

Irish Covid Dashboard

Sean Gavin

BSc. Software Design 2021

Athlone Institute of Technology

Declaration

I hereby certify that the material, which is submitted in this thesis towards the award of BSc. Software Design, is entirely my own work and has not been submitted for any academic assessment other than part fulfilment of the above named award.

Future students may use the material contained in this thesis provided that the source is acknowledged in full.

Signed.....

Date.....

Abstract

The aim of the Irish Covid Dashboard is a web application which uses Covid-19 datasets from Ordnance Survey Ireland and displays information in statistical and graphical features. The application gathers the data from the Ordnance Survey Ireland and displays information in statistical and graphical features.

Irish Covid-19 dashboard is made for a number of different user groups such as the General public, data scientist's i.e. NPHET and HSE. These three-user group can access the Web application from their systems. Each user will have the same access to all the data, as a major belief or vision for this project is being open and truthful to the public.

Acknowledgements

To begin, I'd like to express my gratitude to all of the lecturers who have taught me at Athlone Institute of Technology over the last four years and who have been supportive and patient with me and my classmates during that period.

I'd like to express my appreciation to Joseph McNamara my supervisor, who oversaw our final year project and has been extremely supportive, providing us with guidance, advice, and the value of his knowledge.

I would also like to express my gratitude to my parents, Clare and Val, for their unwavering support in not just the last few years, but my entire life. I owe you an everlasting debt of gratitude.

Table of Contents

Declaration	ii
Abstract.....	iii
Acknowledgements	iv
Table of Contents	v
List of Tables	vii
List of Figures.....	vii
Chapter 1: Introduction	8
1.1 Context	8
1.2 Rationale	9
1.3 Research Aims and Objectives	9
Chapter 2: Background Research	10
2.1 Introduction	10
2.1 Technologies	11
2.2 Conclusion	13
Chapter 3: System Design	14
3.1 Introduction	14
3.2 Requirements.....	14
3.3 Architecture	16
3.4 Design	20
3.5 Implementation	25
3.6 Conclusion	32
Chapter 4: Testing and Evaluation	33
4.1 Introduction	33
4.2 Testing	33
4.3 Frontend Testing	33
4.4 UX Testing and Feedback.....	34

4.5 Black-Box Testing.....	37
4.6 Evaluation	49
4.7 Conclusion	50
Chapter 5: Conclusions.....	50
5.1 Introduction	50
5.2 Reflection.....	51
5.3 Recommendations.....	52
5.4 Conclusion	53
References	54
Glossary	55
List of Abbreviations	57
Appendix A: Research Covid Dashboard'.....	59
A.1 Appendix WHO Dashboard	59
A.2 Appendix HSE Ireland's COVID-19 Data Hub	60
A.3 Appendix Dashboards with minimal information.....	61

List of Tables

Table 1 Breakdown of requirements	15
Table 2 Architecture technologies	18
Table 3 Store functionality for graph	30
Table 4 UX Feedback Table	36

List of Figures

Figure 1 Architecture Diagram.....	16
Figure 2 Technology Architecture.....	17
Figure 3 Home Statistics Wireframe	21
Figure 4 Home Statistics build	22
Figure 5 Interactive map wireframe	23
Figure 6 Interactive map build.....	23
Figure 7 Covid Store Wireframe.....	24
Figure 8 Covid Store Build	25
Figure 9 Parsing Covid data	26
Figure 10 Fetching Covid data	26
Figure 11 Statistical object	27
Figure 12 JSX code for statistical backend	27
Figure 13 Reading in Geojson data	28
Figure 14 Ploty/Dash html	28
Figure 15 Choropleth mapbox and Callback.....	29
Figure 16 Iframe component.....	29
Figure 17 Graph Store Example.....	31
Figure 18 Configure Store with actions	31
Figure 19 UX Feedback Survey	35
Figure 20 WHO.....	59
Figure 21 COVID-19 Data Hub	60
Figure 22 Minimal information example	61

Chapter 1: Introduction

As part of my final year project, I created Irish Covid-19 Dashboard. It's a Web application which uses Covid-19 datasets from Ordnance Survey Ireland and displays information in statistical and graphical features. I will now discuss the purpose of the project and the reason behind it.

The following paper will break down my thesis project into several chapters. In Chapter two, I will discuss the background research that led to the thesis project and in Chapter three, I will discuss the project System Design and how the technologies were implemented. The Fourth Chapter will outline how I used testing methods to ensure that the project was free of bugs and issues. Finally, in the final section of the thesis, I will dissect the project's results as well as possible project recommendations.

1.1 Context

Today we are bombarded with constant COVID-19 information and it can be difficult to separate fact from fiction. Reputable news organizations is where we put our faith to tell us accurate information, however in modern times we have seen these organisations take a biased or alarmist approach to the selection of information to be projected to the public.

The Chief Clinical Officer of the HSE Colm Henry said about these so called Reputable news organisations during the pandemic "Since the beginning of this, there has been a great deal of stories, the great majority of which are not true. I can't emphasise how hurtful these have been to families and to patients" [1]

The datasets being provided from the HSE on the OSI site are clearly a reputable source for the Covid-19 Ireland Dataset, Dr Tony Holohan Chief Medical Officer said about HSE datasets "All the data people need is on the HSE website — that is the trusted source of information." [1]

1.2 Rationale

A tool which uses data collected by reputable sources and projects this information clearly without a narrative behind it can both inform the public and reduce anxiety around the pandemic.

I propose a web application take advantage of the published and maintained datasets by Ordnance Survey Ireland. This information can be implemented with statistical and graphical features that can produce an interactive COVID application to allow users to request specific detailed information.

I found other covid-19 dashboards do not include much information on Ireland and focus more on the superpower counties which was definitely a factor in the application being build.

I want this application to inform the public and be adaptable to any member of the public such as general population, data scientist's i.e. NPHET and HSE. These three-user group can access the Web application from their systems. Each user will have the same access to all the data, as a major belief or vision for this project is being open and truthful to the public.

I hope the application would somehow be useful to the HSE as their frontend application is lacking in the graphical and statistical features for their covid datasets.

1.3 Research Aims and Objectives

- To gain better understanding of developing graphical and statistical features.
- To gain better understanding of other Covid Dashboard applications and be able to develop one.
- To gain knowledge of introducing other scientific Covid data from OSI.
- To gain better understanding of the data to inform the public.

In the next section, I'll go through my background research for the project.

Chapter 2: Background Research

2.1 Introduction

To begin my study, I began by looking at other online resources that have Covid dashboard. The top dashboards on the web contained a lot of information about countries who had high volume of covid cases, the major superpower countries have a large number of graphical and statistical features for their data. A lot of smaller countries in Europe such as Ireland didn't included breakdown of data only the essential details on the number of deaths and confirmed cases.

I believe this data can be taken out of context in the application and cause scaremongering among the general population. I believe if these other dashboards online gave more of an detailed account of the statistics it would give context of the situation occurring in that country. Ireland would definitely be acceptable to this issue. During the early stages of the pandemic, a large number of covid cases were occurring in Dublin City and a lot of data being shared or published did not represent this issue.

Since detailed account of Ireland's covid cases are not the priority of these dashboards, this gave me the inspiration behind the thesis project and I felt like developing my own application will fill the gap the other applications are not fulfilling.

As I mentioned earlier, a lot of these dashboards only contain the bare minimum of information and I feel a lot these dashboards are only trying to inform consumers of social media by showing them enough information until their next scroll Figure 22 Minimal information example. These Dashboards are not open to other users Groups such as Data science, Medical Practitioners and the general public who are looking for more information. If the dashboards were more relevant to all these users groups, people would feel more transparent about the pandemic.

Another major downfall of these applications is a lack of data being introduced, as I think there should be a greater emphasis on more scientific data such as covid laboratory test, ICU data and Admission and discharges in hospitals data. Scientific covid data on counties in Ireland could give the general public information on areas of Ireland to avoid or where high incidences of cases are occurring.

I looked at the most popular dashboards online such as [HSE Ireland's COVID-19 Data Hub](#) and [WHO](#).

I first investigated The HSE COVID- 19 dashboard which is developed by GeoHive and this application satisfied a lot of my objectives. The fact that the application focused on Ireland, the fact that the application implements a wide variety of datasets and the majority of datasets contain Scientific covid data sets which definitely fulfil my objectives Figure 21 COVID-19 Data Hub.

The homepage of the application contains summary charts for covid data which prompted me to achieve similar format in my application. These applications should be more user friendly as you have to load a new webpage to seek information on different datasets. I feel the graphical features in the application are very one dimensional and should be more interactive.

The WHO application which I investigated had a very interactive frontend. The application contains a map feature which shows stats about Ireland which is of interest Figure 20 WHO **Error! Reference source not found.** The WHO application gave me the idea that it would be interesting if the Counties were implemented into the map and have an interactive statistical feature for each county. The user would be able to determine covid cases for each county then.

Both of these applications had features that I decided I would try to replicate a similar format in my application. It was time to start my research after looking into the different applications that were similar to my application idea.

2.1 Technologies

2.1.1 Interactive and Static Graphical feature

I researched different data visualization options and I realised the importance of having good and clear data. A good visualization should tell the picture in a few seconds and I think it is vital to use the covid data in this format.

Static graphics are quite important and are often looked down upon, Static graphics are important for telling a short story for highlighting a current issue which is happening or so on, this could be the number of cases or covid deaths today. Most people these

days live a busy life and a simple static graph could have a major impact on those who don't have time to use the interactive graphs.

Having previously worked with Static graphs using Matplotlib in the data visualization module during my first semester of college this year, I had a fair understanding of how to develop them.

I investigated quite a few interactive graphical packages online and I investigated the pros and cons of each package. The benefit of interactive graphs is the fact the user can dive into data and the user can drilldown to understand the data more effectively will definitely help my Covid dashboard. Interactive graphs can help with the scientific data as ability to compare different statistics can be introduced into the application. I mainly focused on JavaScript graphing libraires such as D3 charts, Chart.js and I also took a few tutorials on Mapbox Choropleth Maps in Python.

The next area to research was frontend frameworks and the general user interface design.

2.1.2 User Interface

UX is a very important part of any application and due to the seriousness of the current pandemic I wanted to be precise and factful with covid information and not to mislead anyone. A lot of the design element I implemented were introduced based from my research of other Covid applications. When I thought about the overall user interface I wanted the features to be contained all in one webpage.

To ensure that the UX interface is completely adaptable and understandable to people of all ages and technological backgrounds, I enlisted the aid of my family to test the frontend design. I had the testers fill out feedback on the frontend to help me understand what I needed to work on, which I will cover in the testing portion of the thesis.

When I first started throwing together a lot of the static graphs from the backend using the number of different graphical packages, I started putting together a simple frontend using jQuery and bootstrap. The flask framework allowed for formating the covid data into JSON object. This approach helped visualise the application and helped the system be ready for the development process.

I knew I wanted the frontend to be developed using JavaScript and at the start I stayed away from Frontend frameworks too as I just wanted a simple approach to understand the concept of the project in an easier way. Later in the project I did implement a framework react which was a major change in the research process as I had to carry out multiple tutorials to get up to speed with react and their life cycle. One of the major things I noticed in my research is that a lot of application didn't integrate all their covid data and to access that data you had to reload the page in order to get the covid statistics.

This framework helped the application to be interactive and definitely filled the gap of other applications that I looked at using static framework approaches.

2.1.3 Flask Microservice

I needed a way for my frontend and backend to interact, so I Implemented some middleware for my application. I understood how to incorporate Flask and that it would work with my application because I had previously worked with it. Prior to this, I used Flask for a data visualization assignment in my first semester. Working on this project gave me some ideas how to approach the thesis, I believed Python would be the best option.

2.2 Conclusion

This concludes the context research chapter and allows the creation process to begin. However, my research would imply that I have completed my learning, which is not the case. There is no fixed timeline for completion of the research. I found that up until the very end of the project, I was following tutorials. Software Engineering is a life of constantly updating your skills in order to keep up with new technologies.

The following chapter will go through the various aspects of developing the application from start to finish.

Chapter 3: System Design

3.1 Introduction

The Irish Covid Dashboard's system architecture is described in the sections below, I will explain the scope, the architecture of the system and the system requirements implemented into the project. This section of my study would also include architectural views that depict the different aspects of the overall structure. It will also define the overall system architecture, the various components and classes and specific methods that enforce core functionality.

I'll also go through the configuration of the database that the program uses, as well as the design of the system's graphical user interface.

3.2 Requirements

At a high level, the Irish Covid Dashboard specifications can be summarized as one key requirement from which all other requirements can be derived from:

The Covid Dashboard is required to take OSI datasets and display the information with statistical and graphical features to inform the user about the Covid-19 pandemic. The program is intended to work flawlessly and to provide the customer with a pleasant experience.

This main requirement can be broken down into sub requirements which focus on other features in the application should satisfy the main requirement in the table below:

Requirements	User Story	Breakdown of requirement
Requirement one	<i>The user should be able to perform graphical features on a variety of OSI scientific Covid datasets, the user should have ability to access a number of different attributes in a dataset in a graphical analysis feature.</i>	<ul style="list-style-type: none">• Hover over attributes on the graph• Select multiple Covid attributes.• The ability to click on multiple different graph stores.

Requirement Two	<i>The User should be able to perform graphical features on interactive Geojson Map, so that the user can view Covid cases in their county.</i>	<ul style="list-style-type: none"> • Ability to select different County Covid interactive map statistics. • The ability to time lapse through County statistics with an interactive slider.
Requirement Three	<i>The user should be able to receive custom email subscription newsletter, so the user can be updated on Covid Pandemic.</i>	<ul style="list-style-type: none"> • The user should have ability to sign up, select custom graph and statistical information. • The user should receive custom email. • The user should receive Covid email newsletter every day
Requirement Four	The user should have ability to view statistical breakdown of the pandemic on the home page	<ul style="list-style-type: none"> • User should be updated with the latest figures. • User should be presented with information clearly
Requirement Five	The user should be provided with a weekly breakdown of covid statistics.	<ul style="list-style-type: none"> • User should have access to most up to date figures. • The graphs should be clear and interactive

Table 1 Breakdown of requirements

In order to use the program, the user must have a laptop or personal computer that is connected to the internet and have a web browser installed to view the frontend.

3.3 Architecture

3.3.1 Overview

The decision I made to use certain technologies in my system is heavily based on my background research into various technologies. I will also provide a brief explanation for each of the technologies I've chosen to use in my application in this section of the thesis.

3.3.2 Architecture Diagram

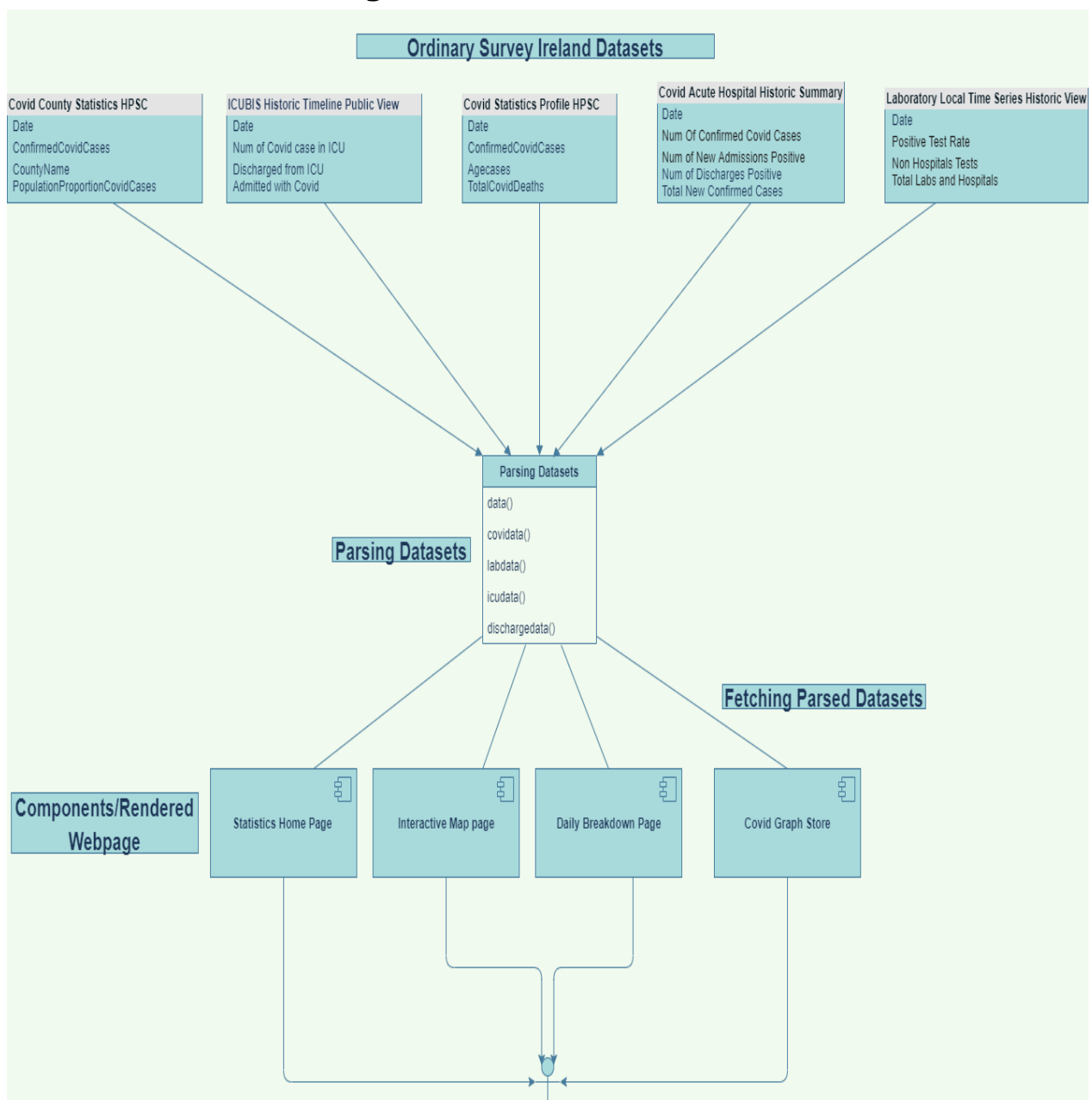


Figure 1 Architecture Diagram

3.3.3 Technology Architecture Diagram

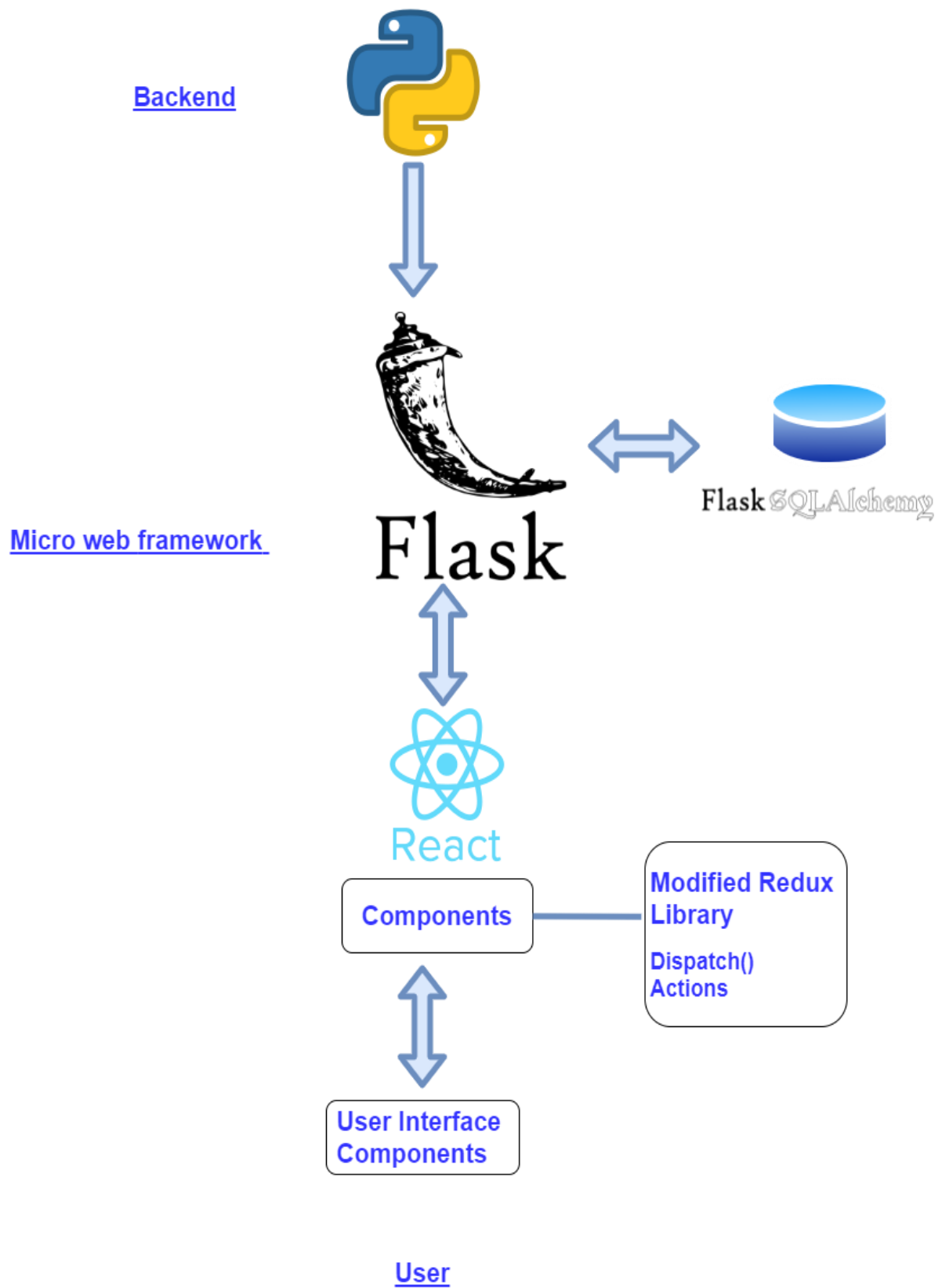


Figure 2 Technology Architecture

3.3.4 Architecture Summary

Service	Technology	Description
<u>Backend Service</u>	Python	Development of static graphs and manipulation of datasets
<u>Web Framework</u>	Flask	The middleware service between the frontend and backend
<u>Email Subscription service</u>	SQLAlchemy	Persistent storage for storing user email information.
<u>Frontend Architecture</u>	React	This service will be in charge of supporting users the physical web page.

Table 2 Architecture technologies

3.3.5 Python

Python is a popular high-level programming language that makes it ideal for developing applications which contain large datasets. Python is a brilliant object-oriented, immersive and understandable language that is easily understood by humans.

The python packages I used for *the application* were pandas, which allowed the storage of data in python and the ability to manipulate the covid datasets. Due to python wide range of libraries, I could incorporate matplotlib to generate static charts and allowed me to introduce more dynamic charting libraries such as plotly/dash which helped development of interactive maps. My major justification behind using python as a technology in my project is due to the fact that it is designed with features to facilitate data analysis and visualization.

3.3.6 Flask

Flask was the middleware platform that I used in my program and it began as an April Fool's joke in 2010 but has since evolved to become a useful solution to Django. Flask helps you to create full-featured web apps with only a few lines of code. As a result, it was the right fit to use as middleware in the project.

The implementation of flask allowed me to be adaptable to provide the datasets from the OSI to the frontend, Flask framework has a large community of modules which can be implemented into the framework. This helped me in the process of email subscription service as I implemented APScheduler which is a python module. Flask built in components such as Jinja allowed me to build dynamic emails which could be send to the subscribers.

3.3.7 SqlAlchemy

SQLAlchemy is a Python SQL toolkit and Object Relational Mapper that provides full SQL control and functionality to web developers. This can be installed as a module in flask which is the major reason behind the introduction of this technology.

The major reason behind the introduction of this technology is the ease of use of the technology and it can be implemented into flask. The database can be initialized all in one application without the need for setting up a sql server.

3.3.8 Frontend Architecture – React Framework

React is a JavaScript library that can be carried out to build dynamic and immersive web application user interfaces. React supports the use of a mix of JavaScript and JSX to create reusable components. Making reusable components eliminates code reuse around the application and speeds up the production time.

React can be used to change a single section of the current program, as I did, and if it works, you can move on to converting the whole app to React.

React enables the use of npm libraries, which made it possible to use JavaScript libraries like Material – UI. With the support of react documentation, I was able to add React components for the user interface, as well as some custom components.

3.3.9 Frontend– State Container Redux modified library

Redux is a JavaScript library for managing the application state. Redux allows the state of the application to live in more than one single component which is opposite to the react live cycle which only allows state within a parent React component. The modified version redux I used works as a custom hook that will act the same as the action and reducer work behind the scenes of redux source code.

My reason behind introducing a modified version of redux is the fact that I have had issues in react when implementing the interactive chart store which I will go into depth later one. I needed multiple components in the application to share state. This not only centralizes all data, but it also makes it very possible for a component to obtain the state it requires without other components being aware of its requirements which was vital in certain sections of the application.

3.4 Design

3.4.1 Introduction

In this section, I'll go through different design elements from different components within the architecture. I will discuss components of the functionality available to the user and the design consideration - the purpose of the design, so the user can fully understand the reasoning behind the styling and formatting of the application.

The application's overall design is user-friendly, as I feel that no technical product should be too difficult to use. All of the pages have navigational features and a header bar to indicate to the user which function they are viewing. Since there is no specific age demographic for the application, it can be understood by anyone.

To give the app a more genuine personal touch, I added custom CSS features that I created myself with the presentational JSX format for react application. The application also contains components from material -UI which helped the appearance of the dashboard.

I created a wireframe to base my design strategy in the application before implementing any components. I divided the application into four dynamic webpages

in the Irish Covid-19 dashboard and I will now clarify three components of the application and the explanations for design decisions taken.

3.4.2 Home Statistics page

The home statistics page displays yesterday's figures as well as overall results for a variety of statistics. The page's style is straightforward, with a dark background to emphasize the significance of the figures, which are shown in a brighter colour. A major design decision was including Font Awesome icons as I feel icons can convey a message in a matter of seconds and can transcend language barriers. The email subscription service is also available on the home statistics tab, with a button that the user can pick to bring up a modal/pop up form for user input. By separating the features, I was able to make the interface more user-friendly.

The following diagram is a wireframe which I designed before the process of making the Home Statistics page:

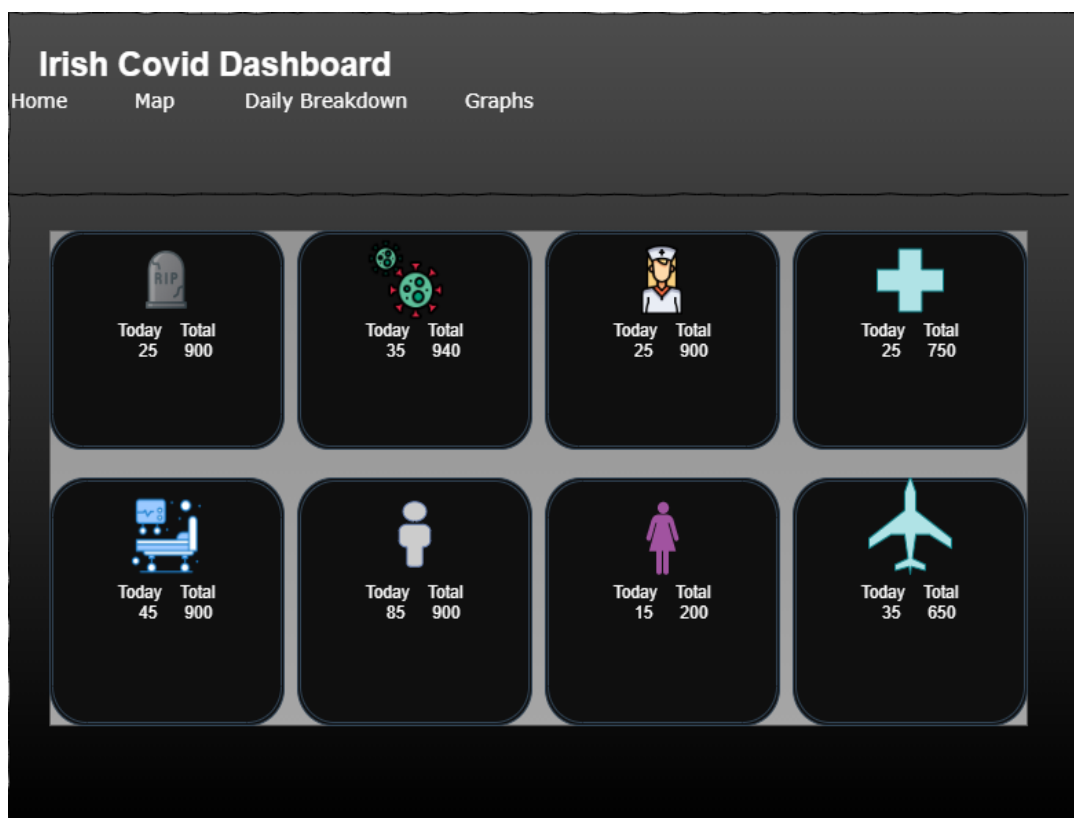


Figure 3 Home Statistics Wireframe

The design result of the Home Statistics page being build:

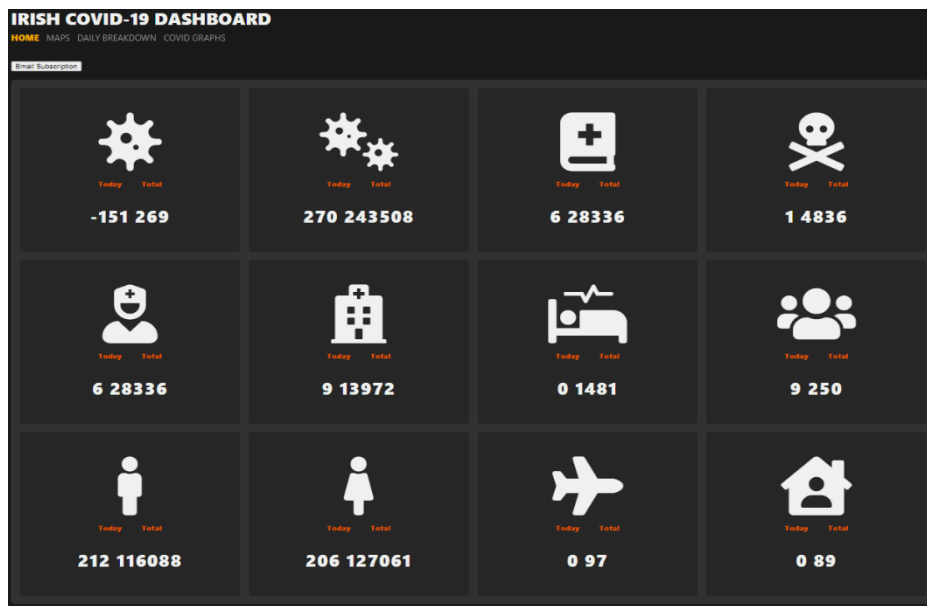


Figure 4 Home Statistics build

3.4.3 Irish Interactive Covid Maps

The Irish Interactive map page consist of interactive Geojson map using components build from Plotly/Dash. The components' design is very simple, as the idea is for the user to investigate and communicate with them, such as the collection box, slider, and interactive map. The program has a scale legend to help the user appreciate what the different colours in the counties represent. The map is fetched using a iframe component with provided a frame for the map to be placed in.

The map is primarily composed of three basic elements that are modified on a regular basis to provide the user with an interactive experience. The user will perform various covid statistics in the page using the selection box, such as total infections, proportional covid cases, and population. The interactive feature allows the user to rotate the map feature and perform a variety of functions such as zooming or taking a screenshot, and the user can hover over a County to get detailed covid information.

The option to time lapse for the covid details for each Country is the final feature I planned during the design process. Both of these features give the user the opportunity to look at covid across Ireland using a user-friendly interactive map.

The following diagram is a wireframe which I designed before the process of making the interactive map features.

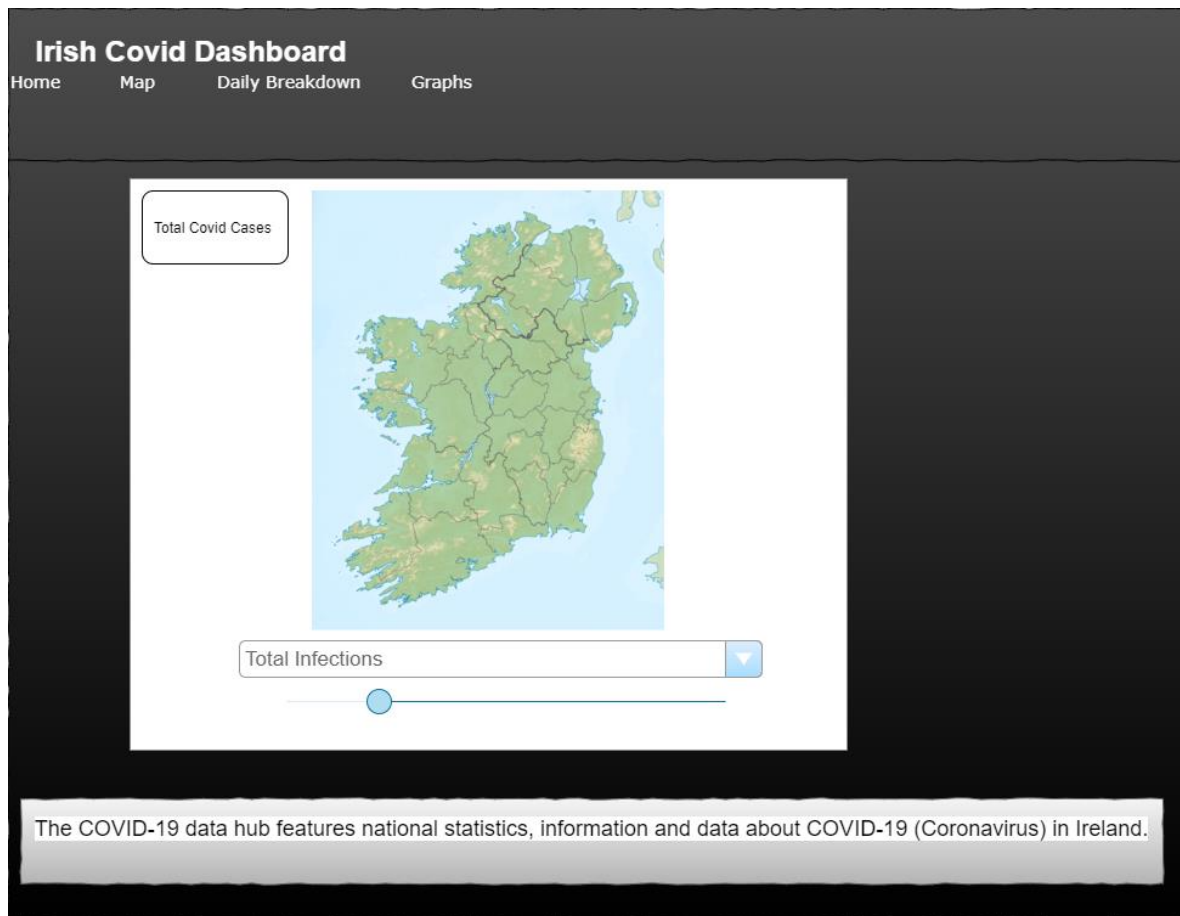


Figure 5 Interactive map wireframe

The design result of the Interactive map page being build:

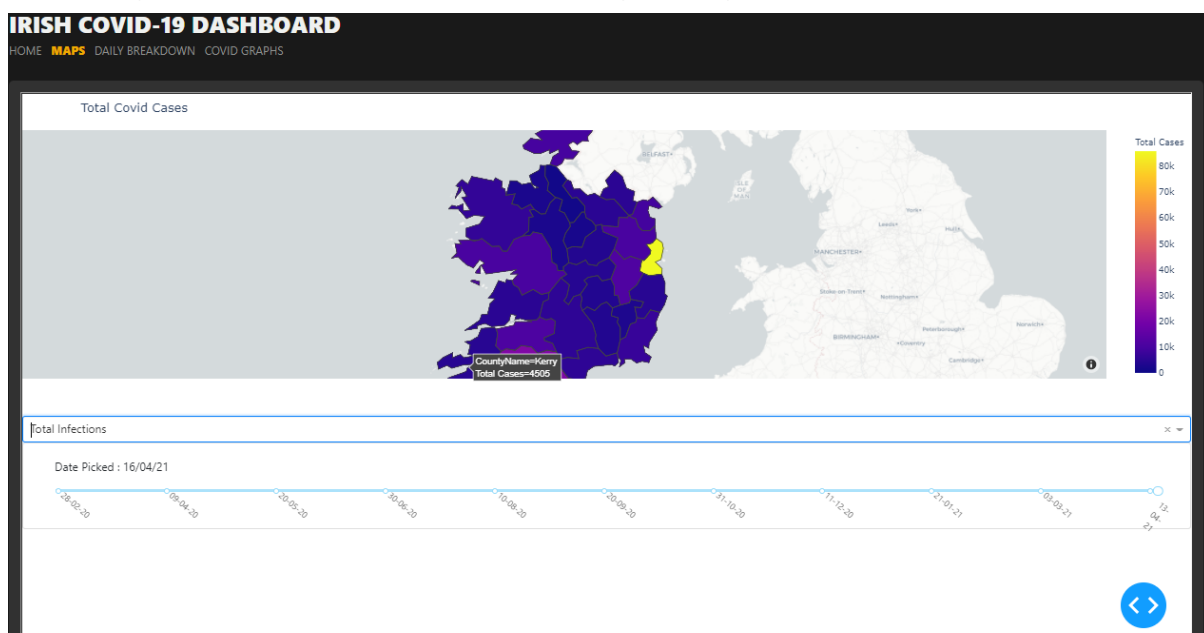


Figure 6 Interactive map build

3.4.4 Covid store components

The Covid store graphs page consists of interactive covid graphs which is build using D3 charts. The design of the page is relatively compact and has a dark appearance which is contrasted against the user input, when the user selects an attribute button a colour scheme occurs for that attribute in the graph and the button.

The graphs contain a number of different features which help ease to use when investigating different covid attributes. Through choosing different icons for the stores on the application, the user can choose from a variety of different datasets. The user will then choose from a variety of attributes in the dataset to fill the graph. The user can compare various values on the graph, which lets them understand covid cases. By creation of a textbox to remind the user, they can hover over attributes to get a reading in the graph.

The components in this program make it much easier to investigate datasets, and the textbox at the bottom of the application, which includes a summary of the dataset, continuously reminds the user what they're looking at.

The following diagram is a wireframe which I designed before the process of making the interactive covid store:



Figure 7 Covid Store Wireframe

The design result of the Interactive map page being build:

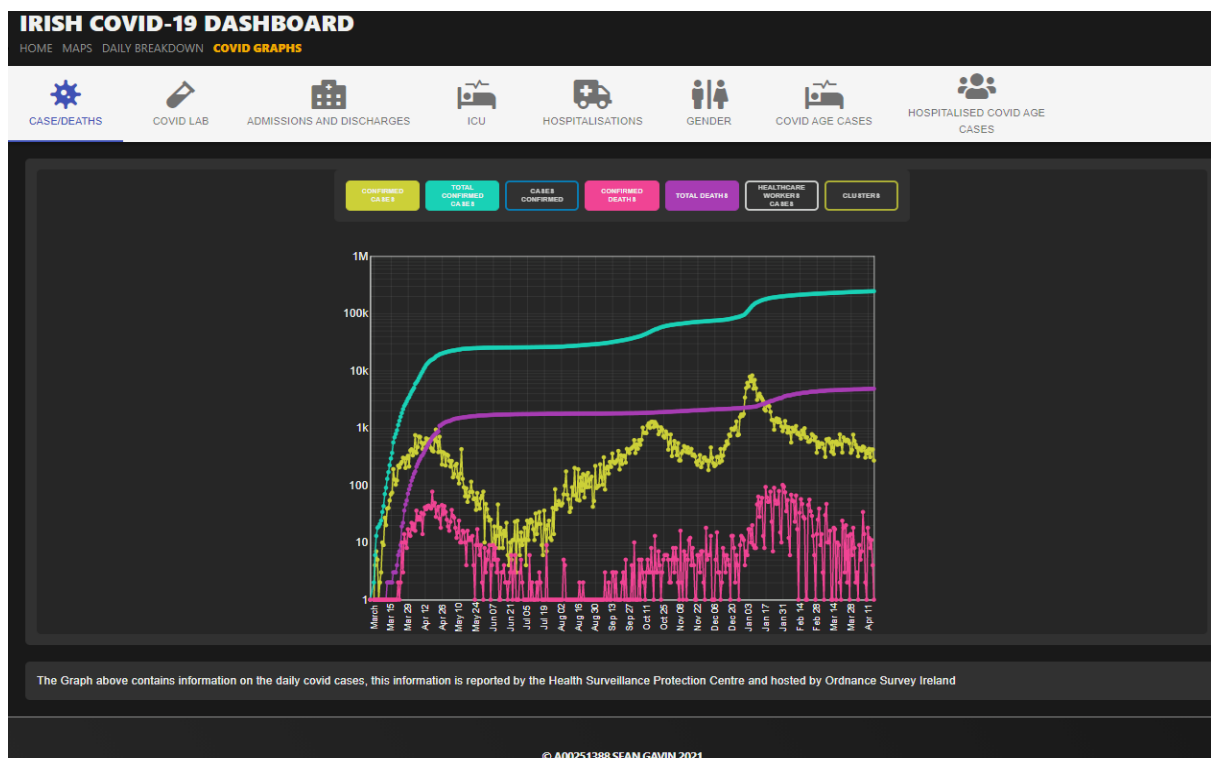


Figure 8 Covid Store Build

3.5 Implementation

This segment would go into how to put the project's functionality into action. Evaluating the programming of all user stories that were developed in the early stages of development is the easiest way to break down the deployment portion, ensuring that the most important features are running as intended.

I'll go through the most relevant features of the Home Statistics page, Interactive Covid Maps, and Covid store. Unfortunately, due to the application's sheer size, all the functionality cannot be reviewed.

3.5.1 Home Statistics page

The user should have ability to view statistical breakdown of the pandemic on the home page

The email subscription service's static matplotlib graphs are created by the program's backend python files. As seen in Figure 9 Parsing Covid data

source not found., I use the flask system to call these statistical numbers from the backend and parse the details for the frontend program.

```
res = {"pop": str(pop), "numberofcovid": str(count), "totalpopire": str(totalpop), "link": breakdown,
      "countychart": countychart, "countybreakdown": countybreakdown, "deaths": deaths, "totaldeaths": totaldeaths,
      "breakcases": breakcases, "concase": concase,
      "male": int(malecases), "female": int(femalecases), "Other": int(Unknown), "Aged1": int(Aged1),
      "Aged1to4": int(Aged1to4), "aged5to14Cases": int(aged5to14Cases), "aged25to34Cases": int(aged25to34Cases),
      "aged35to44Cases": int(aged35to44Cases),
      "aged45to54Cases": int(aged45to54Cases), "aged55to64Cases": int(aged55to64Cases),
      "aged65upCases": int(aged65upCases), "HealthcareCovidCases": int(HealthcareCovidCases),
      "totaldeaths2": int(totaldeaths2), "HospitalisedCovidCases": int(HospitalisedCovidCases),
      "HealthcareCovidCasestoday": int(HealthcareCovidCasestoday),
      "HospitalisedCovidCasestoday": int(HospitalisedCovidCasestoday), "Icucases": int(Icucases),
      "IcuCovidCasestoday": int(IcuCovidCasestoday), "todaymalecases": int(todaymalecases),
      "todayfemalecases": int(todayfemalecases),
      "todaytotaldeaths": int(todaydeaths), "TravelAbroad": int(TravelAbroad),
      "TravelAbroadtoday": int(TravelAbroadtoday), "CommunityTransmission": int(CommunityTransmission),
      "CommunityTrantoday": int(CommunityTrantoday), "ClustersNotified": int(ClustersNotified),
      "ClustersNottoday": int(ClustersNottoday), "totalcases": int(totalcases), "todaycases": int(todaycases),
      "concases": int(concases), "todayconcases": int(todayconcases)}
return jsonify(res)
```

Figure 9 Parsing Covid data

I then fetch data in the react application after the data has been parsed in the flask application, as seen in Figure 10 Fetching Covid data

```
useEffect(() => {
  fetch('/data').then(res => res.json()).then(data => {
    setHealthcareCovid(data.HealthcareCovidCases);
    setyesHealthcare(data.HealthcareCovidCasestoday);
    setHospitalisedCovidCases(data.HospitalisedCovidCases);
    settodHospitalised(data.HospitalisedCovidCasestoday);
    setIcucases(data.Icucases);
    setIcuCovidCasestoday(data.IcuCovidCasestoday);
    setMale(data.male);
    settodayMale(data.todaymalecases);
    setFemale(data.female);
    settodayFemale(data.todayfemalecases);
    settotaldeaths(data.totaldeaths2);
    setClustersNotified(data.ClustersNotified);
    setClustersNottoday(data.ClustersNottoday);
    setCommunityTransmission(data.CommunityTransmission);
    setCommunityTrantoday(data.CommunityTrantoday);
    setTravelAbroad(data.TravelAbroad);
    setTravelAbroadtoday(data.TravelAbroadtoday);
    settodaytotaldeaths(data.todaytotaldeaths);
    settotalcases(data.totalcases);
    settodaytotalcases(data.todaycases);
    setconcases(data.concases);
    settodayconcases(data.todayconcases);
  });
});
```

Figure 10 Fetching Covid data

The variables are then declared into an object, as seen in Figure 11 Statistical object, which contains the total number of covid cases for that attribute as well as the

current values for that attribute. It also includes the font-awesome icon code, which is used to build the icon.

```
const SUMMARY = {
  infoStats: [
    {
      today: 'Today',
      Total: 'Total',
      totalName: concases,
      todayFieldName: concasestoday,
      iconlogo: 'fas fa-virus fa-7x',
    },
    {
      today: 'Today',
      Total: 'Total',
      totalName: totalcases,
      todayFieldName: totalcasestoday,
      iconlogo: 'fas fa-viruses fa-7x',
    },
    {
      today: 'Today',
      Total: 'Total',
      totalName: HealthcareCovid,
      todayFieldName: yesHealthcare,
      iconlogo: 'fas fa-book-medical fa-7x',
    },
  ],
}
```

Figure 11 Statistical object

The object is then passed to the map function in the JSX code below, which formats the

```
return (
```

[illegible]

Figure 12 JSX code for statistical backend

components into a box as seen in Figure 6 Interactive map buildThe code in Figure 12 JSX code for statistical backend creates the statistical frontend's box feature.

3.5.2 Irish Interactive Covid Maps

The User should be able to perform graphical features on interactive Geojson Map, so that the user can view Covid cases in their county.

The interactive map was created in the backend using Python and the Plotly Choropleth Map feature. I obtained a Geojson map from an online resource, which helped me to construct an Irish map using geographic coordinates as shown in Figure 13 Reading in Geojson data.

```
##Geojson data
# https://gist.github.com/eoiny/2183412 data/ireland.json

with open(
    "data/ireland.json"
) as myfile:
    geojson = json.load(myfile)
```

Figure 13 Reading in Geojson data

Plotly/dash allows the application to be dynamic by providing a call back feature that allows the map to be modified as seen in Figure 15 Choropleth mapbox and Callback and allows the application to be built in html within python Figure 14 Plotly/Dash html

```
app = dash.Dash(__name__, external_stylesheets=[dbc.themes.BOOTSTRAP])
app.layout = html.Div(
    html.Div(dcc.Graph(id="irl-map", )),
    html.Div(
        dcc.Dropdown(
            id="map-dropdown",
            options=[
                {
                    "label": "Total Infections",
                    "value": "total",
                },
                {
                    "label": "Proportional Infections",
                    "value": "proportional",
                },
                {
                    "label": "Population Census16",
                    "value": "population",
                },
                {
                    "label": "Population Proportion Covid Cases",
                    "value": "PopulationProportionCovidCases",
                },
            ],
            value="total",
        ),
    ),
)
```

Figure 14 Plotly/Dash html

The OSI county dataset is mapped to the Geojson data properties that are read in which then generates a choropleth map box.

```
@app.callback(
    Output("irL-map", "figure"),
    [
        dash.dependencies.Input("map-slider", "value"),
        dash.dependencies.Input("map-dropdown", "value"),
    ],
)
##This build the map and return each map depending on the dropdown value
def update_map(slider, dropdown):

    df_slice = df[
        df["TimeStamp"].str.contains(pd.to_datetime(slider, unit="s").strftime("%Y/%m/%d"))
    ]

    if dropdown == "total":
        fig = px.choropleth_mapbox(
            df_slice,
            geojson=geojson,
            color="ConfirmedCovidCases", # Data to be plotted
            locations="CountyName", # Dataframe entry to match to geojson
            featureidkey="properties.county", # GeoJSON entry to match dataframe
            center={"lat": 53.45, "lon": -8}, # Center map display
            mapbox_style="carto-positron",
            zoom=5.4,
            labels={"ConfirmedCovidCases": "Total Cases"},
            range_color=(0, max(df_slice["ConfirmedCovidCases"])),
        )
        fig.update_layout(title="Total Covid Cases", margin=dict(l=0, r=0, t=50, b=50))

    elif dropdown == "proportional":
        fig = px.choropleth_mapbox(
            df_slice,
            geojson=geojson,
            color="CovidOverPopulation", # Data to be plotted
            locations="CountyName", # Dataframe entry to match to geojson
            featureidkey="properties.county", # GeoJSON entry to match dataframe
            center={"lat": 53.45, "lon": -8}, # Center map display
            mapbox_style="carto-positron",
            zoom=5.4,
            labels={"Covid 19 Over Pop": "% of population"},
            range_color=(0, max(df_slice["CovidOverPopulation"])),
        )
        fig.update_layout(
            title="Proportional Covid Cases", margin=dict(l=0, r=0, t=50, b=50)
        )
```

Figure 15 Choropleth mapbox and Callback

Once the map feature is up and running the application can be featured in an iframe component in the react framework as shown in Figure 16 Iframe

```
componentDidMount() {
    this.setState({ showIframe: true });
}

render() {
    const { showIframe } = this.state;
    return (
        <Design>
            <SectionPanels>
                <div>
                    {showIframe &&
                        <IframeComponent src="http://127.0.0.1:8050/" height="500%" width="100%" />
                    }
                </div>
            </SectionPanels>
        </Design>
    );
}
```

Figure 16 Iframe component

componentFigure 15 Choropleth mapbox and Callback. The build application can be seen in Figure 6 Interactive map build

3.5.1 Covid store components

Function	Store Explanation	Context in relation to the graphs store example	Context in screenshots
Dispatch()	Trigger a state change in the store	dispatch('SET_ALL_CASEDEATHS_STATS', getCases) dispatch('POPULATE_ATTRIBUTE', {fieldName, graphId, storeName: 'CasesandDeaths', });	Figure 17 Graph Store Example
configure Store	Contains the actions	It a class which contains the Actions	Figure 18 Configure Store with actions
Actions	action is a plain object that represents an intention to change the state	SET_ALL_CASEDEATHS_STATS POPULATE_ATTRIBUTE	Figure 18 Configure Store with actions

Store	A store is a state container that stores the state of an application.	CasesandDeaths	Figure 18 Configure Store with actions
-------	---	----------------	---

The user should be able to perform graphical features on a variety of OSI scientific Covid datasets, the user should have ability to access a number of different attributes in a dataset in a graphical analysis feature.

In this section, I will discuss the functional methods which are called in a graph store component. The dispatch method trigger a change in the CasesandDeaths store in Figure 17 Graph Store Example twice, in the 'SET_ALL_CASEDEATHS_STATS' dispatch it has the fetched dataset which is calling the action in Figure 18 Configure Store with actions which then cleans the dataset and populates the CasesandDeaths store. The other instance of the dispatch being called is 'POPULATE_ATTRIBUTE', this method returns the current state of the store and I check all the attributes in the store to see if this attribute populated in the graph or not and if not, I then add this new attribute to the state of the store in order to populate in the graph.


```

useEffect(() => {
  (async () => {
    //getting data from store
    try {
      const getCases = await getcasseanddeaths();
      dispatch('SET_ALL_CASEDEATHS_STATS', getCases);
    } catch (e) {
    }
  })();
  //
}, []);
//Add graph
const handleSelectData = (e, graphId) => {
  const fieldName = e.target.name;
  console.log(graphId)
  dispatch('POPULATE_ATTRIBUTE', {
    fieldName,
    graphId,
    storeName: 'CasesandDeaths',
  });
};

return (
  <div>
    <SectionPanels>
      <PanelMain>
        <SectionPanels>
          <LabelButtons>
            availableAttributes={graphs.avail}
            graphIndex={graphs.id}
            handleSelectData={handleSelectData}
          </LabelButtons>
        </SectionPanels>
        <LineGraph>
          graphId={graphs.id}
          storeName="CasesandDeaths"
        </LineGraph>
      </PanelMain>
    </SectionPanels>
  </div>
);

```

Figure 17 Graph Store Example

```

const configureStore = () => {
  const actions = {

    SET_ALL_CASEDEATHS_STATS: (curState, response) => {
      const copy = curState.CasesandDeaths.graphs;
      // returns alldata the data removed from nested object
      const datasetclean = cleanall(response)

      let ans = copy.map((graph) => (graph.alldata = datasetclean));
      ans = splitdataintoAttrData(copy);
      //returning the modified store with alldata[] and avail[] full
      return { graphs: ans };
    },
    POPULATE_ATTRIBUTE: (curState, { fieldName, graphId, storeName }) => {
      curState[storeName].graphs.map((graph) => {
        //check to see if the graph exists
        if (graph.id === graphId) {
          const attributedata = graph.avail.map((d) => {
            //returns value for the button if its true
            if (d.fieldName === fieldName) {
              d.selected = !d.selected;
            }
            return d;
          });
          graph.avail = attributedata;
          console.log(graph.avail)
          graph.AttributeNameselected = addremoveattr(
            graph.AttributeNameselected, fieldName
          );
        }
        return graph;
      });
      return { dailyAlt: curState[storeName].graphs };
    },
  };
  initStore({actions, {
    CasesandDeaths: CasesandDeaths,

```

Figure 18 Configure Store with actions

3.6 Conclusion

Despite the fact that several issues arose during the implementation, the basic criteria were met and the majority of any issues were resolved. In my view, the collection of technologies that I chose to work with worked very well together. The React architecture fits extremely well with Python, thanks to the use of Flask as middleware.

I think the system's overall design is appealing and consumers may find the program useful. This completes the project's implementation section, with all user stories completed and the project completed. Testing and evaluation will be carried out in order to assess the applications functionality and obtain input on future changes.

Chapter 4: Testing and Evaluation

4.1 Introduction

This chapter will include a summary of my application's testing, usability, and evaluation. Testing is an important part of any software project because it detects errors or vulnerabilities in the code and ensures that the core functionality work as intended. Due to time restraints, I was unable to implement many junit test for my application, but each part of the project was subjected to extensive user acceptance testing.

4.2 Testing

In this section, I will go through the various components of the system and discuss the various approaches to testing that were used in each component. Testing the different components, as with any software application and is a valuable and essential part of the development lifecycle. Since one of main objectives of the project is user experience of the overall system, I focused on satisfying the test requirements of the user experiences.

4.3 Frontend Testing

As I mentioned earlier on I focused on the UX of the frontend of the application and incorporated the seven practices that promote or improve the quality of the user's interaction: Useful, Usable, Desirable, Findable, Accessible and Credible.

Useful – The system is extremely useful as it provides Covid statistics and graphical feature to inform the public on the current situation of the pandemic.

Usable – To Frontend is extremely easy and useable to use, the website is very easy to navigate around and doesn't contain much user input to find your way around the statistical features.

Desirable - A web application should be pleasing to the eye, with minimal design, which I believe I have achieved with my application's design.

Findable – To render a web application 'findable,' make sure that the user can easily navigate between the various components. I built the navigation bar for the graphs on

the interface for this purpose, as it helps the user to quickly switch between the various components.

Accessible - Any users, regardless of cognitive capacity, should be able to use a web application. Other features I would have liked to include to make the app more usable would be making it completely functional with a keyboard for people who may have a disability. I would have also liked to incorporate voice command into the application to facilitate users with visual impairment or disabilities.

Credible - Making a web application look legitimate and professional is a crucial step in gaining confidence in it. This is only possible if the concept is of high quality. I believe I have given the frontend a professional look and have given the projects serious subject matter. All information Datasets I used in the project come from a credible source i.e. OSI. They are constantly updating the datasets to project the most accurate covid numbers available.

If the device ever goes live on a server, I've integrated AWS into the software, which includes a cloud-based error reporting dashboard. The dashboard offers real-time error messaging to the developer, describing where and how the error occurred. This is a great service that would aid in large-scale monitoring of the application if it went into development.

4.4 UX Testing and Feedback

To ensure that I achieve the seven practices that promote user interface, I let my family and friends carry out the user experience by letting them interact with the dashboard. This process helped me identify the usability defects within the application. I didn't give them much detail about the program when letting them demo it because I wanted them to have the experience of discovering the software and finding out how to use the features on their own. The image below is a survey I gave to every tester to give me feedback on the user experience of the application. The survey is vital to get constructive feedback back from the users and it lets them rate overall experience of using the system. The questions are phrased to reflect the seven practices that promote or improve the quality of the user's interaction.

4.4.1 UX Feedback survey

Irish Covid Dashboard User Experience Survey

1. Please select the category that best describes your profession:

2. Was the information easy to find?

☐ Yes ☐ No

3. Was the Covid-19 information clearly presented?

☐ Yes ☐ No

4. Were you able to find the Covid-19 information you were looking for?

☐ Yes ☐ No

5. Were the interactive Covid-19 features in the application easy to use ?

☐ Yes ☐ No

6. What other Covid information should we provide on the Dashboard ?

7. How could we make the Dashboard easy to use?

7. What changes should we make to make the Dashboard more appealing?

8. Overall, how would you rate the Dashboard?

	Great!	Good	Average	Fair	Poor
Accessibility	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Content	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Presentation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

8. Please enter your email address (optional):

Figure 19 UX Feedback Survey

4.4.2 Survey Feedback from users

Users	<i>User One</i>	<i>User two</i>	<i>User Three</i>	<i>User Four</i>
Q1	Civil Servant	Construction	Financial	Student
Q2	Yes	Yes	Yes	Yes
Q3	No	Yes	Yes	No
Q4	Yes	Yes	No	Yes
Q5	Yes	Yes	Yes	No
Q6	N/A	Other pandemics in history	Vaccine dataset	Covid cases for schools
Q7	Voice assistant	Quick Navigation menu	N/A	Alerts for first time users
Q8	Brighter Appearance	N/A	Bigger Font	N/A
Q9 Accessibility	Average	Good	Good	Average
Q9 Content	Good	Good	Average	Great
Q9 Presentation	Good	Great	Good	Great

Table 4 UX Feedback Table

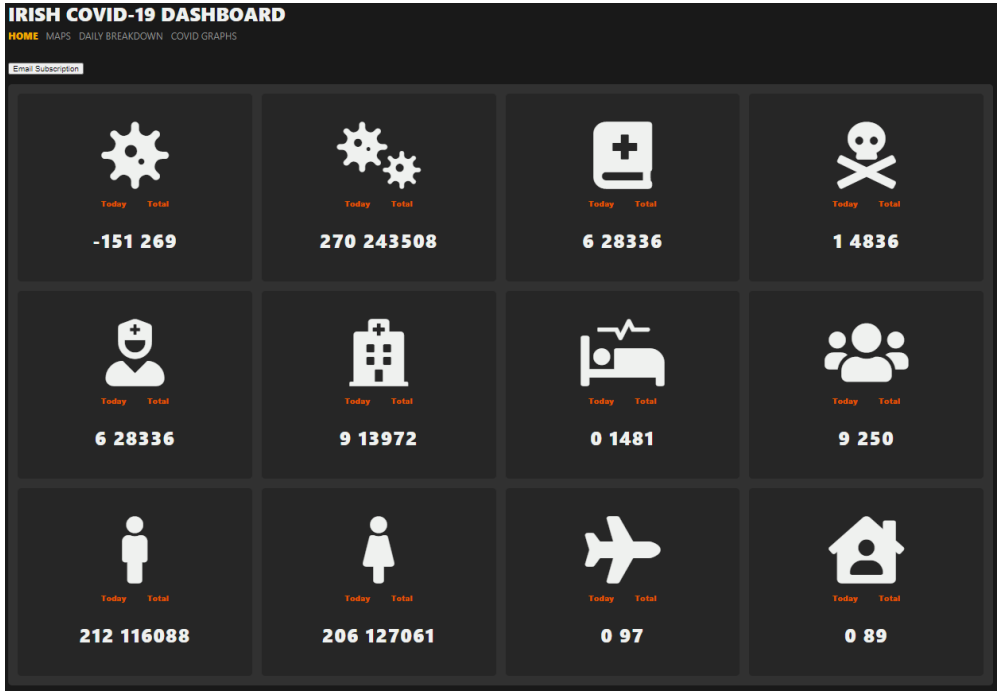
4.5 Black-Box Testing

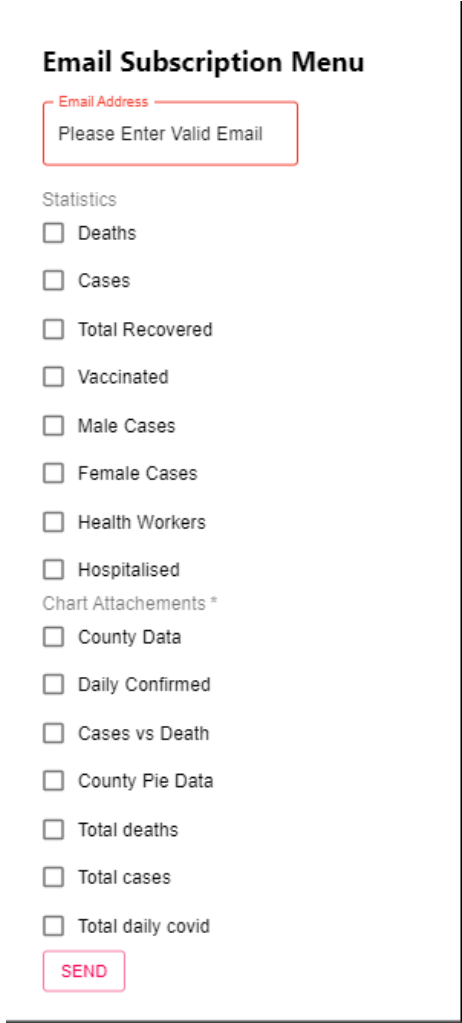
Black-Box testing examines the functionality of an application when the tester is unaware of the code that underpins each test. By entering irregular inputs that deviate from a given element's standard function, the tester tries to find errors. For instance, if a user attempts to login and enters incorrect information, they should be greeted with an error message. For any inputs made, the consumer should be mindful of the output at all times and Black-Box testing means that all options are taken into account.

4.5.1 Fundamental Testing

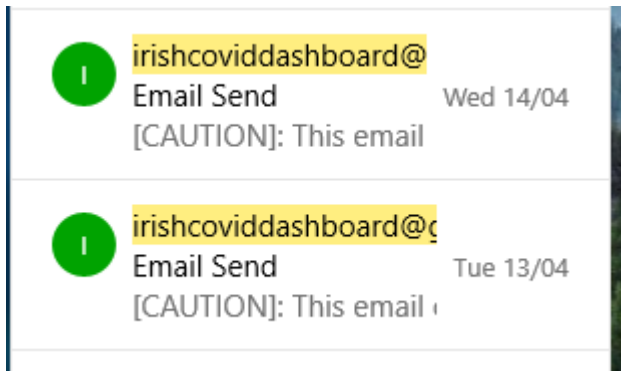
Functional tests ensure that the application performs as intended, with test cases written in a user-friendly format. Users reviewed the software after it was finished to give feedback to make sure the functionality met standards.

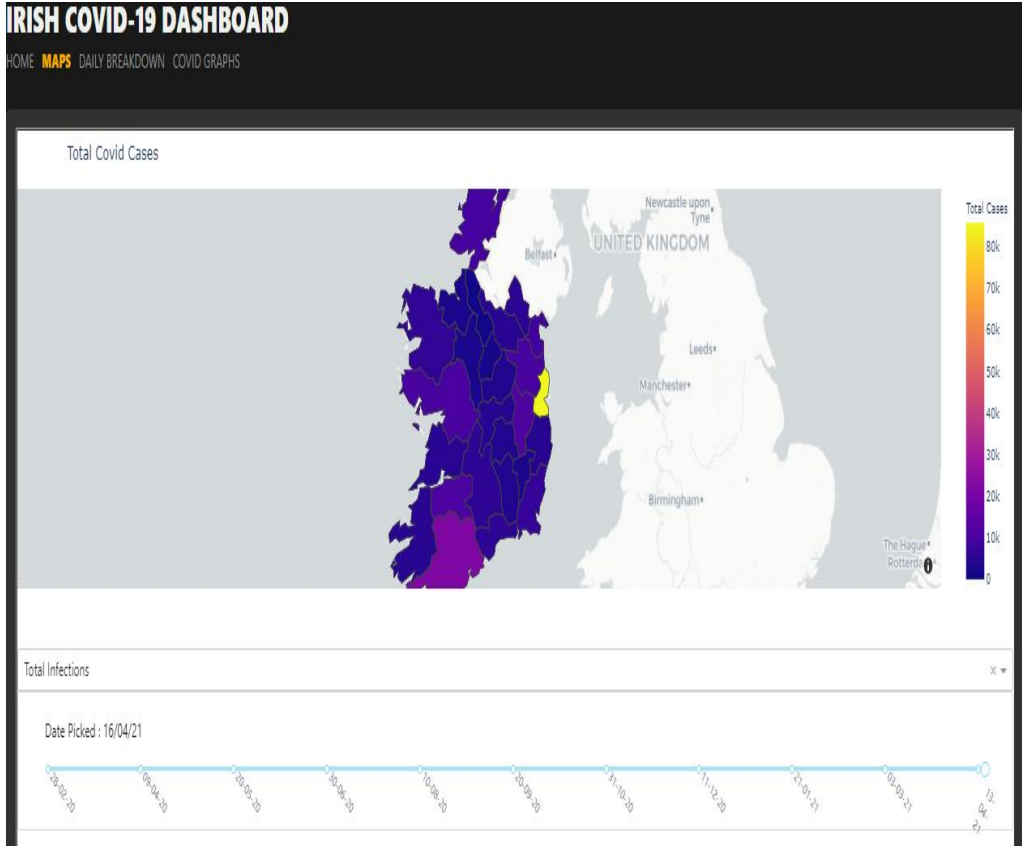
In the table below, I built a template table and filled in the test definition and expected result for each. After that, I had a member of my family observe each test to make sure the real result mirrored the expected result. The following are the outcomes:

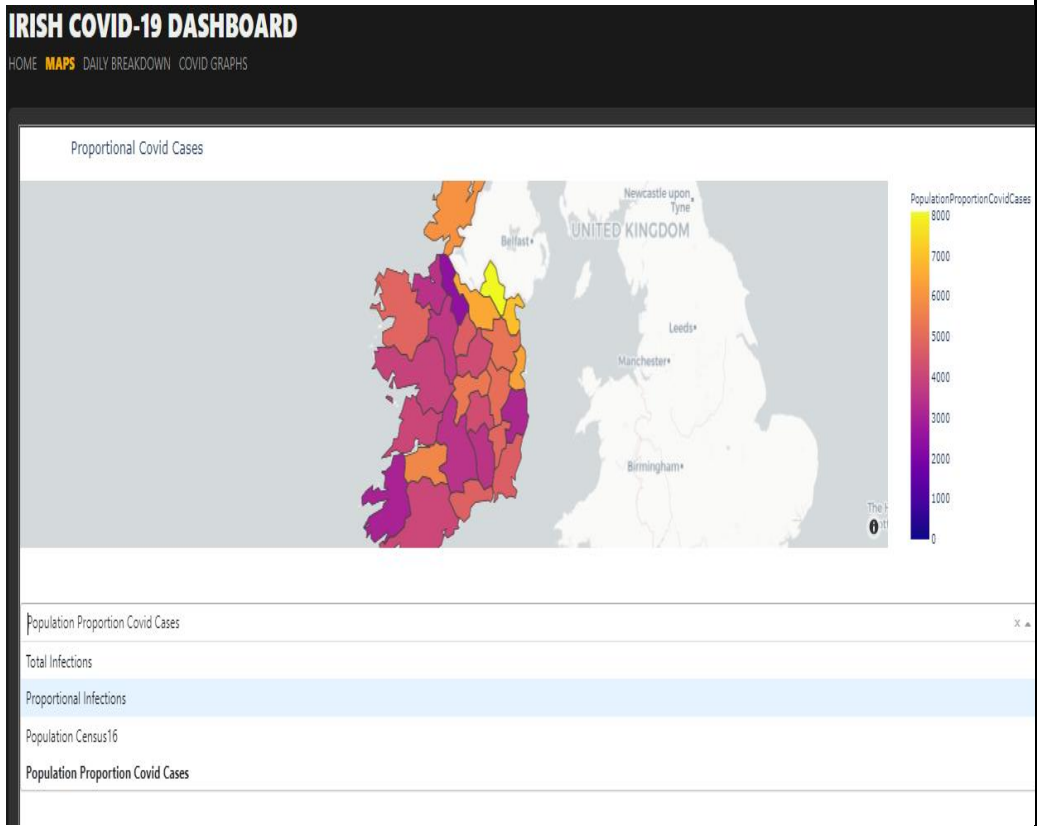
User Story	As an Irish citizen, I want to view Covid statistics, so that I can inform myself about the pandemic.
Test Case ID	1
Description	View statistics frontend page
Actions	View Home page
Expected Results	<p>Home page should consist of Covid statistics with icons such as below:</p>  <p>The screenshot shows the 'IRISH COVID-19 DASHBOARD' with a dark theme. It features a grid of 12 cards, each with an icon, a 'Today' and 'Total' count, and a large numerical value. The cards represent: 1. Gear icon: -151 269. 2. Two gears icon: 270 243508. 3. Medical cross icon: 6 28336. 4. Skull and crossbones icon: 1 4836. 5. Person with shield icon: 6 28336. 6. Hospital icon: 9 13972. 7. Bed with heart rate icon: 0 1481. 8. Group of people icon: 9 250. 9. Single person icon: 212 116088. 10. Female person icon: 206 127061. 11. Airplane icon: 0 97. 12. House icon: 0 89.</p>
Test Status	Passed

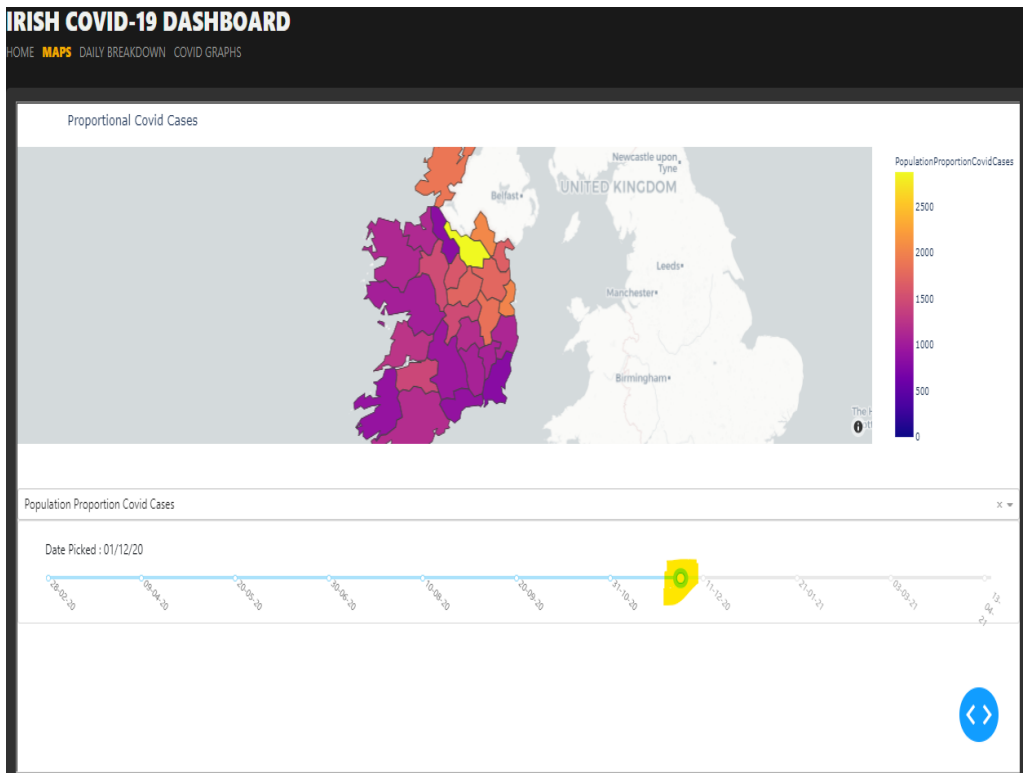
User Story	As an Irish citizen, I want to subscribe to the email Subscription service, so I can get an update on Covid-19 statistics.
Test Case ID	2
Description	Subscription button for email service
Actions	View Home page Click on email subscription button
Expected Results	<p>Modal textbox field should appear looking for user selection details such as below:</p> 
Test Status	Passed

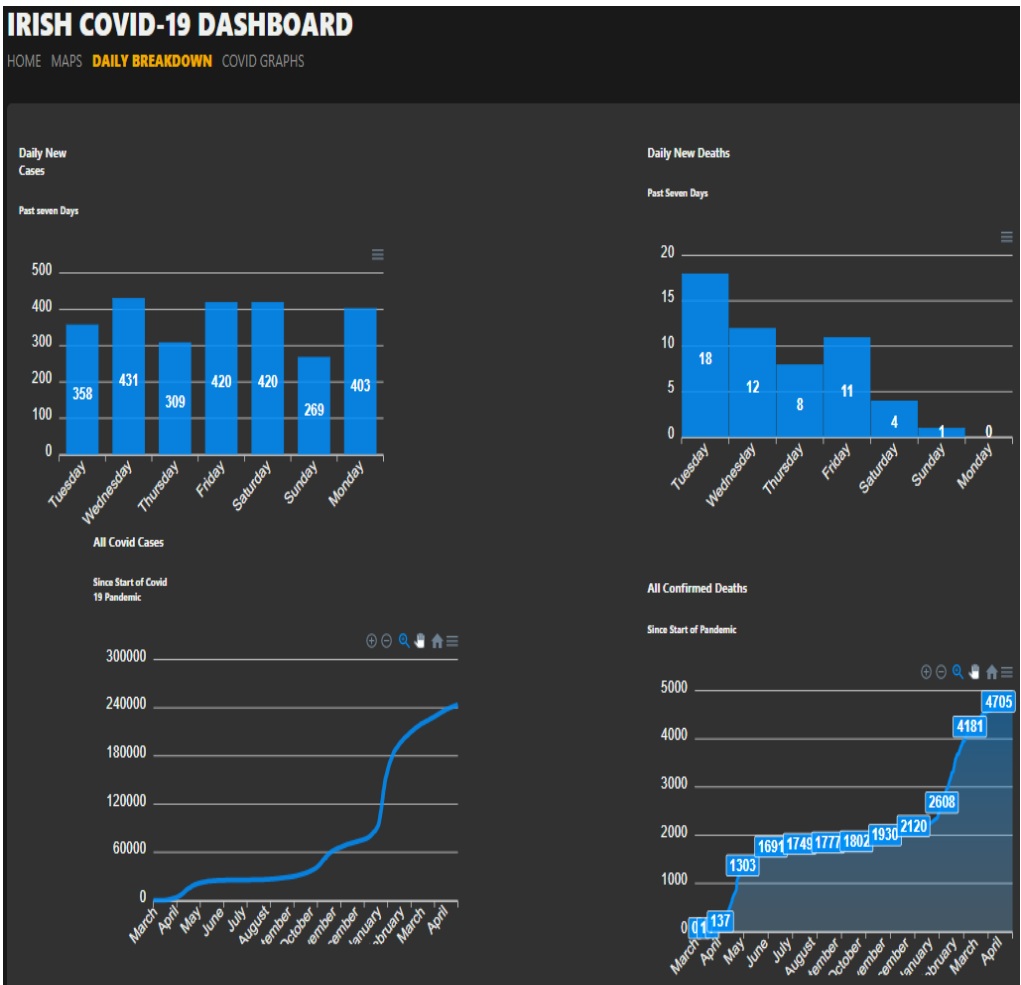
User Story	As an Irish citizen, I want to enter my email and covid subscription options, so I can receive a custom covid subscription email.
Test Case ID	3
Description	Sign up Covid-19 subscription email service
Actions	Enter Email and select tick boxes for covid information. Click on submit button
Expected Results	<p>Covid Subscription email should be received in the user inbox, Example below:</p>
Test Status	Passed

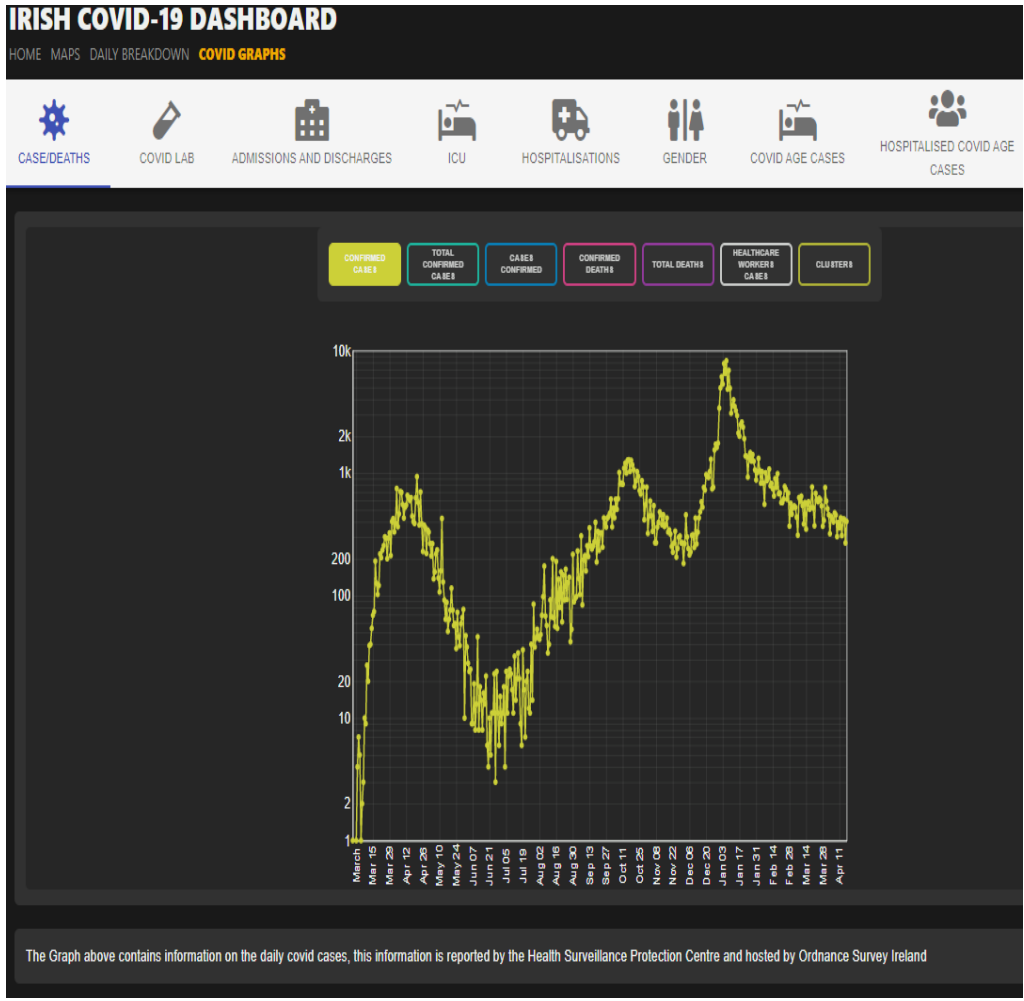
User Story	As an Irish citizen, I want to receive Covid email newsletter every day, so I can be updated daily on covid cases in Ireland.
Test Case ID	4
Description	Email Subscription newsletter send to user every day
Actions	Check email inbox
Expected Results	<p>User should receive custom email subscription newsletter everyday depending on covid statistics selected. Example below:</p> 
Test Status	Passed

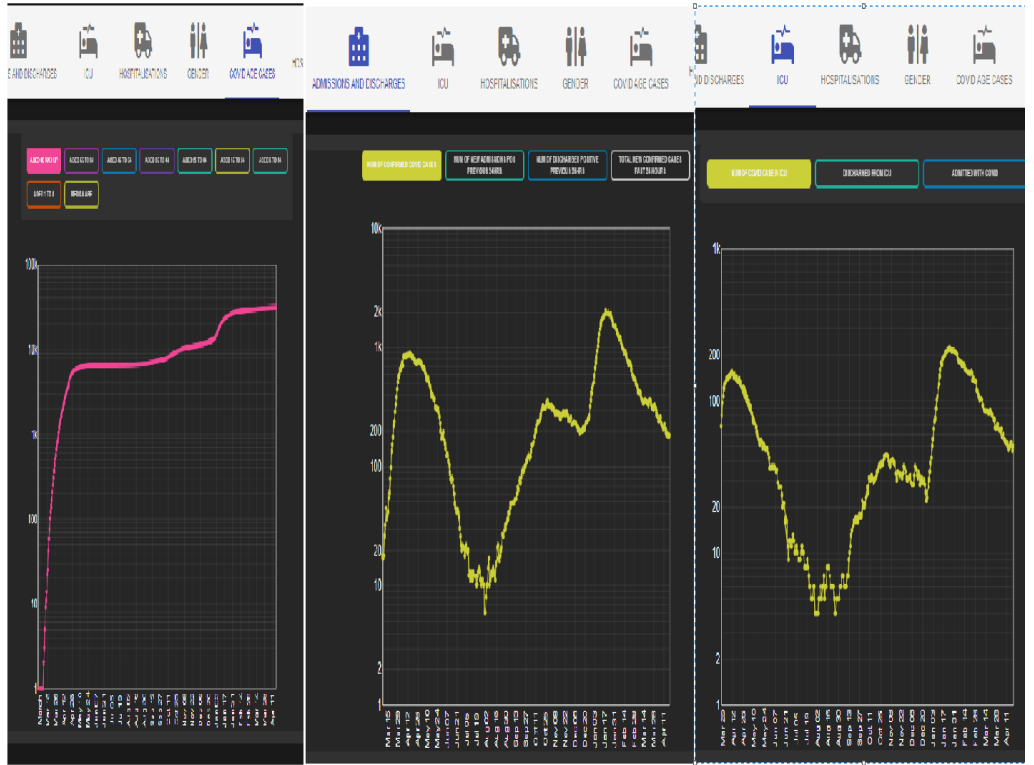
User Story	As an Irish citizen, I want to view County Covid interactive map, so that I can inform myself about Covid cases in my County.
Test Case ID	5
Description	County Covid interactive map
Actions	View map page Interact with the Irish map
Expected Results	<p>Rendered Irish interactive map should be present.</p>  <p>The screenshot displays the 'IRISH COVID-19 DASHBOARD'. At the top, there are navigation links: HOME, MAPS (highlighted), DAILY BREAKDOWN, and COVID GRAPHS. The main section is titled 'Total Covid Cases' and features a map of Ireland. The map's regions are color-coded according to a legend on the right, which ranges from 0 (dark purple) to 80k (yellow). The map shows that some regions in the east of Ireland are colored yellow, indicating higher case counts. Below the map, there is a section for 'Total Infections' with a line graph. The graph has a date picker set to '16/04/21' and shows a timeline of data points from 01-02-20 to 19-04-21.</p>
Test Status	Passed

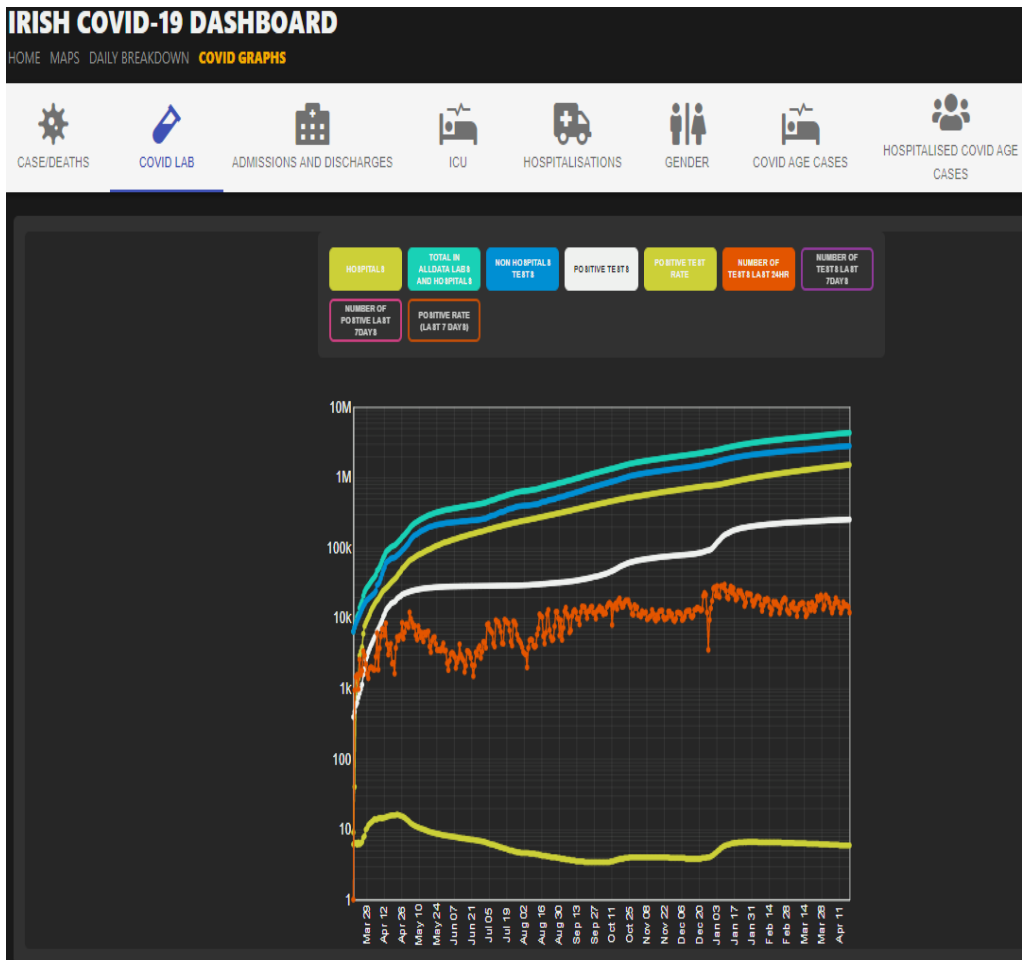
User Story	As an Irish citizen, I want to select different County Covid interactive map statistics, so that I can inform myself about different Covid statistics cases in my County.
Test Case ID	6
Description	Selection panel for choosing different County Covid statistics
Actions	View map page Select statistic from the selection panel. Interact with the map
Expected Results	<p>User has ability select different statistic to render in the page</p> 
Test Status	Passed

User Story	As an Irish citizen, I want the ability to time lapse through County statistics, so that I can inform myself about different covid statistics cases in my County.
Test Case ID	7
Description	The user has ability with slider panel time lapse through Covid statistics on the map
Actions	View map page Select statistic from the selection panel Interact with slider panel Interact with the map
Expected Results	<p>The user has ability to time lapse through covid statistics on the interactive Irish map while carrying out other covid map features.</p> 
Test Status	Passed

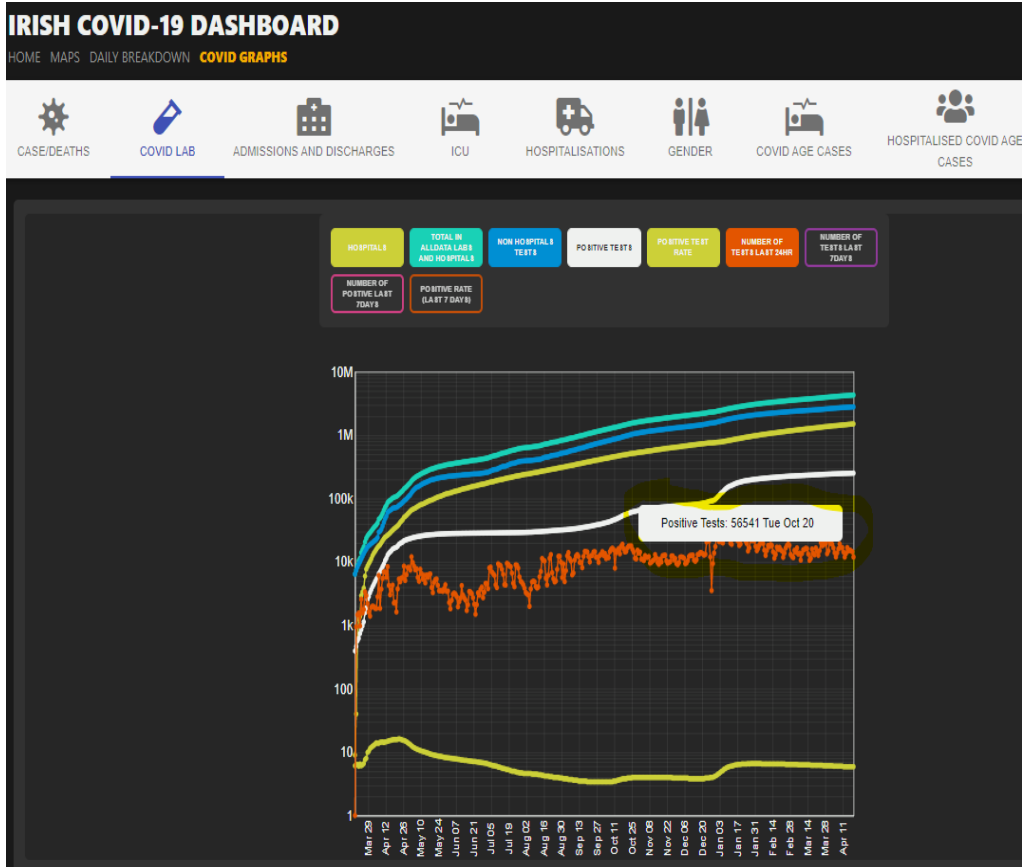
User Story	As an Irish citizen, I want to view weekly breakdown of Irish Covid statistics, so I can analyse Irish statistics for the pandemic.
Test Case ID	8
Description	View Weekly breakdown of covid statistics graphs
Actions	View Daily breakdown page
Expected Results	<div>The user should be presented with interactive graphs for covid-19 statistics for the week.</div> <div></div>
Test Status	Passed

User Story	As an Irish citizen, I want to view interactive covid graph, so that I can investigate multiple different Covid datasets.
Test Case ID	9
Description	Investigate Covid statistics on interactive graph
Actions	View Graph store page View interactive graph
Expected Results	<p>Graph store page should consist of interactive graph loaded with covid graphical information.</p>  <p>The Graph above contains information on the daily covid cases, this information is reported by the Health Surveillance Protection Centre and hosted by Ordnance Survey Ireland</p>
Test Status	Passed

User Story	As an Irish citizen, I want to click on multiple different graph stores, so that I can investigate multiple different covid datasets.
Test Case ID	10
Description	Investigate different covid graph stores
Actions	View Graph store page Click on tab icons to view a different store
Expected Results	Graph store page should consist of menu tab which consist of multiple different graph stores 
Test Status	Passed

User Story	As an Irish citizen, I want to click a covid statistic attribute, so that I can view the statistic in the interactive graph.
Test Case ID	11
Description	Adding a new attributes to the graph
Actions	View Graph store page Click on tab icons to view different store Click on a button under tab icon section to add attribute to the graph
Expected Results	<p>Once a button attribute is clicked a new line should appear in the interactive graph and button should be highlighted</p>  <p>The screenshot shows the 'IRISH COVID-19 DASHBOARD' with a navigation bar including 'HOME', 'MAPS', 'DAILY BREAKDOWN', and 'COVID GRAPHS'. Below the navigation bar are icons for 'CASE/DEATHS', 'COVID LAB', 'ADMISSIONS AND DISCHARGES', 'ICU', 'HOSPITALISATIONS', 'GENDER', 'COVID AGE CASES', and 'HOSPITALISED COVID AGE CASES'. The main area displays a line graph with a logarithmic y-axis (1 to 10M) and a date-based x-axis (Mar 28 to Apr 11). The graph shows several data series: 'TOTAL IN ALL DATA LABS AND HOSPITALS' (red line, highest values), 'HOSPITALS' (blue line), 'NON HOSPITALS TESTS' (green line), 'POSITIVE TESTS' (yellow line), 'POSITIVE TEST RATE' (orange line), 'NUMBER OF TESTS LAST 24HR' (purple line), 'NUMBER OF TESTS LAST 7 DAYS' (brown line), 'NUMBER OF POSITIVE LAST 24HR' (pink line), and 'POSITIVE RATE (LAST 7 DAYS)' (grey line). The 'TOTAL IN ALL DATA LABS AND HOSPITALS' series shows a steady increase from approximately 10k in late March to over 1M by early April. The 'POSITIVE TEST RATE' series shows a peak in late March followed by a decline and then a slight increase in early April.</p>
Test Status	Passed

4.6 Evaluation

User Story	As an Irish citizen, I want to hover over attributes on the graph, so that I can investigate the exact date and value for that covid statistic.
Test Case ID	12
Description	Investigating date field and statistic value for the covid attribute
Actions	View Graph store page Click on tab icons to view different store Click on a button under tab icon section to add attribute to the graph Hover over attribute line to investigate the values
Expected Results	<p>A box should appear for the attribute value with extract covid value and date depending on where the user leaves the mouse pointer.</p> 
Test Status	Passed

I hope to address these issues before the project's final submission, but the project is otherwise stable. I followed acceptance guidelines and checked the functionality during development, which I believe is the reason why the web application did not show many issues.

4.7 Conclusion

From what I've discussed in this segment, we can see the testing techniques I used and how they applied to my thesis. In the following chapter, I will wrap up this entire thesis. This Section will allow me to demonstrate everything I've done while also offering a summary of recommendations.

Chapter 5: Conclusions

5.1 Introduction

This chapter will dissect the project's findings as well as potential recommendations I'd make to someone developing a similar project or if this project is continued in the future. First, this section will determine whether or not the general and precise goals have been met.

5.1.1 Fulfilment of the objective

The general objective of the project is to develop a number of graphical and statistical features to inform people about the Covid-19 pandemic. A goal of the project was to create a Covid application using a large number of OSI datasets and to gain a better understanding of other Covid dashboards on how they are built.

I feel these objectives have been fulfilled to a satisfactory standard, as I have implemented over five covid OSI datasets in the covid dashboard and I feel I have developed graphical and statistical features which will help and inform the public about the pandemic. The application is very adaptable to different people needs, if the user looking for an scientific background to covid the user can interact with the laboratory dataset and admission and discharges datasets. Also, if the consumer prefers a more straightforward approach, they can look at a static breakdown of the numbers of covid cases and communicate with graphs depicting the number of covid cases and deaths. Throughout this process, I gained a better understanding of other Covid Dashboard

applications, which assisted me in determining which features to include in my Covid dashboard.

Overall, I believe my application has satisfied my context and rational, the application contains a large number of datasets from a reputable source the OSI and I believe I have projected this information clearly without a narrative behind it. Every user has the same access to all the data which was a major objective in the context of the application as people will feel everyone is being open about the pandemic.

5.2 Reflection

Once I chosen my project idea for my final year project, I was eager to get started. I used the three presentation dates as a checklist to ensure that specific user stories were completed by specific deadlines. I wanted to make sure there was a physical application to display the examiners on December 1st, the first presentation date, so I immediately began by creating a simple html and JQuery frontend. I implemented my backend matplotlib charts into the frontend and integrated my Geojson interactive county graph in the front-end page. I believe this was a good base to develop from, the feedback which I received from the supervisors was to incorporate more of a dynamic frontend by using a java script framework.

I'm glad I got this feedback so early in the project cycle because it gave me time to learn a new JavaScript framework, which shaped the project's next two demos. The key advantages of switching to a framework are that it makes the project more adaptable and the inclusion of components makes the frontend easier to manage and create graphical functionality.

When carrying out the project they were days when I couldn't find the drive to keep working on the project due to other commitments being carried out in other modules and there were times when I thought I'd never be able to solve a problem. There were also moments when I enjoyed working on the different aspects of the application's implementation.

Overall, I enjoyed the process of researching a topic, creating a structure, and integrating the different components, at least for the most part. The topic I picked Covid dashboard is extremely relevant do the current pandemic. I can now think back and

see how beneficial this phase was to me as a student and as a soon-to-be-graduated software developer. Not only in academia, but also in the software industry, research and documentation are important.

5.3 Recommendations

Due to time, ability and outside feedback constraints, several elements of this application that I wanted to include were omitted. Unfortunately, OSI didn't publish datasets which I would have liked to implement into the project such as vaccine datasets and more statistical data on Counties in Ireland. I feel I should have made more of an effort earlier in the project to contact OSI and get the possibility of permission to access the datasets I mentioned above.

If I was to carry out the project again, I would have liked to implement interactive graph for County Covid data, I have integrated such features into an interactive geojson map but however I think if the user had ability to carry out this feature on a chart and have automatic time lapse through the Covid data for each County in Ireland to the present time.

I strongly suggest that a news feed aspect be added to the project, I believe that if users had access to news articles related to the Covid figures in my application, they would be able to comprehend the project's static and graphical features more easily. It would lend legitimacy to the datasets by allowing them to be compared to broadsheet newspapers.

I believe that implementing the project on a server would be beneficial for testing purposes to ensure that the project is production-ready, as well as for receiving input from users to ensure that the project is fulfilling its user interface requirements.

Next time, if I was to implement a framework like react and introduce subcomponents of redux, I think the introduction of a testing harness for the project like react Jest which allows development of test cases for react projects and allows the testing of redux hooks.

I would have liked to make the project more dynamic in relation to the datasets as a lot of the code is repetitive for building the stores and would have given project level of flexibility of implementing any dataset from the OSI without manual inputting of

attribute names. A lot of attributes from the datasets I used were being removed by the OSI. As a result, I had to manually update the store which can take time, therefore being fully dynamic would solve a lot of issues and certainly is a limitation of the system.

5.4 Conclusion

My reflections and the conclusion chapter come to an end with that. Overall, I consider this to have been a beneficial and enjoyable experience, and I now feel prepared and enthusiastic to pursue a career in Software Development. I'd like to keep working on this app because the concept has grown on me and it has turned into a project, I'm proud of. I thoroughly enjoyed the development cycle of the project. I'd like to thank my supervisor Joseph McNamara, once more for his support.

References

- [1] P. H. B. Paul Hosford, "Tony Holohan hits out at 'hurtful' fake news about nurse deaths," irishexaminer, 01 April 2020. [Online]. Available: <https://www.irishexaminer.com/news/arid-30991398.html>.

Glossary

Acceptance Criteria - A set of tests which verify the completion of a user story.

JSON - JSON is a lightweight data storage and transport format.

Python - Python is a high-level, general-purpose programming language that is interpreted.

React - React is an open-source JavaScript front-end library for creating user interfaces or UI components.

Flask - Flask is a Python-based micro-web development platform.

Material UI - React components make web development quicker and simpler.

Matplotlib - Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy

Chart.js - Chart.js is a free, open-source JavaScript data visualization library.

Dash/Plotly - Dash/Plotly is the most common and well-respected Python platform for developing machine learning and data science web applications.

jQuery - jQuery is a JavaScript library that makes traversing and manipulating the HTML DOM tree easier.

Bootstrap - Bootstrap is a sensitive CSS platform that is free and open-source.

Jest - Jest is a JavaScript testing framework maintained by Facebook.

JavaScript - JavaScript, also known as JS, is a programming language that adheres to the ECMAScript specification.

Redux - Redux is a predictable state container for JavaScript apps.

Geojson - GeoJSON is an open standard format for representing basic geographical features as well as non-spatial attributes.

SQLAlchemy - SQLAlchemy is a Python-based open-source SQL toolkit and object-relational mapper.

APScheduler - Advanced Python Scheduler is a Python library that allows you to schedule the Python code to run later, either once or on a regular basis.

Jinja - Jinja is a modern and designer-friendly Python templating language based on Django's templates.

Npm - Npm is a JavaScript package manager for the programming language.

D3 - D3.js is a JavaScript library that allows web browsers to create immersive, interactive data visualizations.

ApexCharts - ApexCharts is a modern charting library that aids developers in creating attractive and immersive web page visualizations.

Wireframe - A wireframe is a diagram or model that programmers and designers may use to learn about and talk about the layout of the software or website they're creating.

Font Awesome - Font Awesome icons are made from scalable vectors, allowing you to use high-quality icons on any screen size.

Choropleth - A choropleth map is a kind of thematic map that colorizes a series of pre-defined areas.

Store - A store is a state container that stores the state of an application.

List of Abbreviations

OSI - Ordnance Survey Ireland

CSS – Cascading Style Sheet

HSE - Health Service Executive

ICU - Intensive Care Unit

Covid - Coronavirus

JSX – JavaScript XML

JS – JavaScript

JSON - JavaScript Object Notation

HTML - HyperText Markup Language

SMTP - Simple Mail Transfer Protocol

HTTP - Hypertext Transfer Protocol

WHO - World Health Organisation

NPHE - Institute of Public Health in Ireland

APP - Application

UX - User Experience

APScheduler - Advanced Python Scheduler

Npm - Node Package Manage

AWS - Amazon Web Service

Appendix A: Research Covid Dashboard'

A.1 Appendix WHO Dashboard

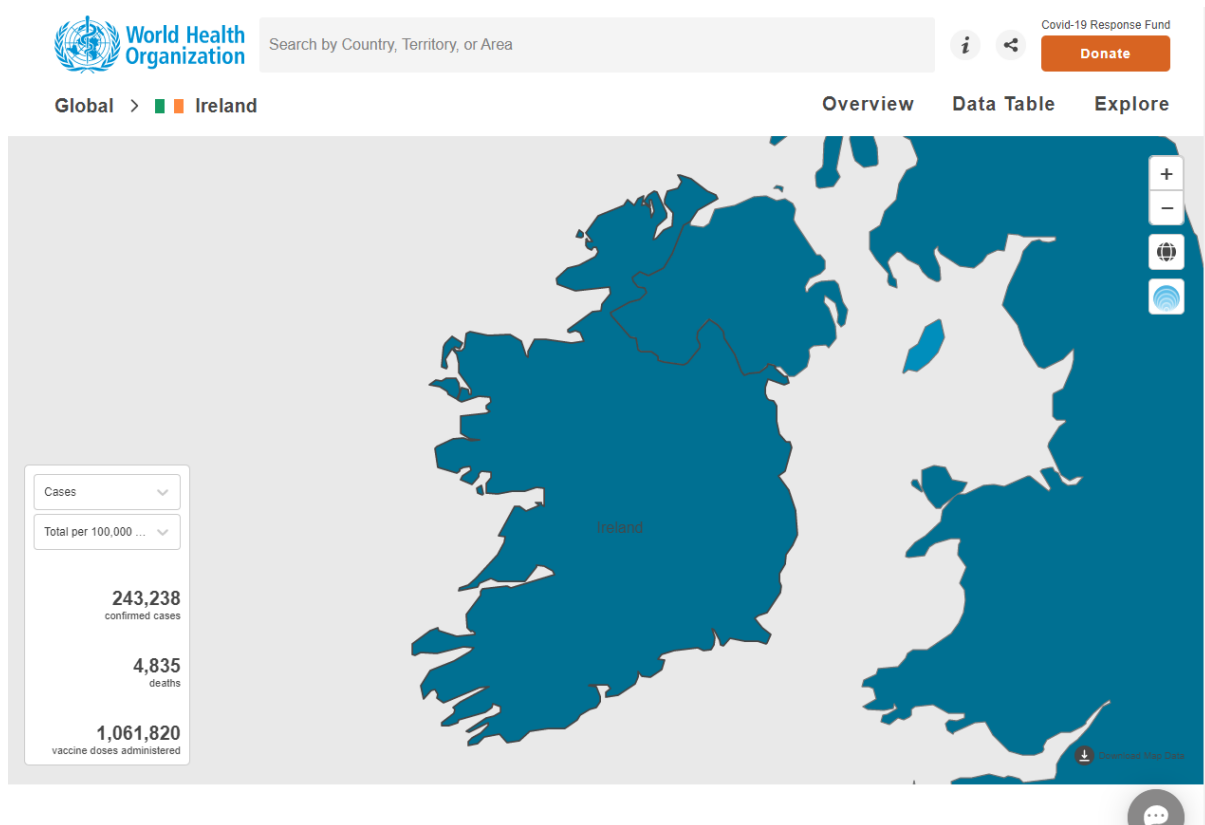


Figure 20 WHO

A.2 Appendix HSE Ireland's COVID-19 Data Hub

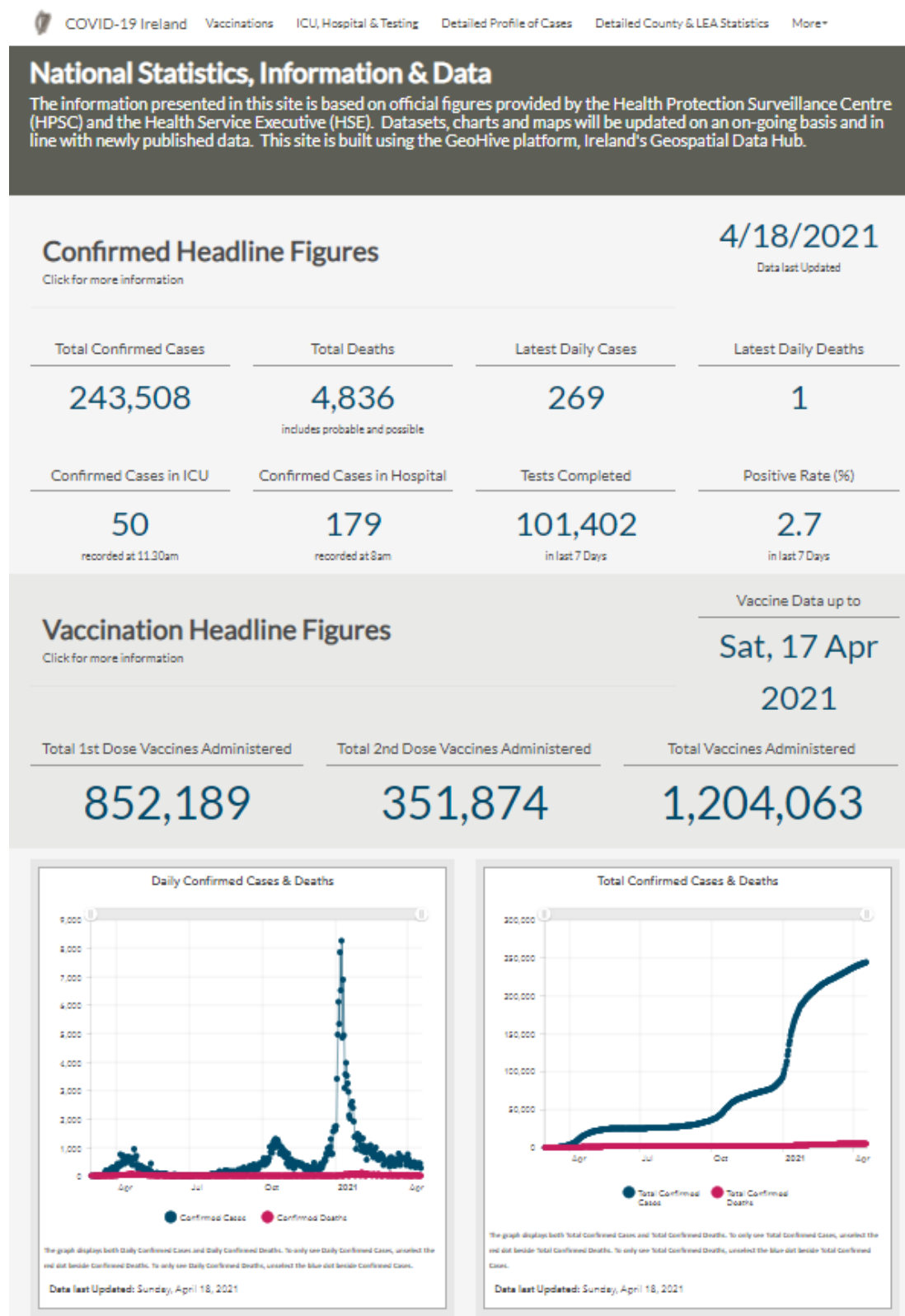


Figure 21 COVID-19 Data Hub

A.3 Appendix Dashboards with minimal information

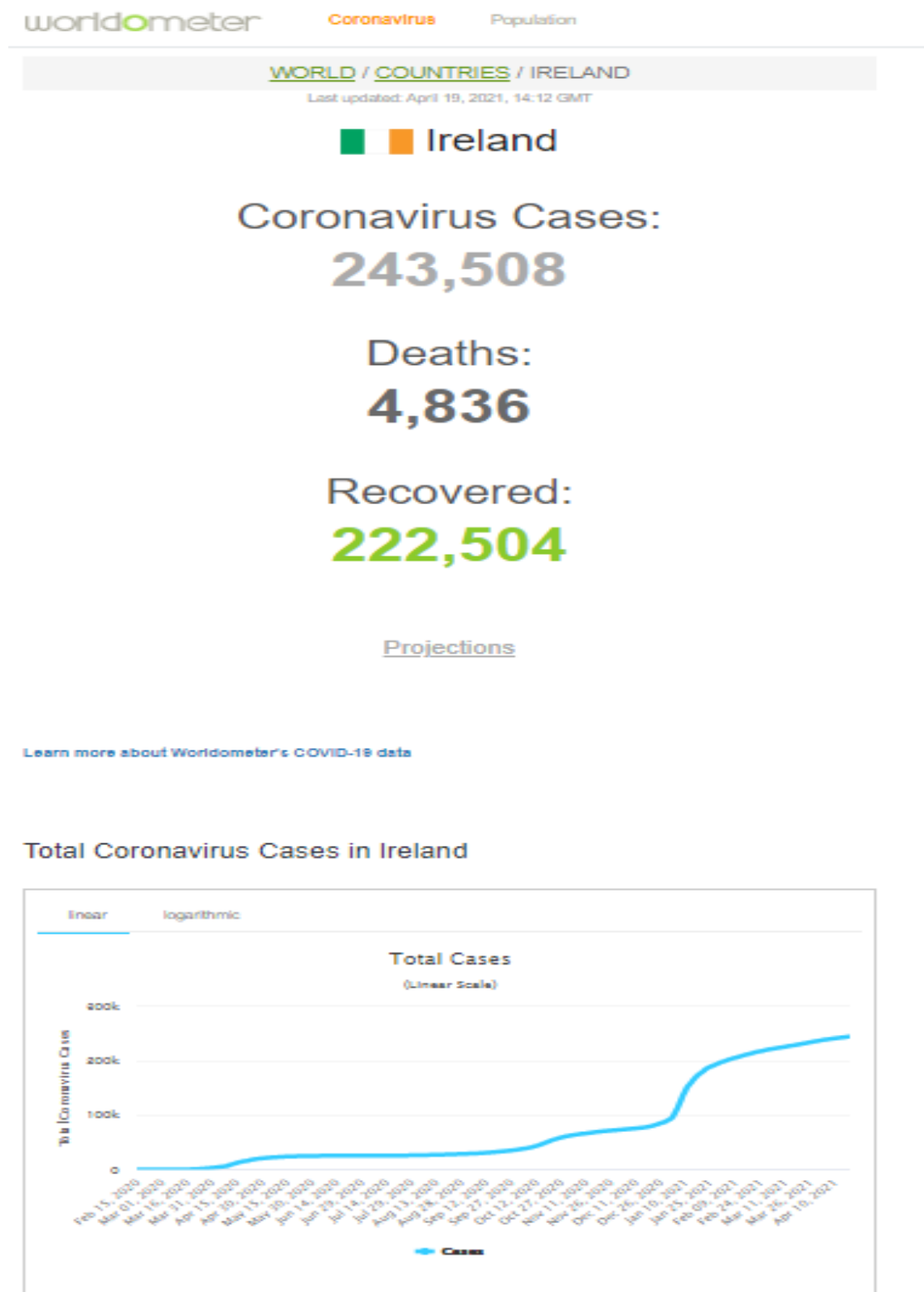


Figure 22 Minimal information example