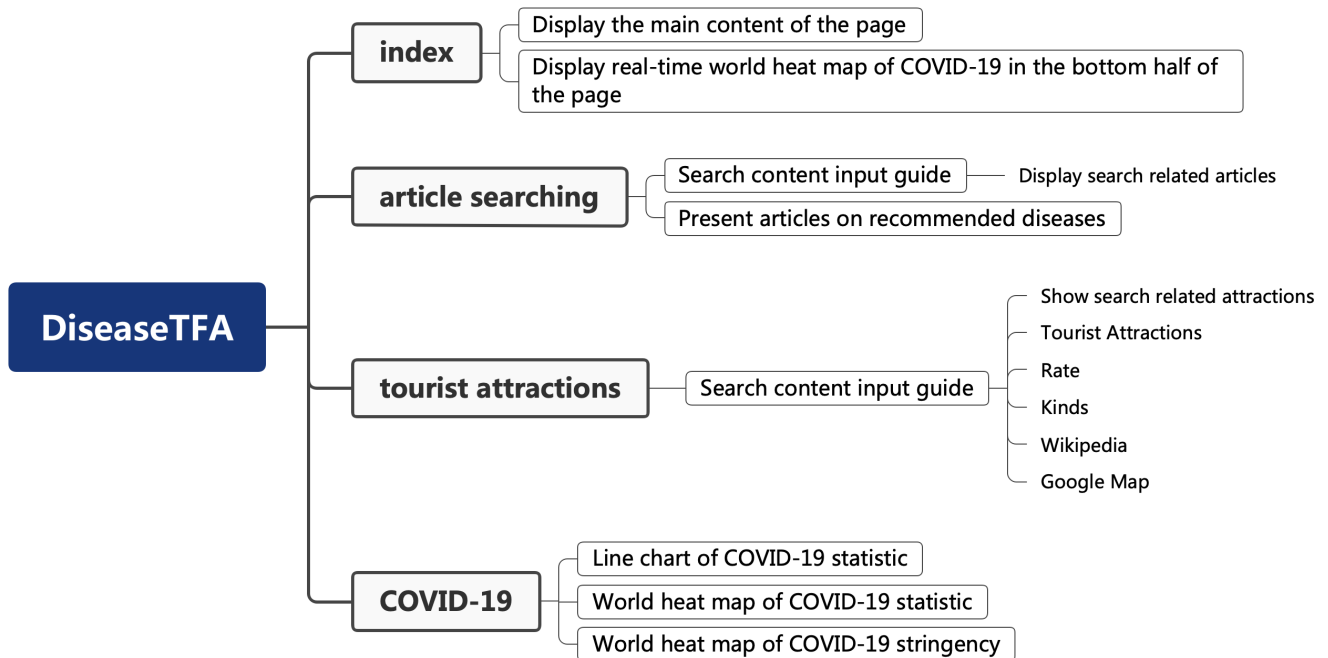


D4 Final Report

Repository Link: https://github.com/JackWhoooo/SENG3011_GroupName

Analytics Platform



Requirements

Requirement 1

The client can find and access all disease reports related to a given search criteria. The client input information is in JSON format, including start time, end time, address, and key terms.

Requirement 2

Users can search local recommended tourist attractions according to relevant locations.

Requirement 3

Users can intuitively understand the current epidemic information on the website, such as the number of infected people, etc

Requirement 4

The API should follow REST design principles

Requirement 5

A log file of API on the back end is also needed, which helps developers to track resource usage and monitor API performance.

Requirement 6

API should be viewable by Swagger, easy for developers to debug and update maintenance, and easy for users to use

User Cases

User Case 1: Disease Search

| | |
|------------------------------|--|
| User Case 1 | Search for disease information |
| Story | As a user, I want to know information about diseases so that I can gather information on outbreaks for research. |
| Relevant Requirements | Requirement 1, 4, 5, 6 |
| Actors | Average user |
| Basic Flow | When user type in their search information according to web instructions Then the information of disease which user is interested in will be shown on the website |
| Alternative Flow | When user clicks on the recommended articles Then the information of latest disease will be shown directly on the website |

User Case 2: Tourist Attractions Search

| | |
|------------------------------|---|
| User Case 2 | Search for tourist attractions by location |
| Story | As a user, I want to search for tourist attractions in cities so that I can plan my trip. |
| Relevant Requirements | Requirement 2, 4, 5, 6 |
| Actors | Average user |
| Basic Flow | When user type in the city name or key terms of tourist attraction and click on the search button Then the information of tourist attractions which will be shown on the website as a table with rating and famous picture in google map |

User Case 3: Global Epidemic Information

| | |
|------------------------------|--|
| User Case 3 | Get real-time epidemic information |
| Story | As a user, I want to intuitively understand the current global epidemic information, so that I can understand the current disease severity through simple operations, and can reasonably plan my travel plan |
| Relevant Requirements | Requirement 3 |
| Actors | Average user |
| Basic Flow | When user clicks on the COVID-19, Then statistic of COVID-19 will be shown as line chart and heat maps of the world |
| Alternative Flow | When user roll down at the index page Then the heat map of statistic of COVID-19 in the world will be shown directly on the website |

Web Application Design

Previous report API Design

Implementation Decisions:

1. Python requests is used to send get requests to CDC websites to get the HTML information and extract information which related to user requests from HTML by using BeautifulSoup4 package.
2. While parsing HTML pages from CDC website, GeoText package is used to find cities' name from texts.
3. We chose to use multivalued query parameter style when passing array of parameters from GET method. For example if user needs to pass 2 key_term items, user needs to put "key_term=test&key_term=tmp" in the url. The reason is that we think this way of passing parameters is more clear, and in the additional dataset we used, some cities in the US contain commas in their names if we use commas to separate the multiple parameters it might be confusing.
4. For the log file, our API returns team name, accessed time and data source in the response. And we use AWS Cloudwatch to log lots of other information including full request, full response, number of invocations in graph and integration latency etc.

D3 presentation feedbacks

From the feedback of our D3 presentation, we did not have a clear business value,

The layout of our website was focusing on Covid-19 which misled our audience and doesn't really match with the business value we talked about during the presentation.

We find out the main reason that we did not have a clear business value is that when we assign each group member works. We were thinking more about the functionality, instead of the business value. We were trying to build as many features as we can, and did not carefully thinking about business value and the layout of our website.


Website has the basic function ability but not fully implemented yet.

The Article Searching uses API to get data from WHO, and display the most recent outbreak articles on the website. Searching feature not implemented yet.

The Tourist Attraction uses API to get tourist attraction details of a city. The basic function ability is finished but with poor layout and could add more features to improve user experience.

The Covid Statistic Uses API to get covid-19 information of each country, and display in a graph and on a world map.

In this example you can see that the functionality is working, but the website layout is very poor. The information is all huddled together, which makes it harder for users to understand. The user experience is extremely bad. But because during D3 we were focusing more on the functionality, The layout and the business value got much improved at D4.

 DiseaseTFA Covid ▾ Article Searching Tourist Attractions

sydney

Confirm city Name

Get Tourist Attraction!

Tourist Attractions from sydney:

- Attraction name: Sydney Tower Attraction Rate: 3 Attraction point: 151.208954 -33.870499
- Attraction name: Sydney tower entrance Attraction Rate: 3 Attraction point: 151.208466 -33.870705
- Attraction name: The Star Attraction Rate: 3 Attraction point: 151.195053 -33.867924
- Attraction name: Bradleys Head Lighthouse Attraction Rate: 7 Attraction point: 151.24675 -33.853592
- Attraction name: Obelisk Beach Attraction Rate: 3 Attraction point: 151.26088 -33.829781
- Attraction name: Bronte Park Attraction Rate: 3 Attraction point: 151.267029 -33.903168
- Attraction name: Bronte Beach Attraction Rate: 3 Attraction point: 151.268295 -33.903622
- Attraction name: Bronte Baths Attraction Rate: 3 Attraction point: 151.269302 -33.905235
- Attraction name: watsons bay Attraction Rate: 3 Attraction point: 151.281601 -33.844181
- Attraction name: Macquarie Lighthouse Attraction Rate: 7 Attraction point: 151.285172 -33.853943

D4 presentation feedbacks

We learned our lesson from D3 feedback. We quickly re-designed our website layout, which will not mislead our audience and also enhance our business value. Every feature now tries to focus on the business value. We group together every part as a whole instead of spread out, this will help improve user experience.

The Article Searching now has a search feature that can search between time location and key terms. Each article now has a report summary which gives users a basic understanding of what the article is about. The layout was re-designed and gives better user experience.

The Tourist Attraction now has a feature that can search for different categories, and users now have the freedom to search whatever they need. Kinds give the user basic understanding of attraction, Rate gives users the best options that users can choose and if the user is really interested in the attraction there will be a link to Wikipedia and google map helps users to learn more about the attraction. The background color is using our main theme color purple, which gives visual comfort to the user and The functionality is more focusing on our business value

Our Covid-19 Statistics were only showing Covid-19 data of a country. Now Our Covid-19 Statistics is able to show the Covid-19 data of each state.

Years after Covid-19 finishes. Our website may not show about Covid-19 anymore, but will be more related to local outbreaks. In the future we can add more features to Tourist Attractions. For example when searching a city, we also search for if there is an outbreak going on, and display with the attractions. We will keep group things together as a whole instead of spread out, focusing on business value and improving user experience.

In this Example you can see that the layout compared to D3 is not messy anymore, but clear and visual comfort to the user. The functionality now is focusing on our business value and huge improvement on user experience.

sydney,casino

You can search for City or search by your categories, with input format "city,type,type,type"

some of the popular options: museums, art_galleries, zoos, casino, beaches, sport, bars

[here is all categories you can choose](#)

sydney,casino

Start Searching

| Tourist Attractions from sydney | Attraction Rate | Kinds | Wikipedia | Google Map |
|---------------------------------|-----------------|---------------|---------------------------|----------------------------|
| The Star | 3 | casino, adult | Wikipedia | Google Map |
| Poker Palace | 1 | casino, adult | Wikipedia | Google Map |

APIs

Team Viral API - <https://teamviral-api.herokuapp.com/docs/>

Disease API - <https://disease.sh/docs/>

Oxford Covid-19 Government Response Tracker - <https://covidtracker.bsg.ox.ac.uk/>

Open Trip Map API - <https://opentripmap.io/docs>

Open Trip Map API is used for searching tourist attraction for users, provides users a list of options for them to visit.

sydney

You can search for City or search by your categories, with input format "city,type,type,type"

some of the popular options: museums, art_galleries, zoos, casino, beaches, sport, bars

[here is all categories you can choose](#)

sydney

Start Searching

| Tourist Attractions from sydney | Attraction Rate | Kinds | Wikipedia | Google Map |
|---------------------------------|-----------------|-----------------------------|---------------------------|----------------------------|
| Barrenjoey Lighthouse | 7 | lighthouses, towers | Wikipedia | Google Map |
| Cape Bailey Lighthouse | 7 | lighthouses, towers | Wikipedia | Google Map |
| Reservoir | 7 | towers, architecture | Wikipedia | Google Map |
| Hornby Lighthouse | 7 | lighthouses, towers | Wikipedia | Google Map |
| Macquarie Lighthouse | 7 | lighthouses, towers | Wikipedia | Google Map |
| Bradleys Head Lighthouse | 7 | lighthouses, towers | Wikipedia | Google Map |
| Jones Beach | 3 | beaches, interesting_places | Wikipedia | Google Map |
| Garie Beach | 3 | interesting_places, natural | Wikipedia | Google Map |
| Wattamolla Beach | 3 | interesting_places, natural | Wikipedia | Google Map |
| Marley Beach | 3 | interesting_places, natural | Wikipedia | Google Map |
| Shelly Beach | 3 | interesting_places, natural | Wikipedia | Google Map |
| Elouera Beach | 3 | interesting_places, natural | Wikipedia | Google Map |
| Wanda Beach | 3 | interesting_places, natural | Wikipedia | Google Map |
| Greenhills Beach | 3 | interesting_places, natural | Wikipedia | Google Map |
| Boat Harbour | 3 | beaches, interesting_places | Wikipedia | Google Map |

Our website used the Team Viral API to provide search article function to users, users can use it to search articles about a specific disease that they are interested in. With the Viral API search function, users know can know if there is any outbreaks in their destination or not.

Please enter the following fields to find articles

Start Date

e.g. 2009-09-23

End Date

e.g. 2021-09-24

Location

e.g. India

Key Terms

e.g. nipah virus

Search



Recommended Articles of Outbreak

• Wild poliovirus type 1 (WPV1) - Malawi

Published on 3 March 2022

On 17 February 2022, WHO received an update regarding the detection of wild poliovirus type 1 (WPV1) [Read More](#)

Reports Summary

Date: 2022-02-17

Diseases

- poliomyelitis

Locations

- Malawi

Syndromes

• Circulating vaccine-derived poliovirus type 2 (cVDPV2) – Yemen

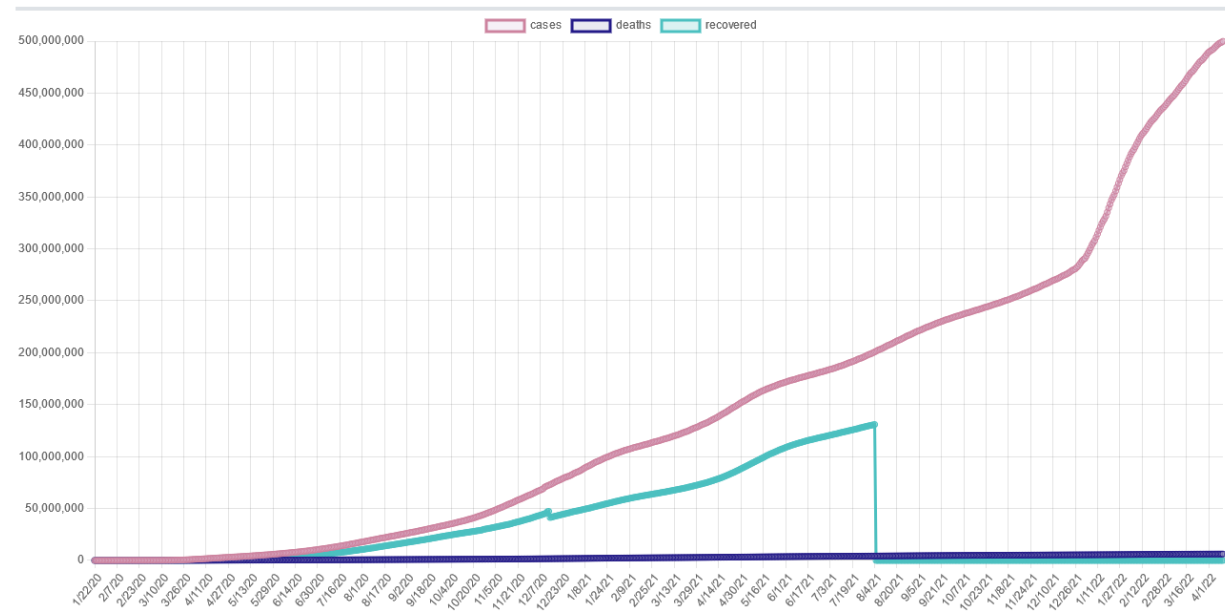
Published on 9 December 2021

On 22 November 2021, the International Health Regulations national focal point (IHR NFP) for Yemen n [Read More](#)

Reports Summary

Disease API is used to fetch world wide Covid-19 statistic, where the website uses these statistic to display what is the general situation in every country and states. Since from 2020, this Covid-19 pandemic has been going on for two years, it is a major concern for people when they are planning their trip, with the disease API, the website is able to give users a first idea about how is the Covid-19 situation in their destination, so users can prepare for their trip in advance.

Covid-19 Statistic

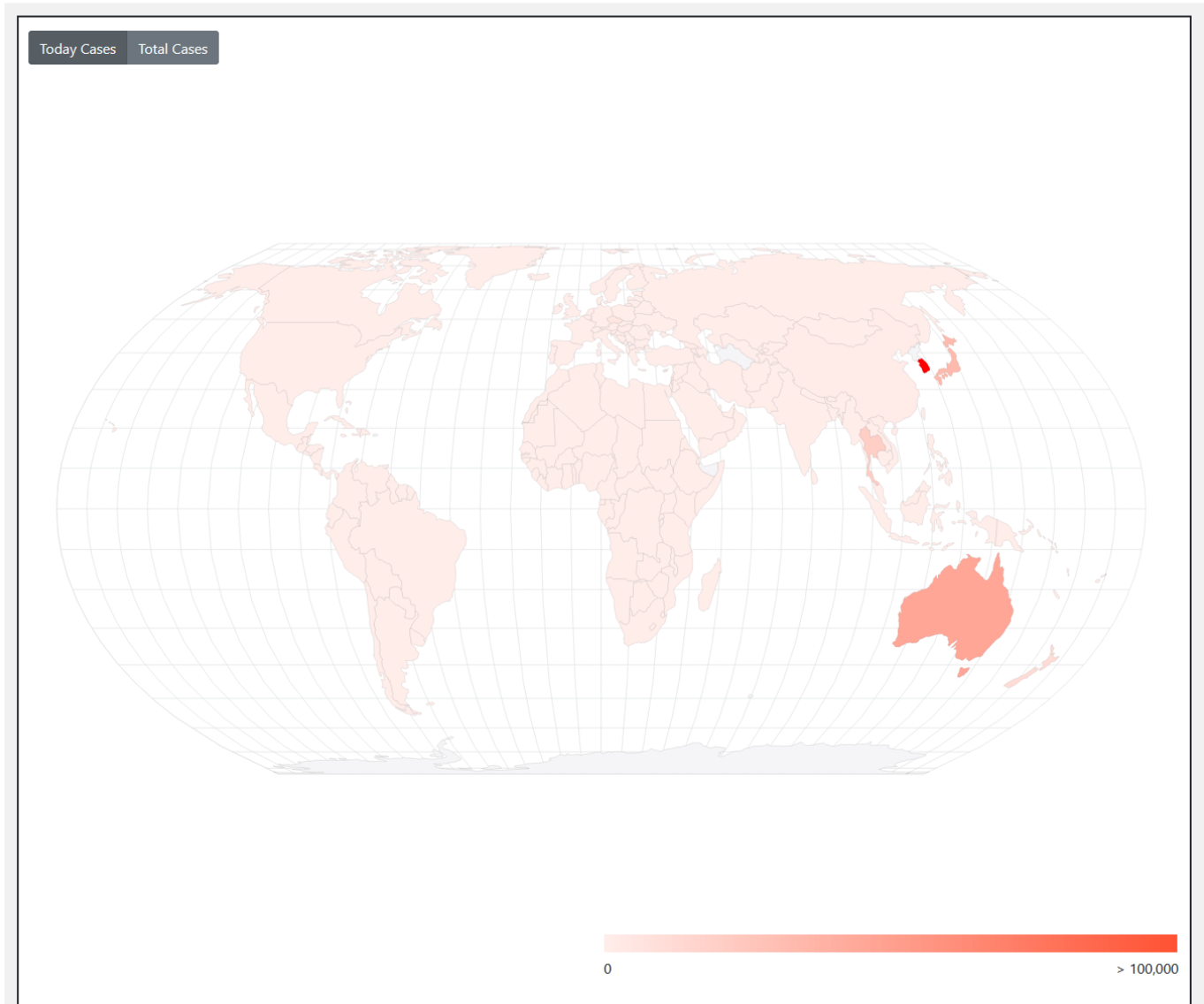


Search Date

You can check a specific date

| date | Confirmed Cases | Deaths | Recovered |
|--------|-----------------|---------|-----------|
| 4/5/22 | 493669057 | 6159243 | 0 |

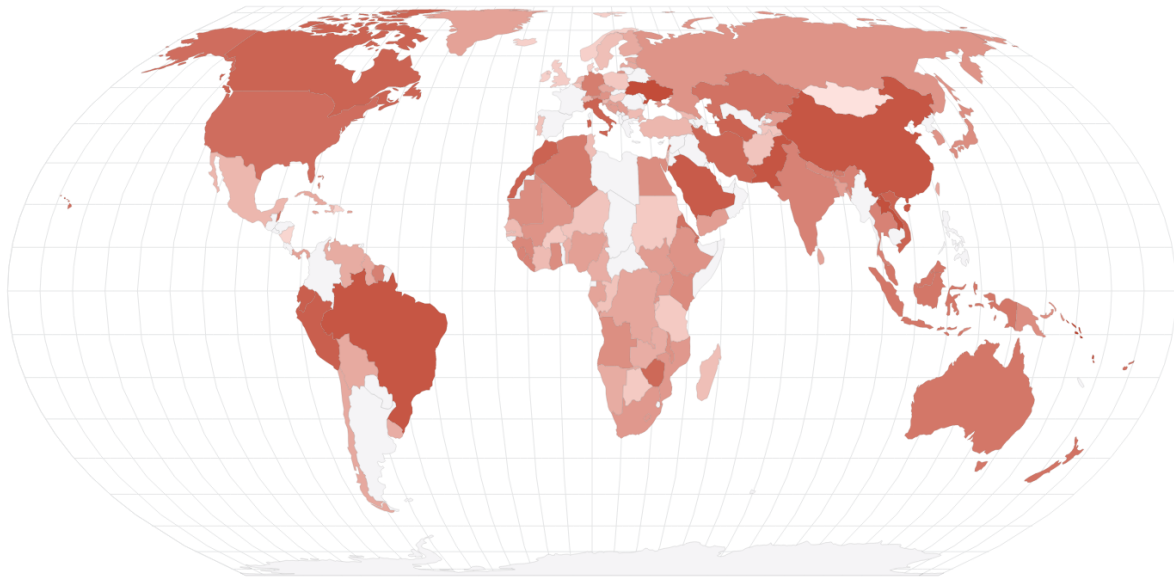
More about Covid-19



The Government Response Tracker is used to fetch data about how world wide government response to the Covid-19, our website will show users the stringency of each government, so users can have a first impression about how strict the policy for Covid-19 precaution going to be.

COVID-19 Stringency Index, Mar 30, 2022

The stringency index is a composite measure based on nine response indicators including school closures, workplace closures, and travel bans, rescaled to a value from 0 to 100 (100 = strictest). If policies vary at the subnational level, the index shows the response level of the strictest subregion.



2020-01-21

2022-03-30

Algorithms

No algorithm is applied in the project development. The team developed the web application by providing intuitive illustration for each API data.

Key benefits/achievements

The article searching design allows user to check the articles of outbreaks or directly read through the report summary to get the info about outbreak happened now or before, so that they can evaluate this place is suitable to travel or not, and they can also have the precaution before the trip.

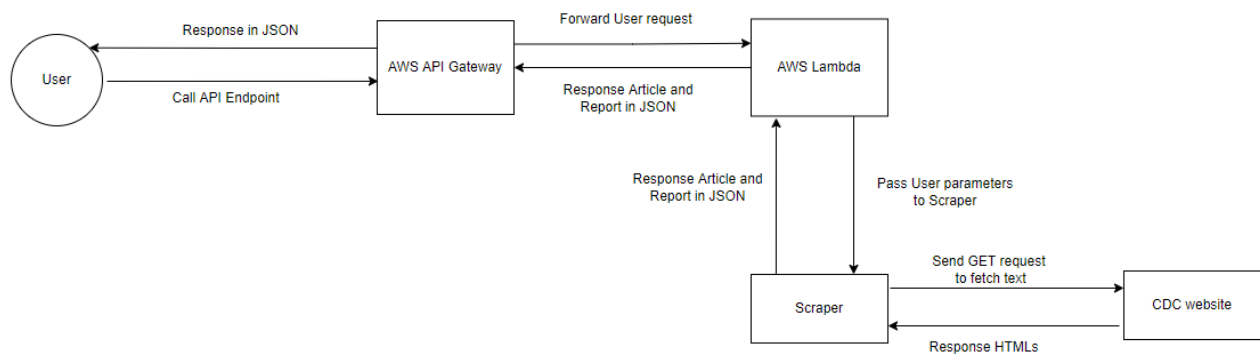
The tourist attractions design allows user to find out attractions with rates, so they can find out this place is worth to go or not, and then they can choose some preferred attractions before they start the travel. Also, if user don't have a idea about which attraction is worth to go, we will have a recommended attractions in the future so that user can find more place they may would like to go for a look. and we may have advertisements in the recommend attractions or sell the fight tickets to make money.

The Covid-19 pandemic has been affecting the entire world almost two and half years, most of countries decided to live with Covid-19. We design few features to provide users real-time Covid-19 related data to users.

In the covid-19 themed page, it has the similar idea with article searching page, this page shows a composite index of the repose level of the government, users can get ideas of what measure they might need to take to travel to this country, for example, entering China requires 7 days ahead PCR Covid-19 test and 14 days quarantine. Combine with the Covid-19 map, user can make a rough prediction about the Covid-19 in each region. so that they can have precaution before the trip.

When Covid-19 no longer concerns people, this showing-people-most-concerning-disease design is quite flexible, we can easily change to other most concerning disease. The application will keep users up with the real-time information.

Software architecture

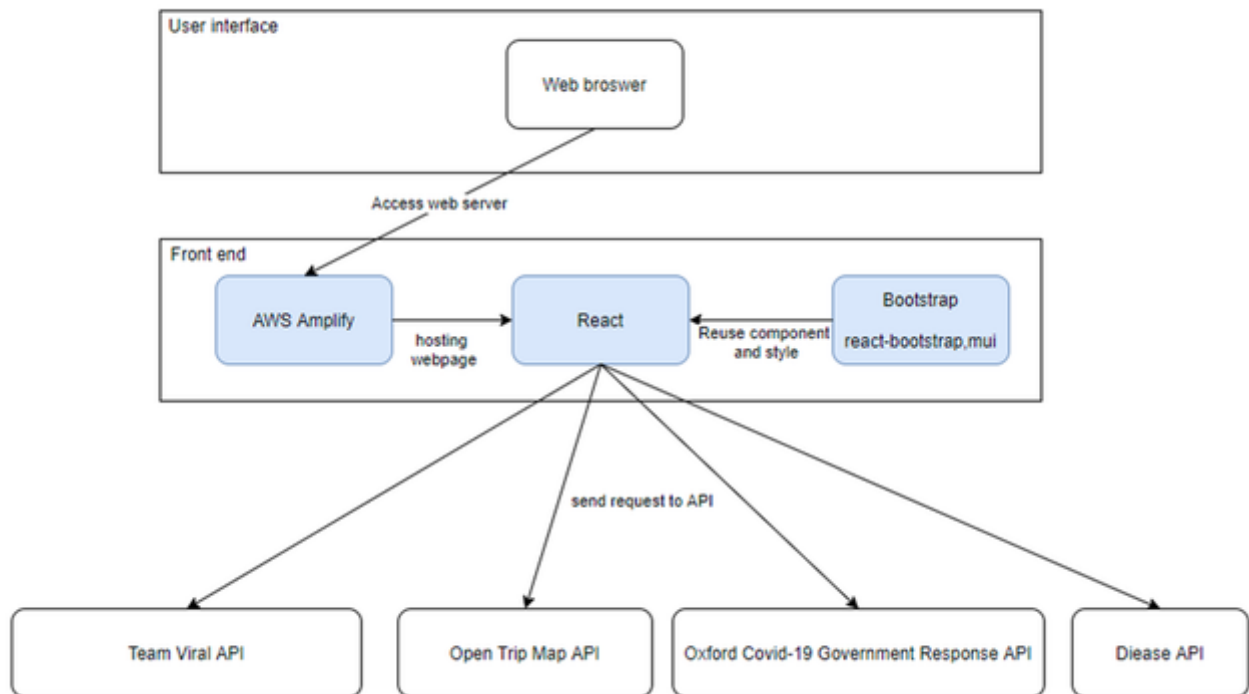


The main component of our API is AWS API Gateway, AWS lambda and CDC web scraper. The way AWS API Gateway works is by choosing the integration type for each endpoint, and that particular integration type is responsible for return the correct required data.

We choose to only use AWS lambda as our integration type. And in order to do this, we need extra component to handle the parameters which AWS API Gateway send to the lambda function in a documented way. For simply reason, we call this part "**lambda function**" at the rest of the report. This lambda function part is actually a replacement the Python flask framewrok which we originally plan to use in D1. After handle the parameter, we forward the request to CDC web scraper and process the data returned also in the lambda function. Finally, once our code is ready, we packdge the lambda function with the specific function from scraper it need in a zip file and uploaded to AWS lambda. Of course, it is connected to a API endpoint in advance.

The most important reason we choose AWS API Gateway and AWS lambda is how it integrated together will save us a fair amount of work. And as the trend of increasing needs of AWS cloud service in the software industry, leaning those technolgies also benifit our future career. The rest of the reason is the same in the D1 design, cited in the following paragraph.

In this project, we pay more attention to Serverless Architecture's low cost, scalability and simplified backend programming. The security feature it provides is also important to us, so that we don't need to worry about security issue and put more time on develop our product. But since non of us learned about using serverless architecture, it definitely cost us more time to learn.



On the top, it's our user interface, since this project is not focus on multiple platform development nor mobile application, we only have our product available at web browser. After the user open our website, the AWS Amplify will handle the request and display our website. And note that we don't have backend and backend server. The way AWS Amplify host our website is we upload a packaged frontend file to it, then it automatically builds it and deploy it to a provided domain name.

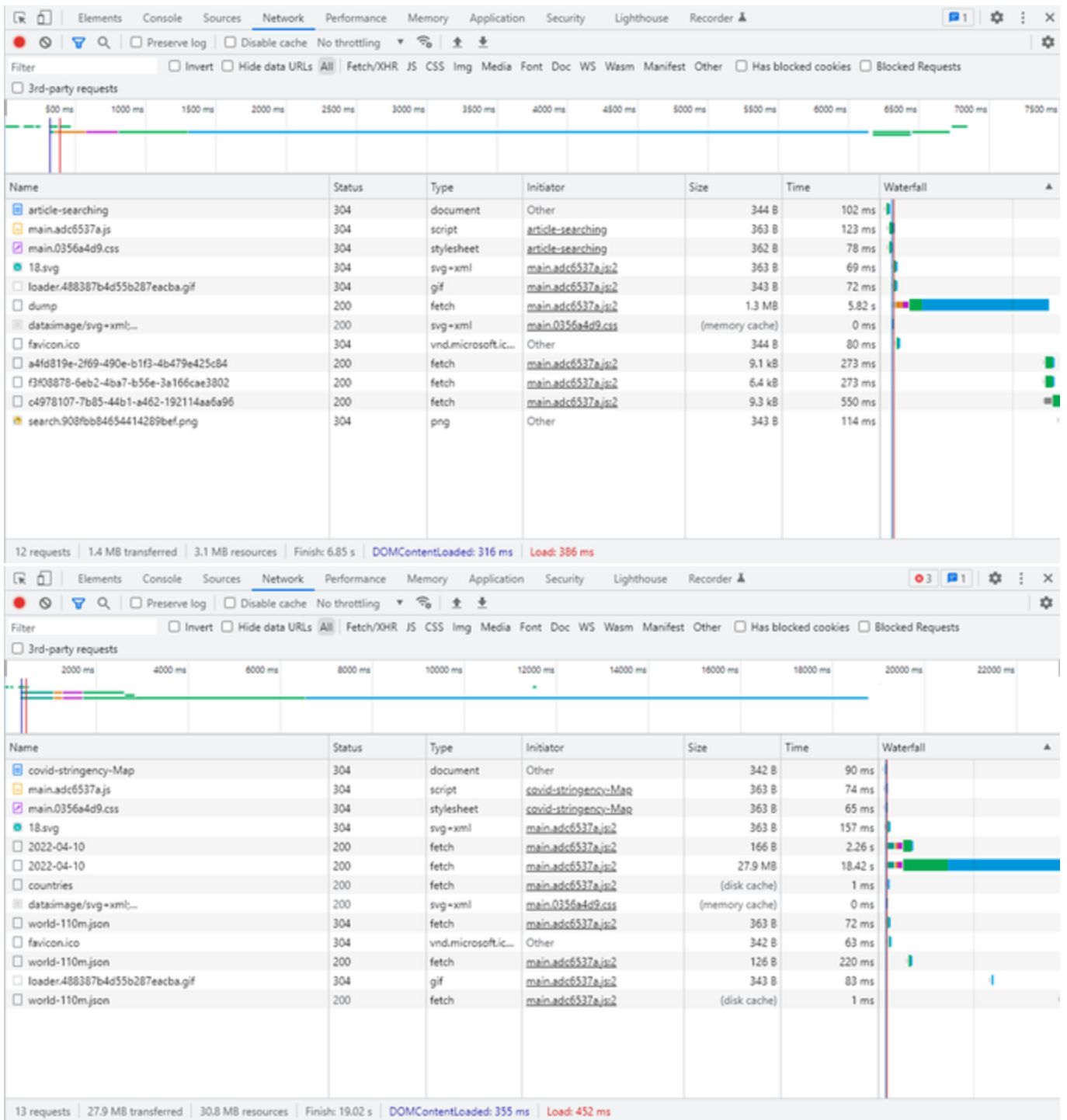
Then is our frontend structure, we use React as our frontend framework, it is in charge of send request to APIs, process the data returned and display them in the right place. We additionally use many Bootstrap libraries to simplify the implementation process like react-bootstrap and mui, they provide tons of powerful reusable and good-looking components. For example, in our stringency index page we have used react-simple-maps to construct the map and slider from mui library, then after fetch the stringency index data from Oxford Covid-19 Government Response Tracker we display each stringency index of its country at each date.

The detailed reason of our software architecture will be discussed in the following paragraph.

At the meeting we hold at the start of D3, we discussed the basic architecture of our website and shuhao raise the idea based on his own experience that we perhaps don't need a backend for many features. And after we split the frontend work feature based, we each did our own research on whether a backend is needed for this feature because based on the Occam's razor design principle we all learned in pervious courses, we want to keep everything simple so if we don't need a backend we won't implement one. In the next meeting, we finalize the feature we each choice and find out none of them need a backend.

The advantage of such choice is that it makes implementation and hosting of our website much easier. The disadvantages are harder to perform complex computation and analysis data, don't have access to database, store all data on the RAM. Backend language and tools like Python is great at process complex data and analysis data, but it is not something we definitely need, as we decided in the meeting all of our features do not involve complex algorithm and the data from the APIs is well structured which do not require much to process.

In term of the data store and RAM usage, in our website, we store all the data that user will need in the RAM for example the data needed to plot the covid related maps. Another solution that in comparison is use database and fetch most data straight from database, for example for covid stringency index map, every time user changes the date fetch the data that day from database instead of loading all data of all dates in RAM. The speed of the database solution might be slower but is sufficient. The most important reason is all the data is at most around 30MB see the images of most RAM used page from blow. And since most of the devices nowadays have more available RAM than that we can safely assume it won't affect our user. So we can say the performance of these two solutions is almost the same, but the database solution is harder to implement since backend and database required.



The reason we choose AWS Amplify is based on the research we did earlier, AWS Amplify is designed for website with single page web application (dynamically rewrite the current web page) frameworks such as React JS, website that do not have server-side scripting like PHP and website that have serverless backend or no backend. It is clearly suitable for our single page no backend architecture. This makes it very easy to use in our case and it is also the reason why no backend makes hosting of our website easier. Given the fact that we used AWS API gateway in previous deliverable, so use another AWS service is simpler.

AWS Amplify is also free to use with 1000 build minutes (the time takes for AWS to build our website) per month, 5 GB stored per month and 15 GB served per month which is way sufficient than we need.

Testing methodology

We have used different Testing methodologies throughout our project in different deliverables.

Below paragraphs is cite from our previous D2 testing documentation.

For our API, we have used two methods we used to test our API, AWS API Gateway ->Method Execution Method Test interface and another method is python with Pytest and requests library. To make it clearly, these two term “aws method test” and “pytest” is used below to represent this two test method.

These two methods actually represent our development stage, we use aws method test more often in the relatively early stage of development. This is because aws method test gives more detailed error message and full request log when something went wrong, make it easier to debug the program.

We use pytest when development is close to finish, when there is less possibility to occur bug. This is because pytest provides great automated testing function which is much more efficient than other methods.

So the usual testing process in later development stage is to use about 20 pytest functions to check one endpoint. When a test function failed, trun to aws method test to try again then give request and response log file to the development team to fix the bug.

End of citation here.

For our website, we only used react fremwork to test our frontend. Basically, everytime the frontend code get updated while the command “hpm start” or other similar command in running, the web page at localhost get updated at the same time. And the process of checking if the webpage is updated the way we want is our main way of testing. We did not use other automtic testing tools given the fact that no backend structure simply our implementation process.

Develop Methology

In the project development, we chose Rapid Application Development. It is for Agile based software development where the concentration is on a rapid prototype or rapid iteration release.

Advantages:

The development of the process is prototype-based and hence quite effortless.

This method allows clients or users to provide quick feedback, and depending on the suggestions; the changes can be easily implemented.

During D3 development, we set our project requirement and UI prototype first, then construct our application according to the requirement. After the D3 presentation, we had feedback from tutors. Then we refine our design based on feedback, quickly finalized our application.

Team Organisation

Our team is called 'groupName', and we have team members:

| Name | zID | e-mail |
|--------------|----------|-----------------------------------|
| Jiahao Zhang | z5266153 | jiahao.zhang2@student.unsw.edu.au |
| Kexin Tian | z5212870 | k.tian@student.unsw.edu.au |
| Shuhao Hu | z5241628 | shuhao.hu@student.unsw.edu.au |
| Haochen Shi | z5255218 | haochen.shi@student.unsw.edu.au |
| Dongfei Fan | z5211565 | dongfei.fan@student.unsw.edu.au |

Responsibilities/organization of the team

In the whole term, the responsibilities for each member are assigned during the group meetings. After the discussion about specs, we could have preview of that deliverable, then members could select what they want to do or be assigned appropriate work.

Here is the responsibilities of team members which would be different from the early design in D1 deliverable:

| Practical role | Name |
|------------------|--|
| Frontend | @ Shuhao Hu @ jiahao zhang @ Haochen Shi @ Dongfei Fan |
| Server Hosting | @ jiahao zhang |
| Web Scraper | @ Shuhao Hu @ Haochen Shi |
| API Design | @ Kexin Tian @ Shuhao Hu |
| API Module Build | @ jiahao zhang |

| | |
|--------------------------------|------------------------------|
| API documentation | @ Kexin Tian |
| API Implementation | @ Shuhao Hu @ Haochen Shi |
| API Testing | @ Dongfei Fan @ jiahao zhang |
| Preliminary Web Designs | @ Kexin Tian |
| Web Application Development | @ Kexin Tian |
| Group Management | @ Dongfei Fan |
| Devops work | <u>the whole team</u> |
| Report / documentation writing | <u>the whole team</u> |

| Agile team roles | Name |
|---------------------------|---|
| Scrum Master(Team leader) | @ jiahao zhang |
| Project Manager | @ Dongfei Fan |
| Product Owner | @ Shuhao Hu |
| Developer | @ jiahao zhang @ Shuhao Hu @ Haochen Shi @ Kexin Tian @ Dongfei Fan |
| Developmental Tester | @ jiahao zhang @ Shuhao Hu @ Haochen Shi @ Kexin Tian @ Dongfei Fan |

Here is the rough Gantt Chart designed in D1 deliverable:

<https://unswseng.atlassian.net/jira/software/projects/SE3Y22G21/boards/53/roadmap>

Here are the work arrangements for each deliverable:

| D1 Deliverable | |
|--|------------------------------|
| API Module Design & Web Service Design | @ Haochen Shi @ jiahao zhang |
| API Parameters Design | @ Shuhao Hu |
| Technical Tools Justify | @ Kexin Tian |
| Management Information | @ Dongfei Fan |

| D2 Deliverable | |
|--------------------------------|---------------------------|
| API Documentation & Endpoints | @ Kexin Tian |
| Swagger Documentation | @ jiahao zhang |
| Web Scraper | @ Shuhao Hu @ Haochen Shi |
| API Implementation (/articles) | @ Haochen Shi |
| API Implementation (/index) | @ Shuhao Hu |
| API Implementation (/covid) | @ jiahao zhang |
| API Testing | @ Dongfei Fan |

| D3 Deliverable | |
|---|------------------------------|
| Frontend Design & Map Chart & Chart Graph | @ Shuhao Hu |
| Stringency Map page | @ jiahao zhang |
| Article Search page | @ Dongfei Fan |
| Tourist Attraction page | @ Haochen Shi |
| Web Prototype | @ Kexin Tian |
| Slides | @ Kexin Tian |
| demo | @ Dongfei Fan @ jiahao zhang |

| D4 Deliverable | |
|---|---|
| demo | @ Kexin Tian @ jiahao zhang @ Dongfei Fan |
| Use Cases & Requirements (final report) | @ Kexin Tian |
| Update from last report (final report) | @ Haochen Shi |
| Architecture (final report) | @ jiahao zhang |
| Api Design (final report) | @ Shuhao Hu |
| Project Management (final report) | @ Dongfei Fan |

Team Communication

In this term, besides the mentor session, the team arranged at least another two meetings to discuss the project content, questions remained in mentor session or lecture, and the work arrangements. If needed, for example, this part of work needs team members to work together, or having some unfinished work left before deadline, the team will also arrange meetings as many as needed. The minutes for meetings will be recorded by @ Dongfei Fan .

In general, the weekly two meetings will be arranged on Tuesday afternoon(after the mentor session) and a weekday that everyone available. On Tuesday, the team discussed the problems from the mentor session and planned for next step of work. On the other days, after several days of working on project, the team members can report what they have in last few days so that the team can sync progress, then have time to fix mistakes and prepare questions before mentor session in next week. Minutes can be found in Confluence[Minutes](#) or Github https://github.com/JackWhoooo/SENG3011_GroupName/tree/main/Reports/Management%20Information/Minutes

During meeting, there is no leader, so basically everyone could discuss what problem they have and then others could help with problems. For example, *Dongfei* throws a question and do not know how to solve that, but *Jiahao* knows, so *Jiahao* could help *Dongfei* to solve that, and if *Shuhao* has another opinion, he could also help with that. In this way, the group meeting is active and efficient. But if the problem is not huge, then the team can just solve it via chat. Therefore, the format of communication is mostly free discussing, there is no particular structure, just ask questions and solve problems. In this way, the team can gather as many ideas as possible to help with project processing. Besides solving problem during the meeting, we would usually talk about our progress at the beginning of the meeting and make plans about next tasks and next meeting usually at the end of the meeting.

For group chat, the team would like to use **WeChat** as tool to have chat. First, team members all have account of this software; Second, some of team members are not in Sydney right now, unlike other tools such as Messenger, WeChat does not need VPN to open, so that we can communication with each other more easily and quickly with no interference from network fluctuations; Third, this tool have enough functions that we can use for chat, such as upload files, have video/audio meeting.

For group meeting, the team would like to use the **Teams** as tool to have meeting as we both use it for mentor session and group meeting which is a common tool for us. Additionally, we can share screen when having a meeting and we can also share files and have simple chat.

Conclusion/Appraisal

In this term, our team successfully build a website based on APIs, here are the evaluation of our project.

Achievements

In this project, the website we built achieve all of our business value designed in early stage of the project, we made the web pages of outbreak articles searching, tourist attractions searching, covid-19 statistic, covid-19 world map, and covid-19 stringency Index to help user to evaluate this place is suitable to travel or not.

Additionally, in the D2 deliverable, we successfully made a scraper to get data from 'CDC outbreak' and then use the scraper to make API about finding contents of outbreaks. By using this API we built which called 'GroupNameAPI', you could get articles and reports of outbreaks related to the key terms you input. Besides, you can also get data of covid-19 by using our API which is because we use an extra dataset as source to build that.

In the website, we make it functional enough that you can use it to search the articles of outbreaks, or tourist attraction in different countries or regions, check the state of covid-19 in different countries or regions. In this way, user can plan their trip and have precaution before they start to travel.

Through the whole term, our team insist on having regular meeting to have discussion on the project, so that we can notice issues in the first place and could solve it as soon as possible by having chat or meeting. In this way, we could ensure the work is efficiently finished step by step without mistakes.

Issues Encountered

However, we still have problems during the work time. The largest problem would be about the API, it is hard to find enough suitable APIs to build the website. For example, about covid-19, there is always no data for some countries or regions; about tourist attraction, it is hard to find an API with enough details such as some description and images.

When we build the API based on the scraper of 'CDC outbreak' in D2, it is very hard to build a general-case scraper to fetch content on the website, hence we have to hard code for each case which make the work more complicated, and the finished API is limited, which cause we have to use another team's API to build our website in D3.

In D2 deliverable, as hardness of making API based on the 'CDC outbreak', it is late when we finish the API build which left very few time to test the API. Fortunately, we finish the test scripts at the end.

In D3 deliverable, we need to use build frontend to make the website, but some of team members were not very familiar with the framework we chose to use which is 'React JS'. Therefore, they need more time to learn 'React JS' before they start to work on frontend which make the process of project slower.

limitation

Certainly, we have limitations on our website. An obvious problem is response time of the website, you need to wait for a couple of seconds to load each page, and it will take longer or even failed to connect when you have a slow network.

In the article searching page, you have to search for exact word for 'disease', so that you can the results.

In the tourist attraction page, it only shows 10 results for each search, and user need to find key term they want to search in another page which make search more complex.

In the covid-19 map, it only shows the cases of today and the cases of total.

In the covid-19 statistic, there is no data of recovered case in the second half.

In the stringency Index, there is no data for each state in each country, but only countries' response level. Also, there is no data for some countries.

Improvement

As we have such issues and limitations, we do wish we have some skills before the workshop, and this way we can try to include them in the future courses:

- Ability to make scraper
- Ability to build API
- Ability to use 'Swagger UI'
- Ability to use frontend framework ('React JS')
- Ability to make style of website more beautiful ('html, scss')
- Ability to use AWS API gateway, AWS web hosting

After study of the whole term, we would have different plan if we redo this project. However, for the technical tools we used in this project would remain the same which we think they are good enough for us to complete this project.

The first thing would be the API, we would spend more time and make more effort on building our API in D2 which will save much time in the D3 when we try to find one suitable API for article searching. Similarly, we would spend more time on searching other suitable API in D3 to make sure we have enough useful data to build our website.

The second thing would be the website, when we have enough useful API, we could have more features and show more details on each pages to make our website more functional. For example, we could have a page for searching flight tickets which helps you plan your trip, we could have page turning function in article searching page and tourist attraction page which make website more neat and organized, we could have more data about covid-19 shown in map or chart which gives user more information to help them evaluate this place is suitable to travel or not.

The third thing would be time management, we would have more detailed plan before we start to work. For example, if we have work that have prerequisite work to do, we need to finish the prerequisite work first and as soon as possible so that we could have time to do the next-step work and leave time to check the whole stage work. Additionally, to achieve this, we would have more communication in the group chat, such as report progress at once when they finish their work or ask question at once when they have problem.

The last would be based on the feedback from each deliverables in this course:

- D1: Make sure answering questions in spec in more detailed, clarify answers in more detailed, why or why not choose this. Will add 'above and beyond' content in D1.
- D2: Be careful with parameter to put in which need to be easier and clearer. Can have more logical and reasonable error message.
- D3: Need to have more clear business value in the demo, need to be more logical when doing the demo (the main sellpoint first). Try to finish basic function at the first demo.
- D4: Still need to clarify each feature with suitable business value.

Summary

Although we have these shortcomings and regrets in this project, we still successfully made a working and functional website. Thanks to all hard work of all team members and thanks to the help from all mentors and lecturers. Hope we can do better in the future.