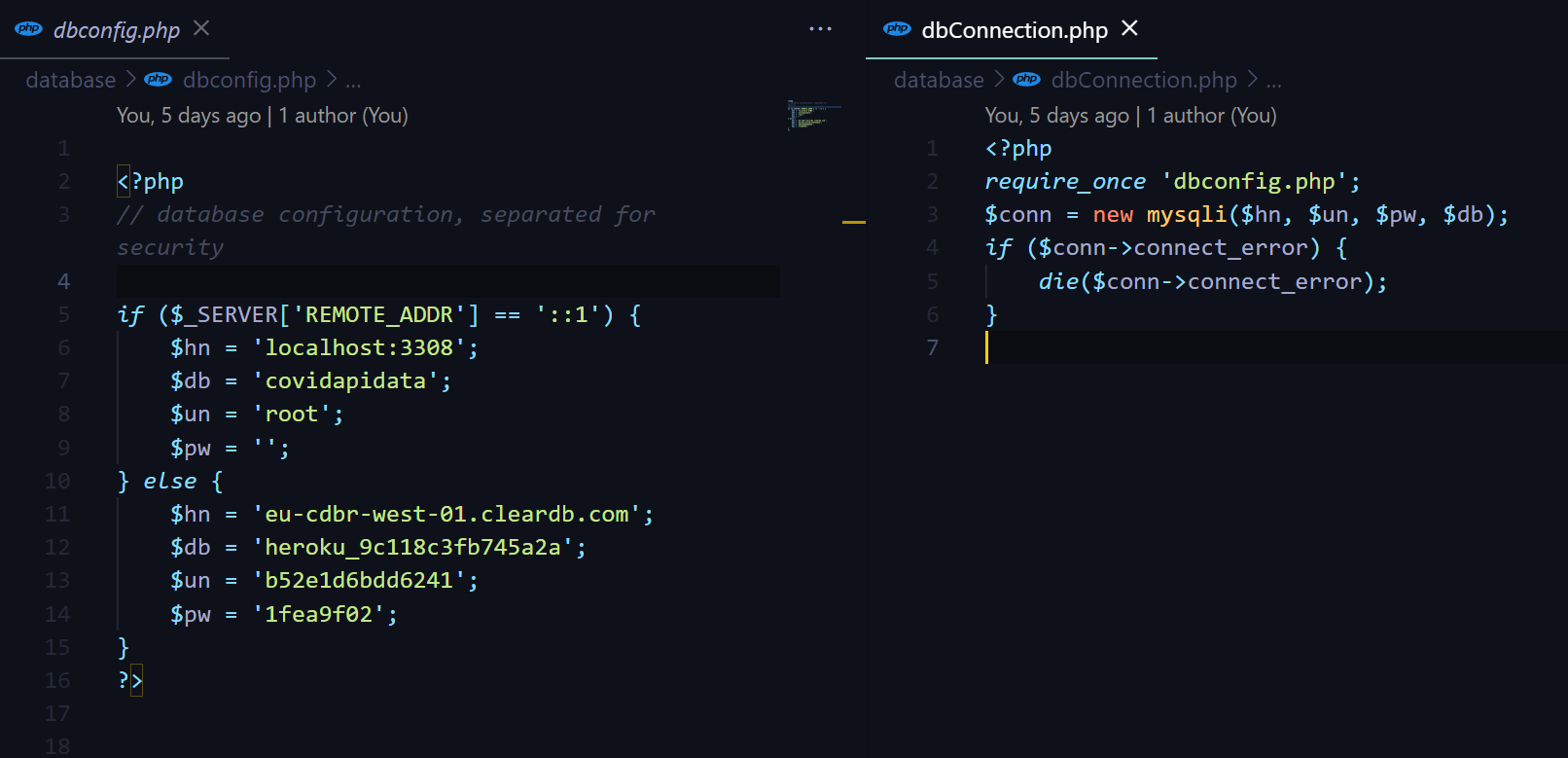
**HOSTING:**Hosting the application I could possibly use the Microsoft Azure platform as this is the only cloud hosting system that’s been taught to us in another module. Or alternatively I think a more suitable hosting solution will be using Heroku as my research has found this to be better suited to PHP applications. This will require more research to confirm if this is a better hosting platform after the application is in a state ready to be hosted.

**Google PERFORMANCE AUDIT:**

I used Google’s website performance and best practice auditing software called Lighthouse to evaluate my web application and was given the below score which helps to identify this application as a best practice build and performance.

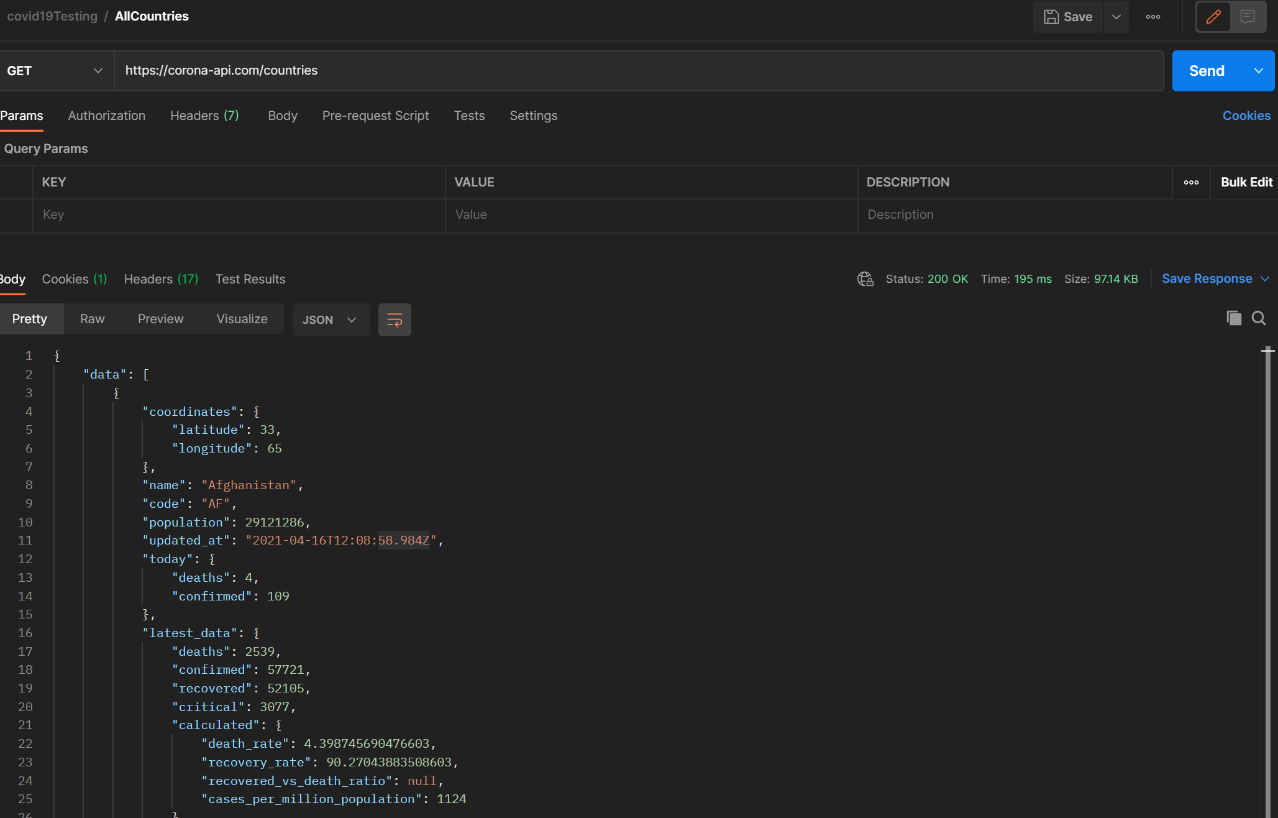
****

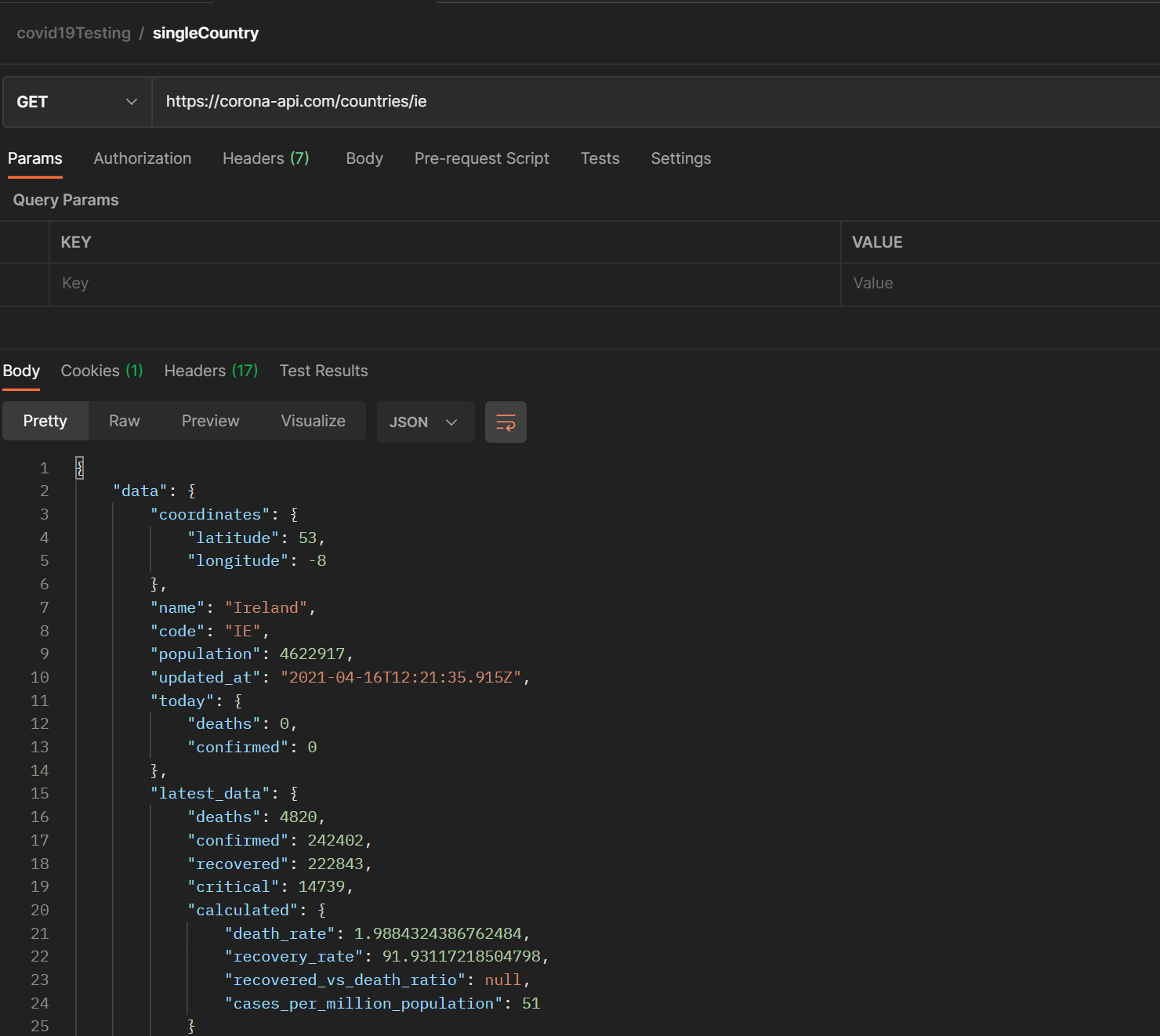
**INITIAL FEATURES:  
Database Connection to historical Data.**I’ll need to research how and create a connection from my front-end website to my backend database to allow MySQL queries to be performed and the data returned to my front-end website.  
  
**Answer:** I’ve been able to create this connection in php by using the mysqli extension. I’ve moved my variables into a separate file for security and have added a if check to swap between connecting to my local database if the IP address stored in $SERVER[‘REMOTE\_ADDR’] or if it is on my Heroku hosting platform then it will connect to the Heroku Database.

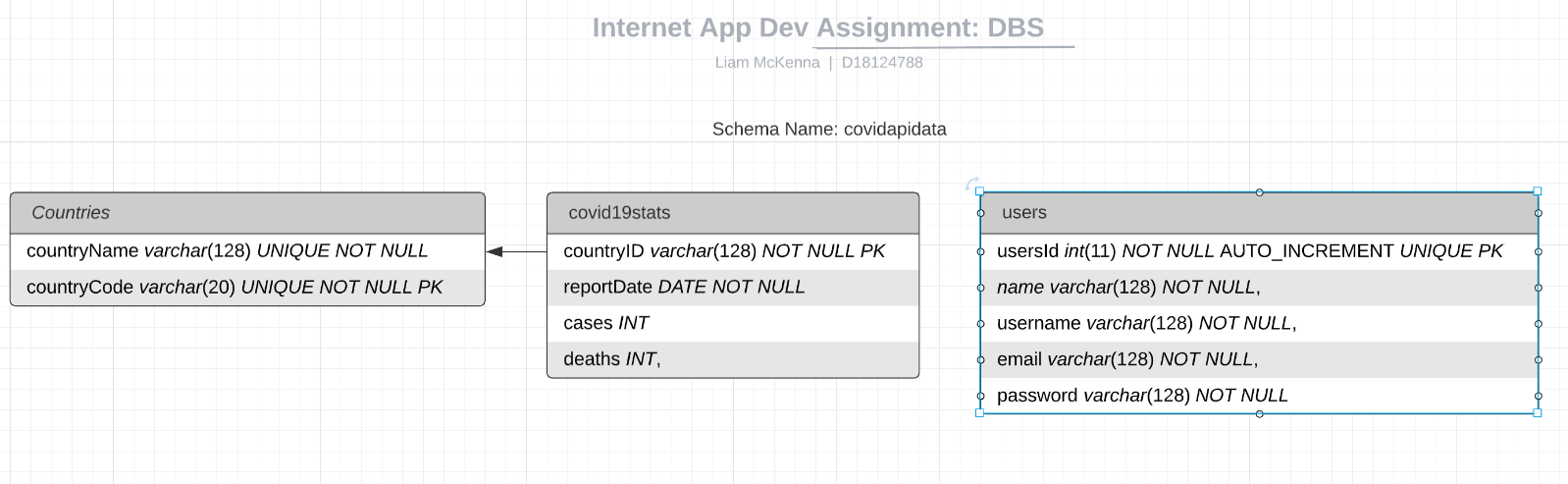


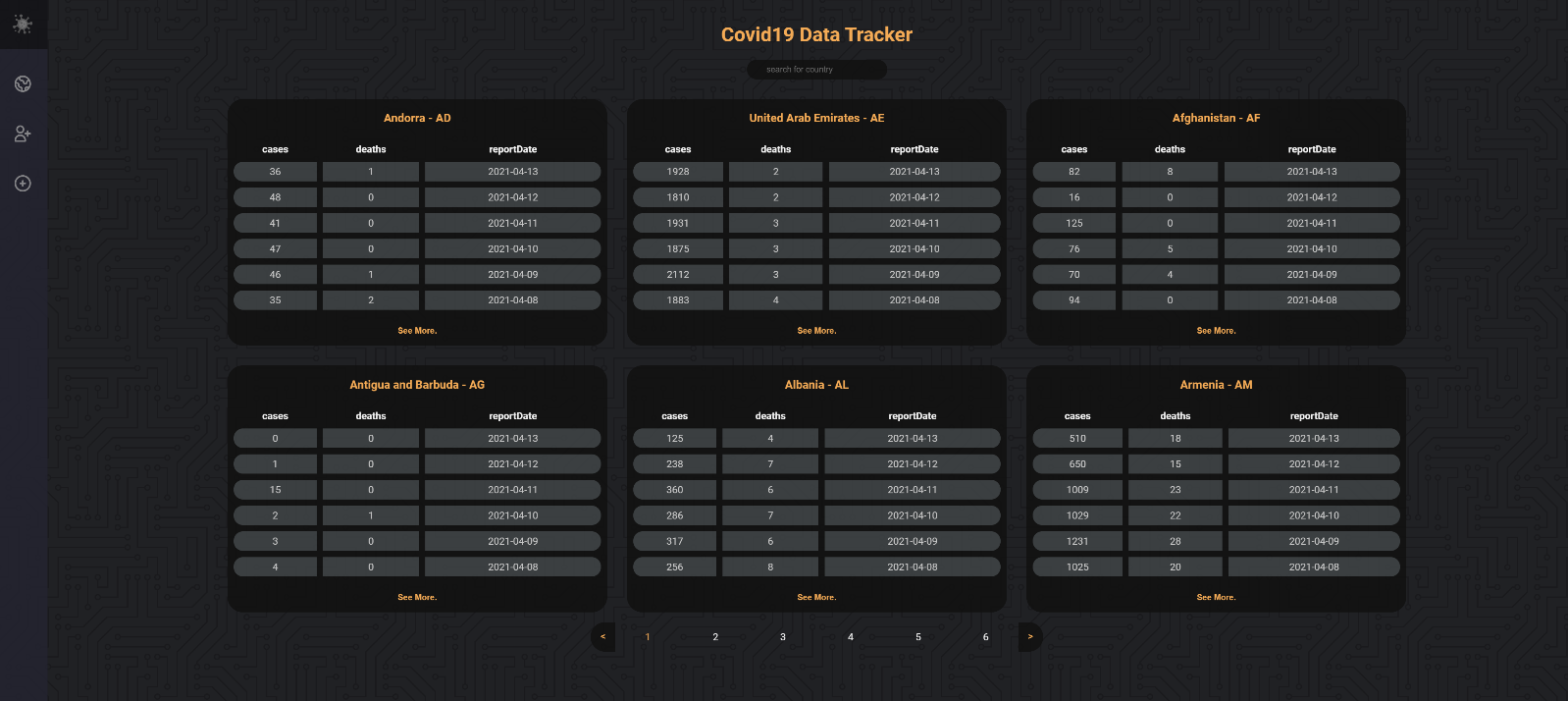
----

**Get Covid19 data from an API and post to database.**Initially I will be using a pre-made dataset from an excel document, however I intend to move over to a public API to gather the Covid19 data. The lecturer has advised this process will be explained in a later lecture.

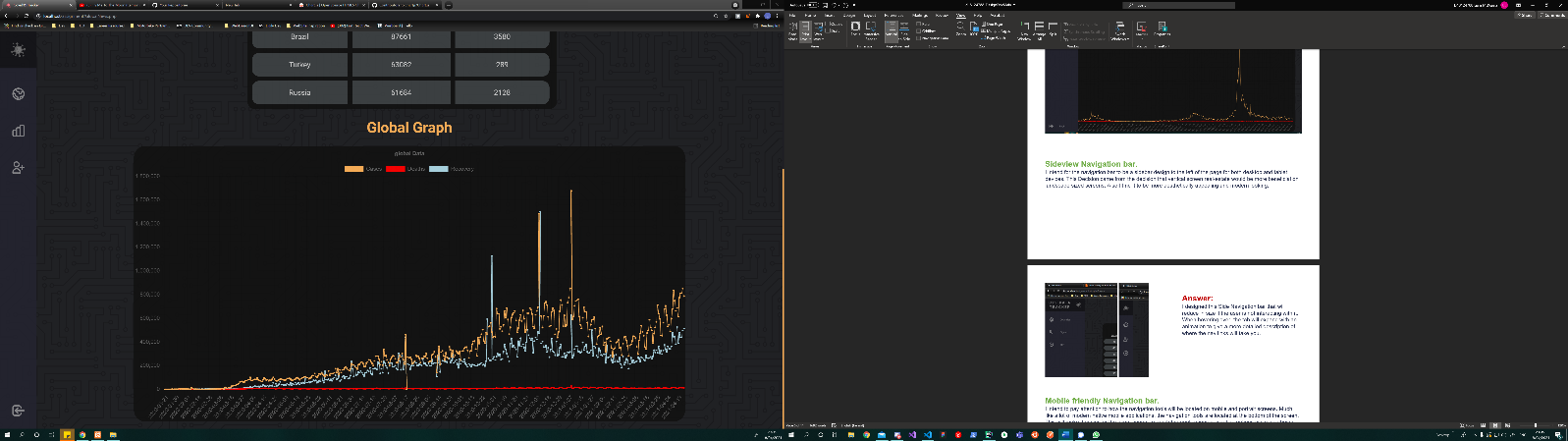
**Answer**:  
I initially mapped the provided dataset in the XAMPP SQL database, however after much searching for a viable pubic API and dataset, I found a freemium API at <https://about-corona.net/> that includes updated daily data for the past 4 months.  
  
I used [Postman](https://www.postman.com/) to test and evaluate the endpoints and data that I required:  
 **Get All Countries with Data:** [**https://corona-api.com/countries**](https://corona-api.com/countries) **I used a get request on this endpoint to return a JSON list of every country that the API stored data on. I stored this data in my database in a table called ‘Countries’ with columns named ‘countryName’ and ‘countryCode’. The code would be required when querying the API for the full dataset of each Country: **

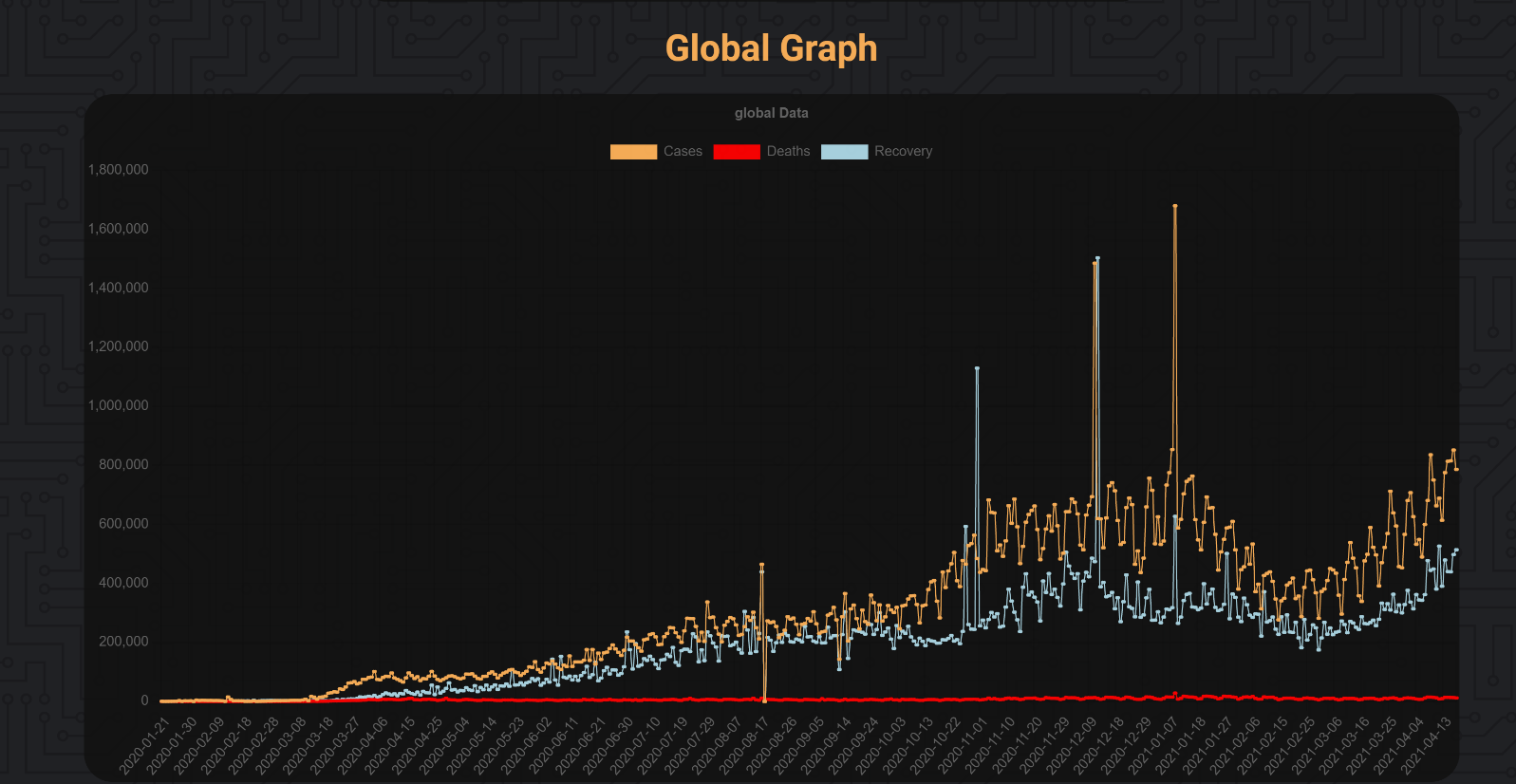
**Get Full data for a single Country:** [**https://corona-api.com/countries/ie**](https://corona-api.com/countries/ie) **I used a get request on this endpoint to return a JSON list of a single country that the API stored data on. I stored this data in my database in a table called ‘Covid19stats with columns named ‘countryID’ , ‘reportDate’, ‘cases’, ‘deaths’. Each country has a ‘timeline’ object which stores the record history for cases and deaths each day.**

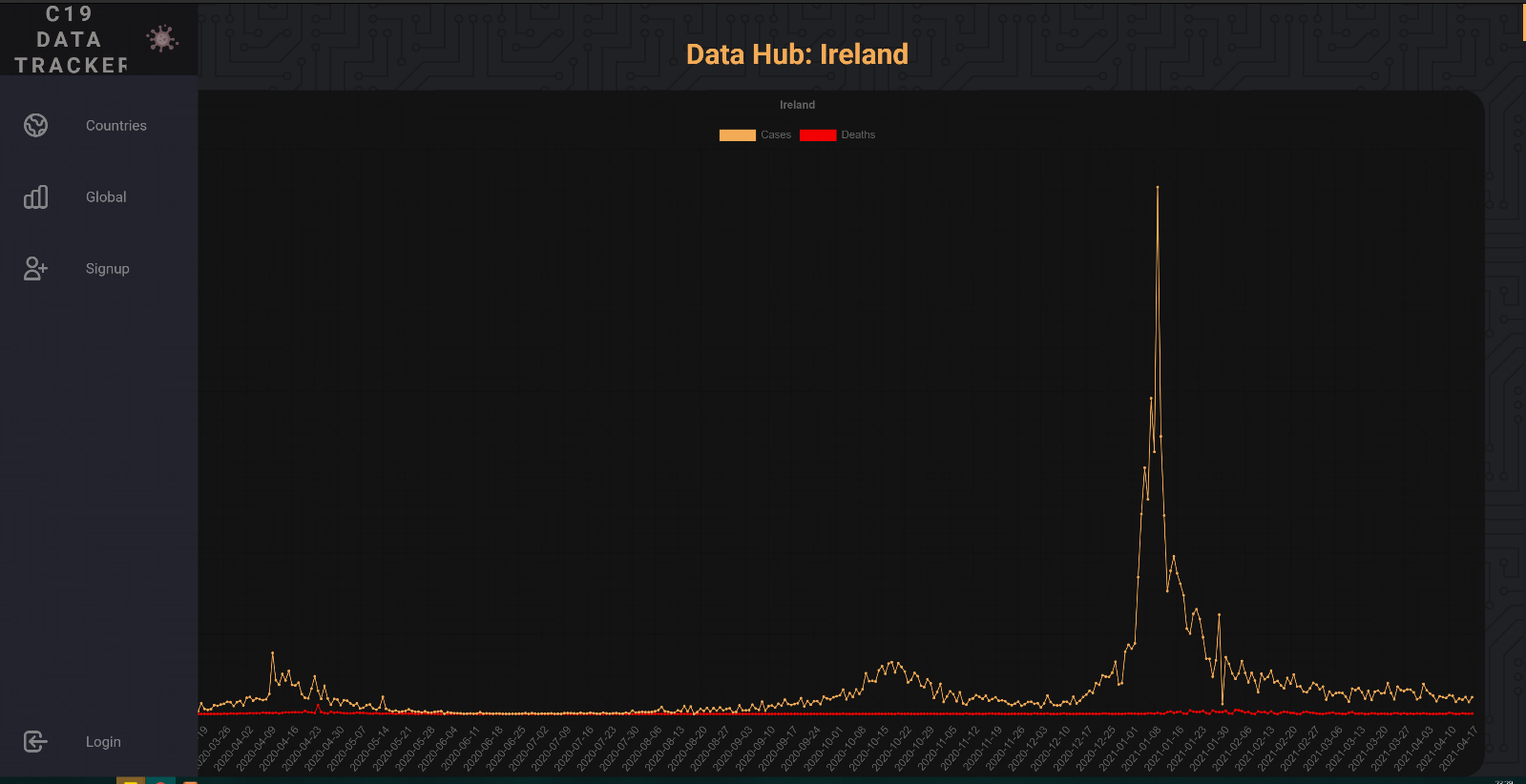
**Here is my Database Schema UML:**  


**Modular data preview Cards for each Country with a ‘see more’ option to get full details.**The data retrieved from the database will be organised and categorised by Country. The home view will have a ‘preview’ view of a limited amount of data for each country, and there will be an option to ‘see more’ to view the complete list of data on a new page. I intend to make the page dynamically populated depending on what countries are available in the database. This will ensure I don’t need to create unique page manually for every country and allows for reusable code.  
**Answer**  
  


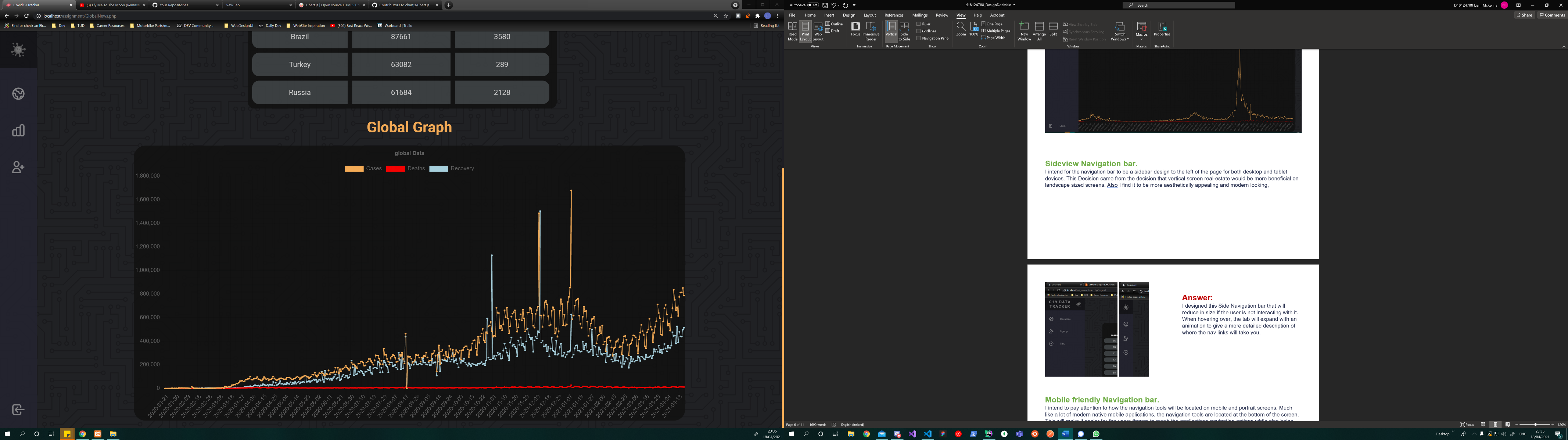
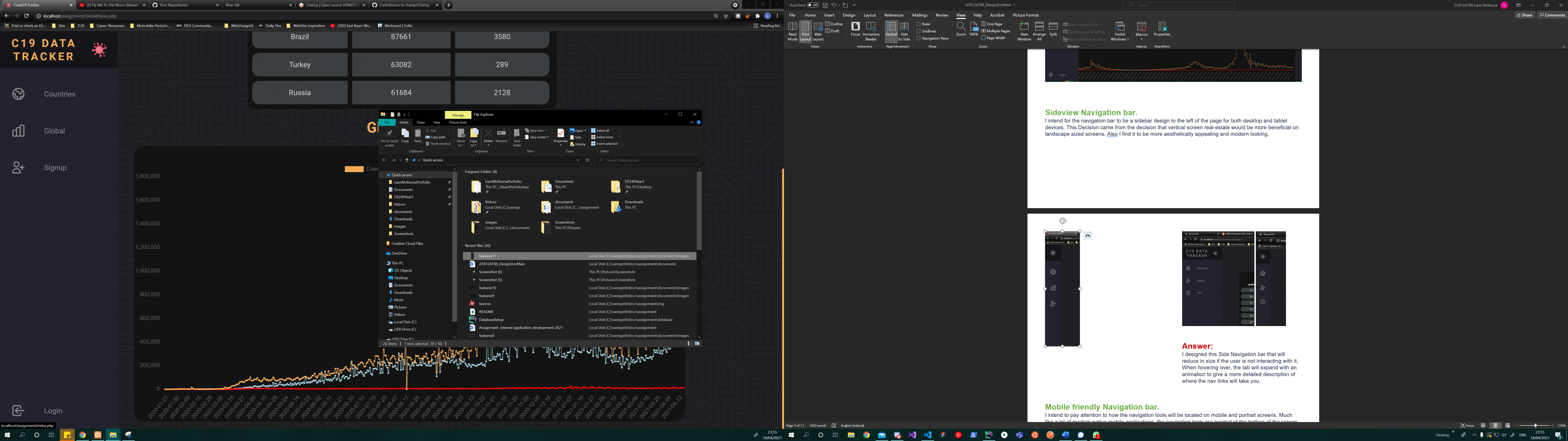
**Data Charts and visual information.**I intend to have visually appealing charts and infographics visible in the countries pages to give a better understanding of the data associated with each country.

**Answer:**I found a open source Javascript library on github call [Charts.js](https://www.chartjs.org/) that allowed me to import modules to create Line graphs with my data. The graphs are interactable and allow the user to click the heading to turn on/off each of the lines. Also each node will show data when the user hovers their mouse over them.





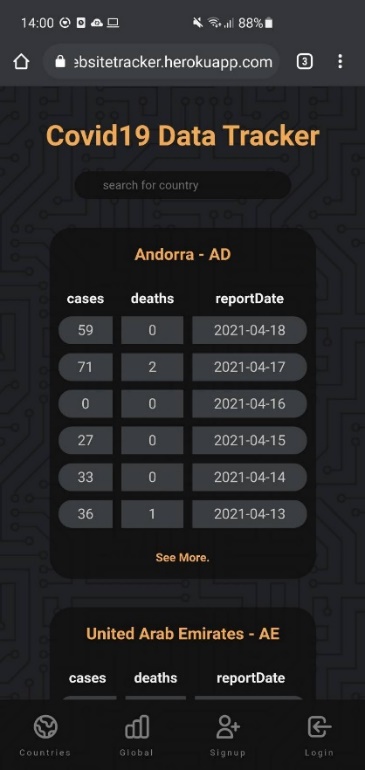
**Sideview Navigation bar.**I intend for the navigation bar to be a sidebar design to the left of the page for both desktop and tablet devices. This Decision came from the decision that vertical screen real-estate would be more beneficial on landscape sized screens. Also I find it to be more aesthetically appealing and modern looking,

**Answer:**I designed this Side Navigation bar that will reduce in size if the user is not interacting with it. When hovering over, the tab will expand with an animation to give a more detailed description of where the nav links will take you

**Mobile friendly Navigation bar.**I intend to pay attention to how the navigation tools will be located on mobile and portrait screens. Much like a lot of modern native mobile applications, the navigation tools are located at the bottom of the screen. This will make it easier for the users fingers to reach the applications navigation options while also being more aesthetically appealing.

**Answer:**



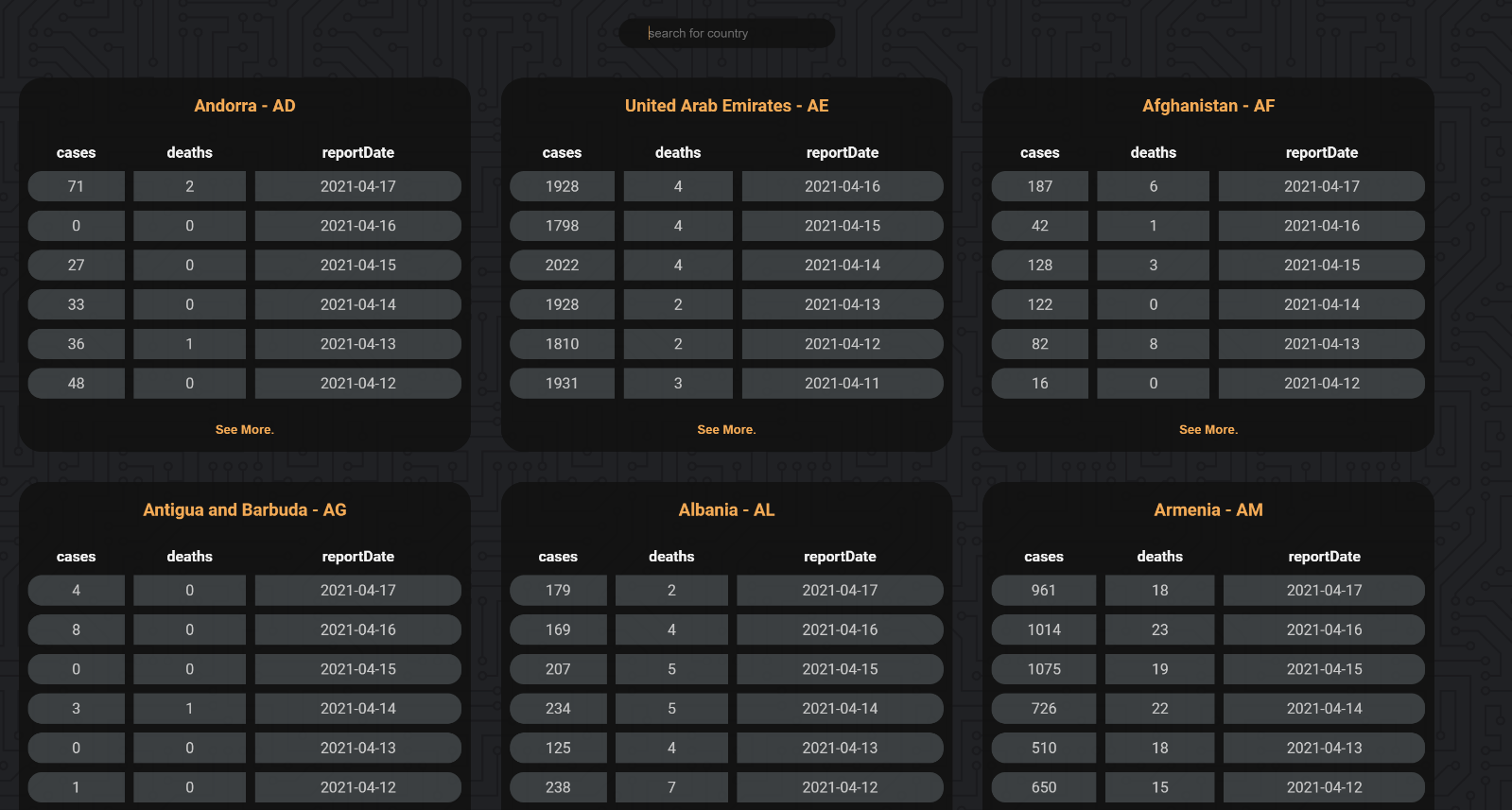
**Page content paginator.**To make sure that there is not too many countries loading on the main page and to make sure that each country doesn’t load too much data, I will create a paginator system to only show a restricted amount of data and a page counter and option to move forward and backwards through the pages of data. This will help to stop the users page becoming way too cluttered and large, it will also prevent loading issues where there may be a large amount of data being retrieved and presented.  
**Answer**

The Paginator was tricky to build but using a combination of native PHP and javacript I have been able to make it work as below. The paginator will only show 6 pages, and one less than the current page if the current page is not already on the first.

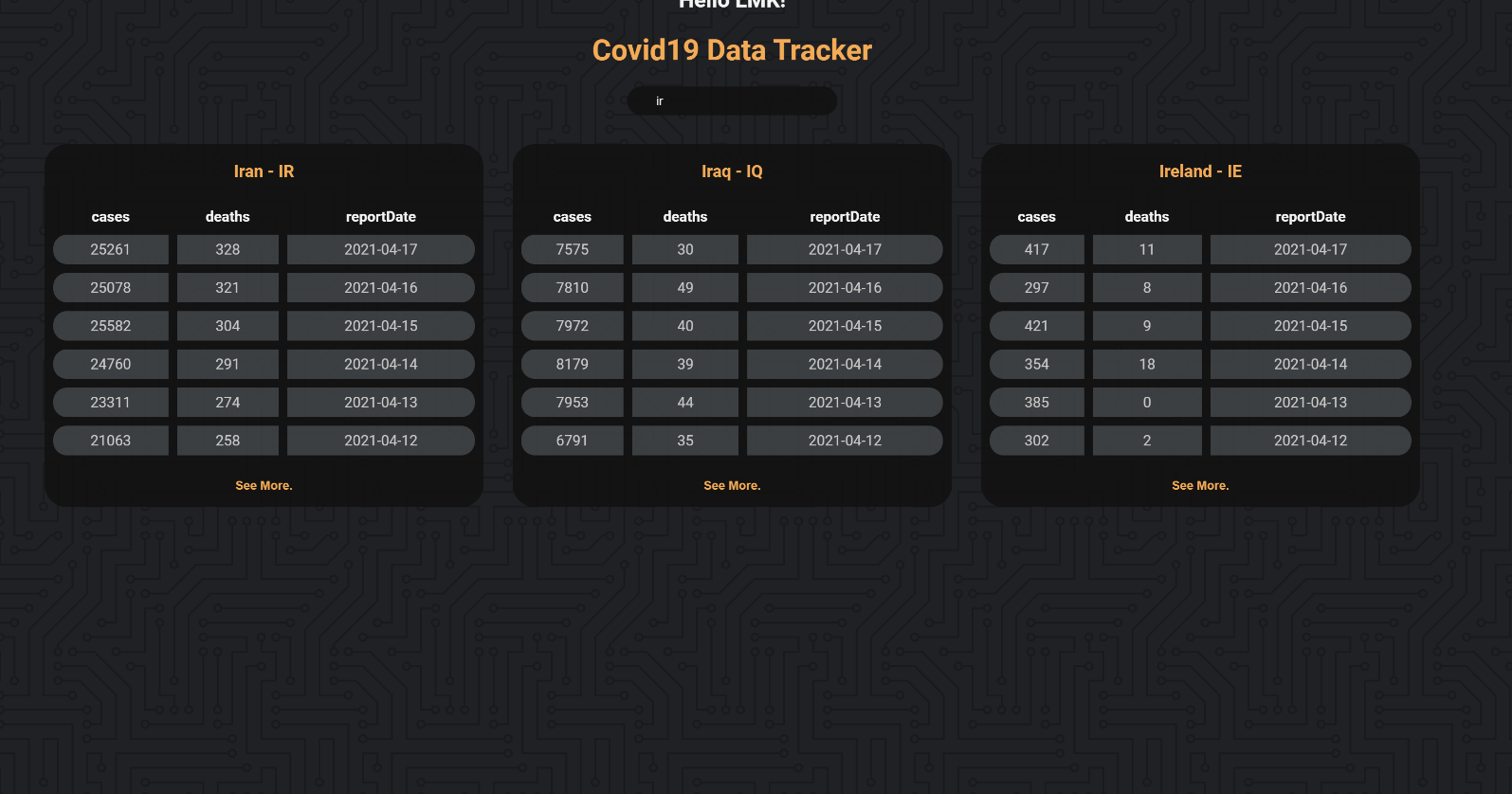
****

**Country search function.**I would like for a search function on the main page that will allow the user to search for the country they want to see the data for. The search function would be able to give auto finishing options depending on what letters have been entered in by the user, much like how Google tries to give suggested search options.

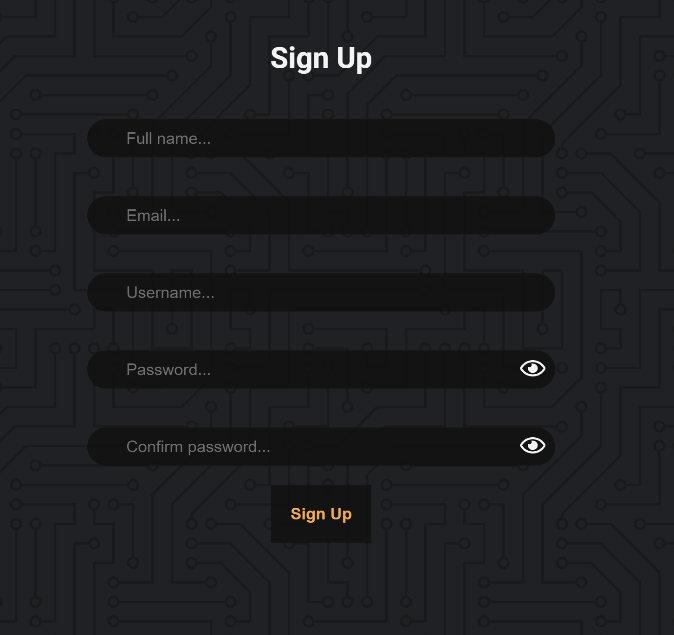
**Answer**

WI used AJAX to create the search function. However I decided to make the entire country card dynamically update to what the user was entering in the search bar. This gives realtime search results as the user enters the countries they want to access. I used a javascript event listener to update the page for every letter that is entered. Below is a comparison without any search input, and then with the letters ‘ir’ entered, the cards will instantly show results for countries that start with those letters  


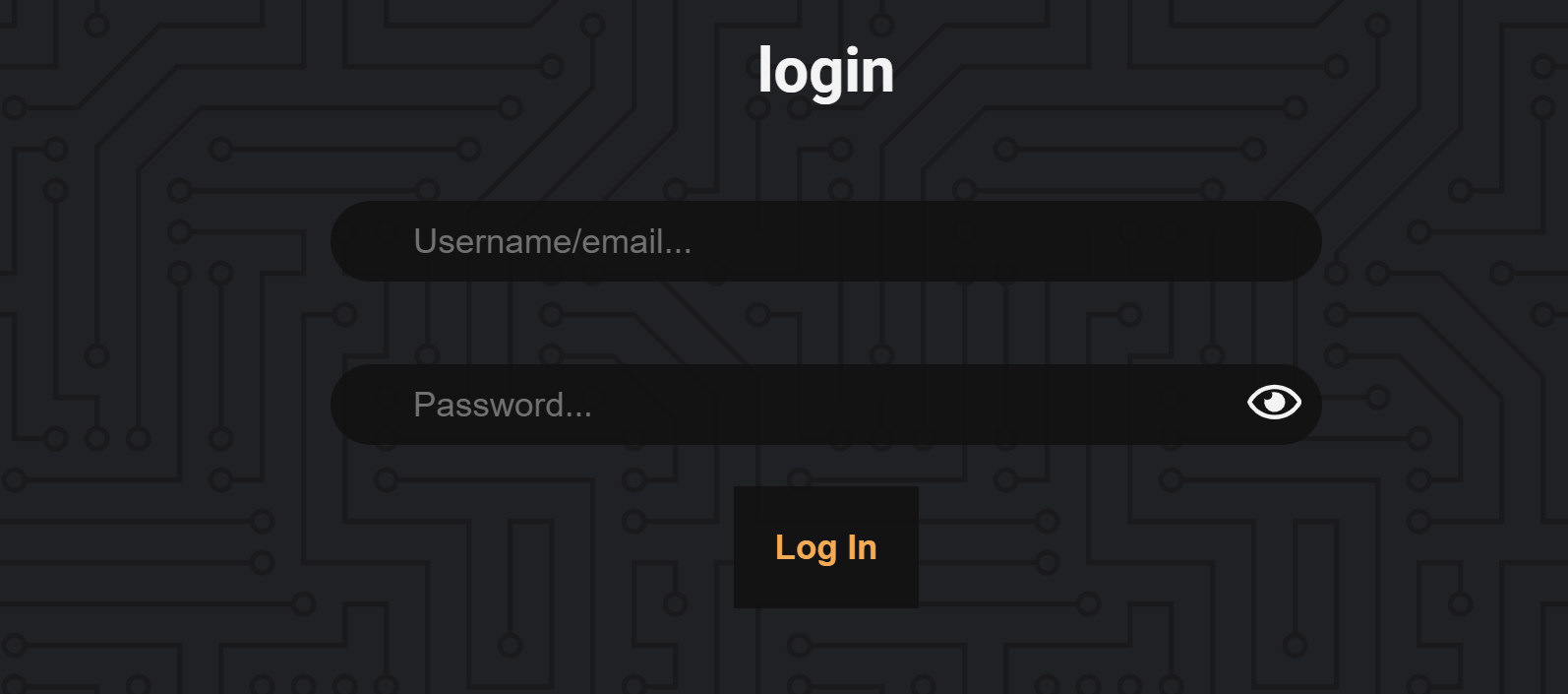
**Search input will change country results for each event using Ajax**

**Client account signup form.**I will have a signup form to allow users to create an account which will give access to private information or private pages inside the application. The intention is to showcase the ability to implement such a feature, however the actual functions given to the user may be partially arbitrary as the nature of a covid19 data aggregation don’t really require for a user account or admin system which was discussed with the Lecturer.

**Answer**I created a signup form that creates user accounts in the database. The form is checks for Valid email formats and forma validation with a full spectrum of detailed error messages if something wasn’t entered correctly.

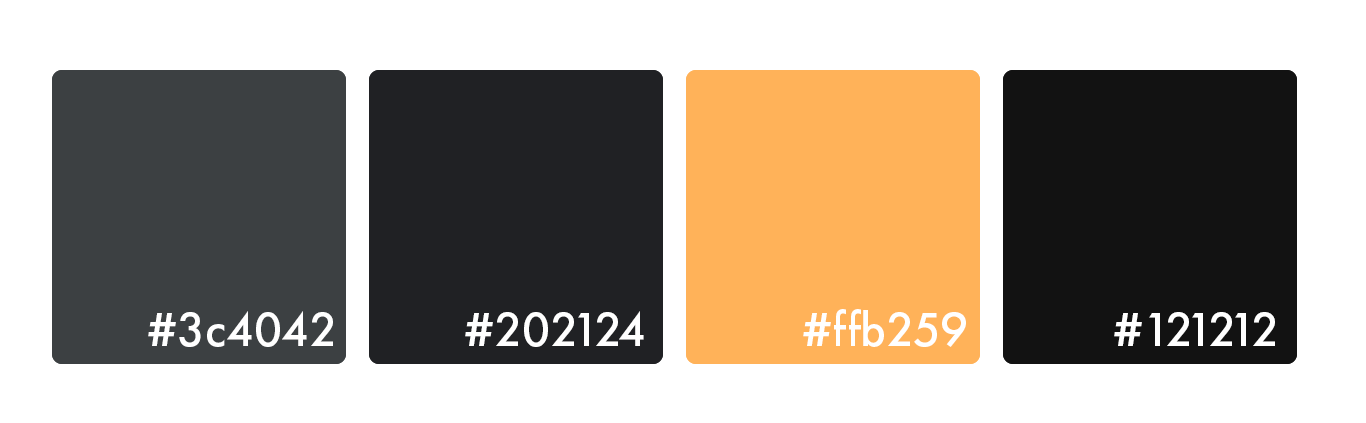


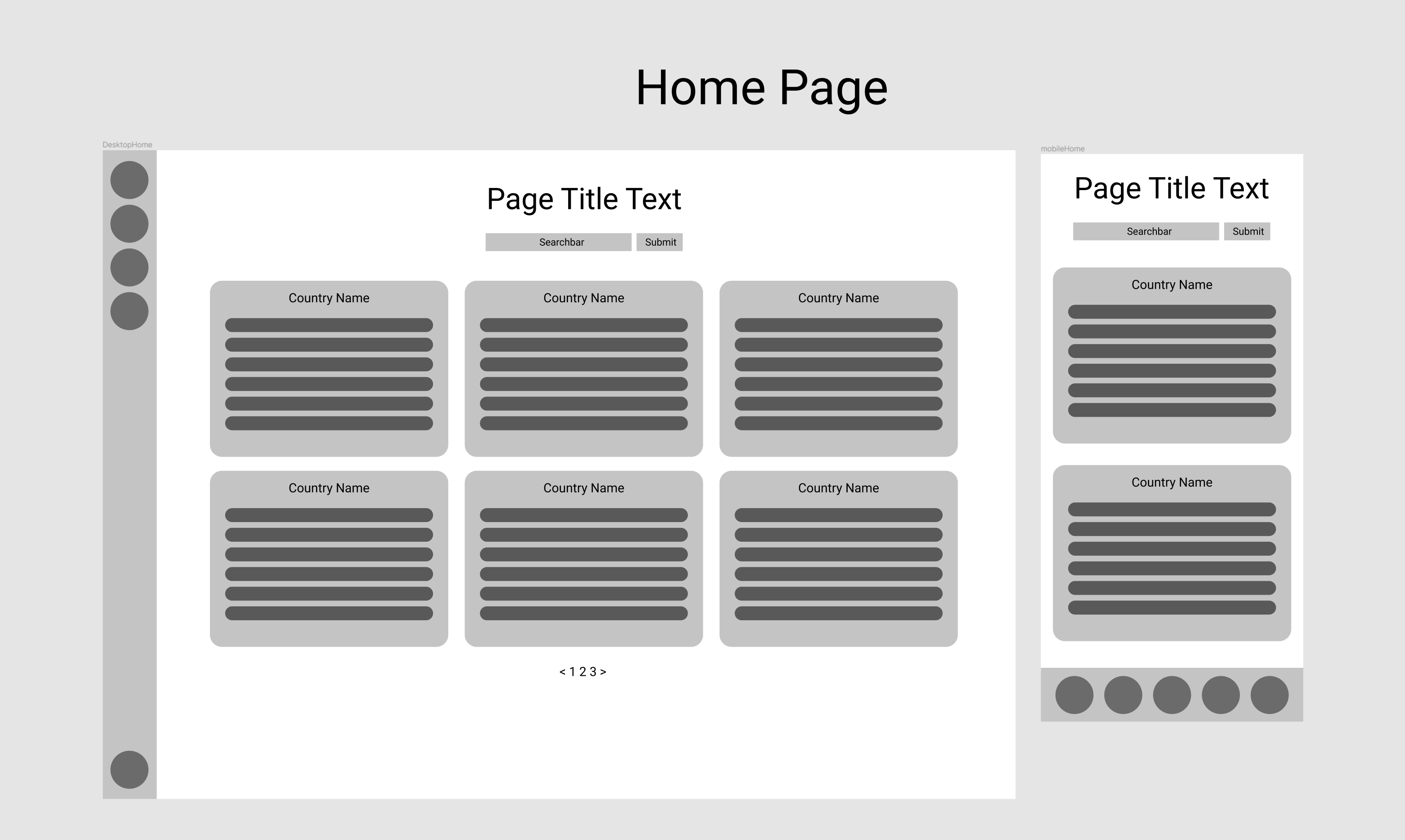
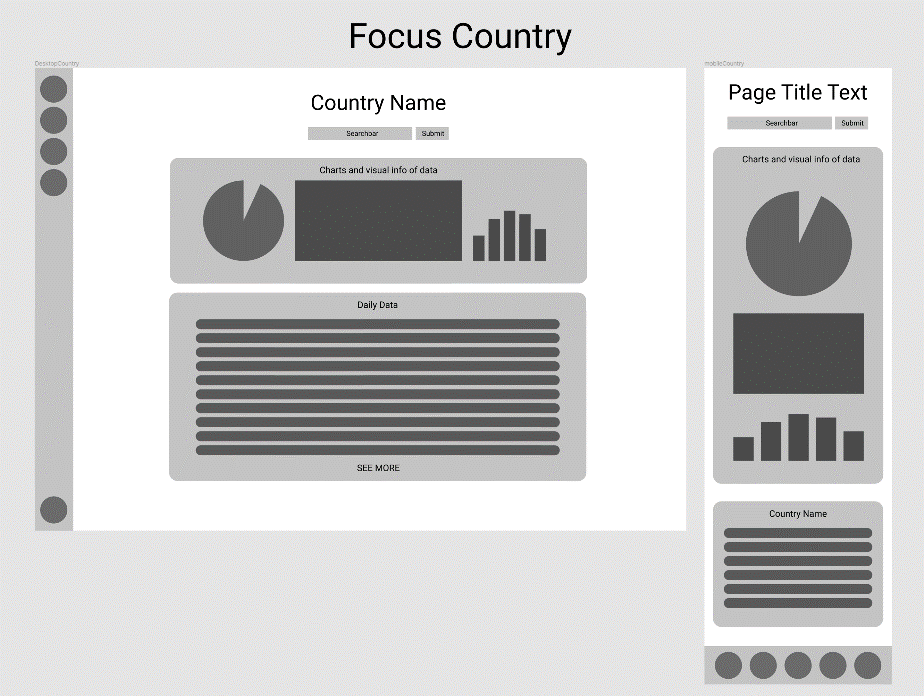
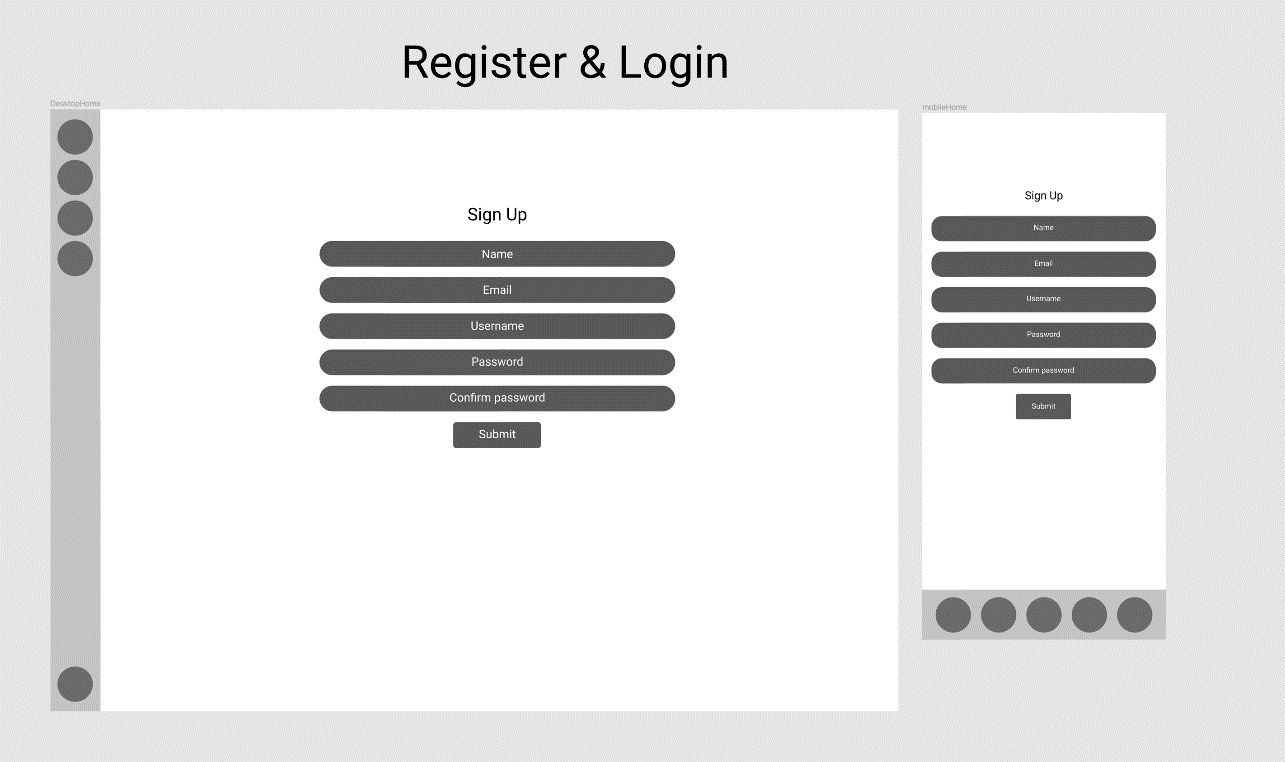
**Client login system with private access features.**this will support the Client account signup form in allowing the users to regain access to their account in the application after they have created an account with the sign-up system.

**Answer**I created a login form and system that will log the user into the application to give access to a “admin settings” tab that will allow the user to update the data to the current date.

**Cloud hosting.**Cloud hosting is important to research and implement as this will ensure there is site reliability, and the service is always available to the users. Further research required as there maybe pricing issues that will prevent this service.  
**Answer**My searching found that Heroku and ClearDB were the perfect hosting platform, however the free tier database did not allow me to store the full data content for the application and my budget did not allow me to by additional servers. However with some limited testing data I was able to host a bare version of the application which allowed me to test some key front-end features in a hosted/live environment.

**Css styling and animations / Dark Mode.**I would like for good use of Css styling to ensure the data and UI is presented in a user-friendly manner with also a possible option for a dark-mode colour palette to help users with issues to bright light and to help prevent device battery wastage.   
**Answer**I decided to go with the below Colour Pallet:



**DESIGN & SITE MAP**  
********

**ADDITIONAL THINGS i learned:**While this has been my first Web Application and the entire contents of this project has been a first time learning experience, the below stand out as things that took longer focus of my time and interest to entire they where implemented correctly.

**Ajax:**  
Ajax has allowed me to action database requests based and update them live to the DOM on event actions instead of reloading the entire page. This has been useful to reduce the amount of REST API requests and database queries that are being performed unnecessarily. And to show live search results to the users.  
Resource:<https://www.youtube.com/watch?v=tNKD0kfel6o>, <https://www.youtube.com/watch?v=XhMGV8PzyOg>  
  
**Prepared Statements:**  
a way to protect the data from users before sending the query to the database, preventing malicious attacks to the database.  
 **Regex:**  
I used regex notation to ensure the username is a correct format when creating an account.

**Postman:**  
I used Postman to test the API endpoints and navigate through the data to find the values I require

**Hashed passwords:**  
I used hashed passwords to add an extra layer of security for the user passwords stored in the database, so that even if there was a data breach, the passwords would still be protected.