

**Khoa Nguyen**

**COVID-19 TRACKING DASHBOARD**

**Thesis**

**CENTRIA UNIVERSITY OF APPLIED SCIENCES**

**Degree Programme**

**May 2022**



**ABSTRACT**

<b>Centria University of Applied Sciences</b>	<b>Date</b> May 2022	<b>Author</b> Khoa Nguyen
<b>Degree programme</b> Information Technology		
<b>Name of thesis</b> COVID 19 TRACKING DASHBOARD		
<b>Centria supervisor</b> Jari Isohanni		<b>Pages</b> 24
<b>Instructor representing commissioning institution or company</b>		
<p>Covid 19 and pandemics in general have been a concern of many people. Therefore, having a tool to show intuitive and accurate indexes, graphs of the Covid 19 affection helps people to raise consciousness to protect themselves from it.</p> <p>Covid 19 Tracking Dashboard provides a variables info about the current pandemic with numbers of affected, recovered and death cases. It also has a list of countries affected, the graphs of cases and videos about Covid 19.</p> <p>For this project, HTML, CSS and JavaScript was used to develop the front end, and the data was provided by Covid19 API via Postman. The website is also responsive to be viewed on mobile devices.</p>		

<p><b>Key words</b> HTML, CSS, Javascript, Bootstrap, Responsive</p>
--

## **CONCEPT DEFINITIONS**

### **API**

Application Programming Interface

### **CSS**

Cascading Style Sheet

### **CSSE**

Centre for Systems Science and Engineering

### **HTML**

Hyper Text Markup Language

### **JS**

Javascript

**ABSTRACT**  
**CONCEPT DEFINITIONS**  
**CONTENTS**

<b>1 INTRODUCTION.....</b>	<b>1</b>
<b>2 PRODUCT VISION.....</b>	<b>2</b>
<b>3 TOOLS AND TECHNOLOGY .....</b>	<b>3</b>
<b>3.1 HTML.....</b>	<b>3</b>
<b>3.2 CSS .....</b>	<b>3</b>
<b>3.3 ES6.....</b>	<b>4</b>
<b>3.4 Bootstrap.....</b>	<b>4</b>
<b>3.4.1 Bootstrap File Structure.....</b>	<b>4</b>
<b>3.4.2 Bootstrap Grid .....</b>	<b>5</b>
<b>3.5 Apex Charts .....</b>	<b>6</b>
<b>3.6 Git and Version Control .....</b>	<b>6</b>
<b>3.6.1 Git.....</b>	<b>7</b>
<b>3.6.2 Version Control.....</b>	<b>7</b>
<b>3.6.3 GitHub .....</b>	<b>7</b>
<b>3.7 Covid API.....</b>	<b>8</b>
<b>4 DESIGN PHASE .....</b>	<b>9</b>
<b>4.1 Planning .....</b>	<b>9</b>
<b>4.2 Layout.....</b>	<b>10</b>
<b>4.2.1 Card-based Layouts.....</b>	<b>11</b>
<b>5 WEBSITE IMPLEMENTATION .....</b>	<b>13</b>
<b>5.1 Navbar.....</b>	<b>13</b>
<b>5.2 Main content .....</b>	<b>14</b>
<b>5.2.1 Country Selector .....</b>	<b>15</b>
<b>5.2.2 Tracking counter.....</b>	<b>15</b>
<b>5.2.3 All time chart.....</b>	<b>17</b>
<b>5.2.4 Videos .....</b>	<b>19</b>
<b>5.2.5 Top Countries.....</b>	<b>20</b>
<b>5.3 Footer .....</b>	<b>21</b>
<b>6 CONCLUSION .....</b>	<b>22</b>
<b>REFERENCES.....</b>	<b>9</b>
<b>APPENDICES</b>	

## 1 INTRODUCTION

The part of a website that interacts with the user is called the front-end. Everything on the Internet is a combination of HTML, CSS, and JavaScript controlled by the computer browser, from fonts and colors to drop-down menus and sliders. The look and feel of a website, as well as the architecture of the user experience, are the responsibility of front-end engineers.

Front-end developers must be fluent in three languages: HTML, CSS, and the JavaScript programming language. The developers also need to have knowledge with frameworks such as Bootstrap, Foundation, Backbone, AngularJS, and EmberJS, in addition to those languages, to guarantee that content appears nice on all platforms, and libraries such as jQuery and LESS, encapsulate code into a more useful and time-saving form.

This project aims to build a front-end Covid-19 dashboard to show the data of the current situation about the pandemic.

A data dashboard is a tool that provides a centralized, interactive means of monitoring, measuring, analyzing, and extracting relevant insights from different datasets in key areas while displaying information in an interactive, intuitive, and visual way.

In this thesis, the process of planning, designing and implementing the website is presented. Each part will have their own chapter and will show the steps to building the website, from planning, sketching to the HTML, CSS and JavaScript code implementing to build the project.

## 2 PRODUCT VISION

In order to get from an idea to a product, a vision for the product is essential to help tracing the initial path. Product Vision or Product Vision Statement is a long-term vision, served as a guidance for the development of a product. The product manager, or in this case the developer, to stay organized and navigate through the process of developing the product. It also helps the customers or the users to be more connected with the product. (Pellegrino 2020.)

A product vision plays an important part in the development of the product due to its clear statement, which provides necessary context around a product, especially its goal. Therefore, it can be the compass to guide the product owner. (Pellegrino 2020.)

This project's vision is to create an informative Covid-19 dashboard that shows data about the pandemic intuitively and easily for people to view, therefore informing and raising people's awareness about the disease. Doctors can use these data to monitor and prepare the required medicines. Furthermore, the authorities can also view the data and make decision on how to deal with the current situation. Not only this project can track Covid-19 data, but it can be extended to track other disease or viruses, with a suitable API.

### 3 TOOLS AND TECHNOLOGY

In this section, the technology stack used in this project will be explained. Web stack is divided into two broad technology categories namely front end (client-side) and back end (server-side). Each of these areas has their respective roles to play in web development and comes with a comprehensive set of tools. (Patel 2020.)

Within this project, only the client side will be present. The client side of an app is the part that is visible to users and represents the “face” or the “facade” of an application. For that reason, it is commonly referred to as the “front-end.” The web stack technologies associated with this development layer are not numerous, HTML, CSS and Javascript for example. JavaScript, HTML, and CSS are regarded as conventional standards in this field, substitute option can be found such as Apache Flex, and others. (Roznovsky 2020.)

#### 3.1 HTML

HTML is a markup language that is used to show documents in a web browser. HTML was one of the primary breakthrough technologies used to launch the world's first website on August 6, 1991, and was first developed by Tim Berners Lee in 1990 while working at the European Organization for Nuclear Research (CERN). HTML has been constantly updated and expanded since then, but its essential function of formatting and structuring web pages has remained the same. (Glass 2020.)

#### 3.2 CSS

CSS is a style sheet language that allows to modify the appearance of a markup language, in this example HTML. Hkon Wium Lie had an idea with the notion in 1994. CSS was created a specification by the W3C in December 1996, and it allows web developers to change the layout and appearance of their web pages. CSS can be used to change the font, size, and color of a specific HTML element, for example. A single CSS file can be linked to numerous pages, allowing a developer to make changes to all of the sites at once. (Hope 2019.) Figure 1 shows a simple CSS syntax using in conjunction with HTML.

```
h1 {  
    color: red;  
    font-size: 5em;  
}  
  
p {  
    color: black;  
}
```

Figure 1 A simple CSS syntax (MDN Web Docs 2022.)

### 3.3 ES6

JavaScript was introduced in 1995 as a way to add programs to web pages in the Netscape Navigator browser. The language has since been adopted by all other major graphical web browsers. It has made modern web applications possible—applications with which can be interact directly without doing a page reload for every action. JavaScript is also used in more traditional websites to provide various forms of interactivity and cleverness. (Haverbeke 2018.)

### 3.4 Bootstrap

Bootstrap is a front-end framework, which includes a collection of HTML, CSS and JavaScript tools to create websites and web applications. It was originally created by Twitter to encourage consistency across internal tools, a hosted as a free and open-source project on GitHub. (Bacinger 2022.)

#### 3.4.1 Bootstrap File Structure

There are two version of Bootstrap; one as a precompiled version using vanilla CSS for those who do not want changes and customize, and a source code version using Less CSS (or SASS) preprocessor. This version allows the developer to make deeper changes and customization as required. (Bacinger 2022.)



The Bootstrap structure includes precompiled files that can be applied quickly in any web project. Glyphicons and Bootstrap theme can also be included besides the compiled CSS and JS files. This structure can be applied into existing project without any modification. (Bacinger 2022.). Figure 2 expressed the basic Bootstrap file structure.

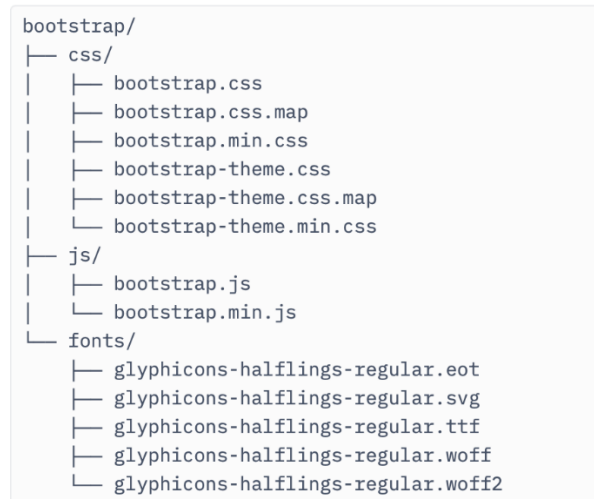


Figure 2 Basic Bootstrap file structure (Bacinger T.)

### 3.4.2 Bootstrap Grid

Bootstrap Grid is one of the important features that was introduced in versions 3; a mobile-first design philosophy, which result in a responsive design. Therefore, Bootstrap can easily scale to fit any screen sizes, from phones to tablets and desktop, from a single code base. This can be achieved by using a fluid Bootstrap Grid system, which can scale up to 12 columns depend on the screen and the viewport. Grids provide the guidelines to the layout, the horizontal and vertical, and provide an intuitive structure for the viewers to follow the layout. Before, grid-based layout was made using tables, where the content was arranged into a table cell. As CSS became more powerful, more frameworks for grid-based layout are developed, such as blueprint and 960 GS. (Bacinger 2022.)

There are rules to follow while applying Bootstrap Grid system. The column elements must placed inside the row elements, which create horizontal groups of columns. There can be multiple rows, but the columns must be immediate children of rows. In a full row, the columns width can be any combination that adds up to 12, but not necessary. (Bacinger 2022.)

Rows need to be placed either in a fixed-width layout wrapper, which has a `.container` class and a width of 1170px, or in full-width layout wrapper, which has a `.container-fluid` class, and which enables the responsive behavior in that row. There are four types of classes in the Grid system. For mobile devices is `xs` (lower than 768px), for tablets is `sm` (equal or larger then 768px), desktop is `md` (equal or larger than 992px) and larger desktop is `lg` (greater than 1200px). These basically define the sizes at which the columns will collapse or spread horizontally. The class tiers can be used in any combination to achieved the dynamic and flexible layouts. (Bacinger 2022.)

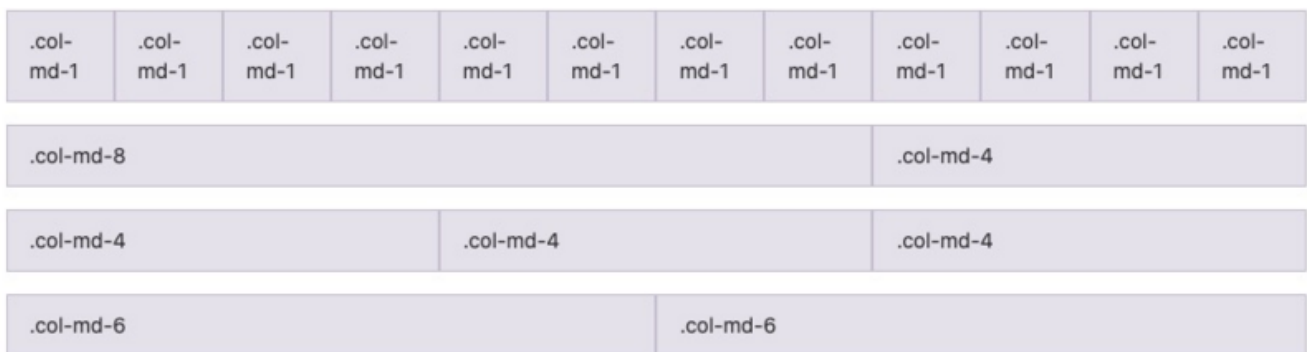


Figure 3 Bootstrap Grid layout

### 3.5 Apex Charts

Apex Charts is a charting library, used to create interactive and modern looking charts for the webpages with a simple API. It is an MIT-licensed open-source project, and it is free to use in commercial applications. React Apex Charts is an Apex Chart integration for React application. (Emad-amerho-Atori 2020.). To install and implement Apex Chart into the project, the syntax `npm install` is used, and by using the CDN links in Code 1, the graph can be implemented into the project.

```
<script src="https://cdn.jsdelivr.net/npm/apexcharts"></script>
<script src="https://cdn.jsdelivr.net/npm/react-apexcharts"></script>
```

Code 1: CDN links

### 3.6 Git and Version Control

Version Control Systems (VCS) lately have gained improvements and stability. VCS are sometimes known as SCM (Source Code Management) tools or RCS (Revision Control System). One of the most

popular VCS tools in use today is called Git. Git is a Distributed VCS, a category known as DVCS. Like many of the most popular VCS systems available today, Git is free and open source. (BitBucket 2022.)

### **3.6.1 Git**

Git is a version control system, which is a system that keeps track of changes to a file or set of files and in case of any problems, enable the developer to go back in history, comparing changes over time, and easily revert to a working state of the source code. SVN, Mercurial, and the massively popular. Git was created to make it easier for developers to collaborate. Speed, data integrity, and support for dispersed processes are among its objectives, therefore, it is popular version control systems for developers. Git is free and open source. (Iliev 2021.)

### **3.6.2 Version Control**

Version control, as known as Source Control, which have the function of tracking and managing the changes to the code. The benefit of using version control is that multiple developers can work on the project seamlessly, as the same time record all the changes into a log and each change made to the code can be traced. Version control provides a safe environment for developers to work via branching and merging. The developer can make a duplicate of the code is being work on, then it can be safely adding new function or changes to the code without interrupting the current one, which is called branching. Then with all the changes is done, it can be merged back to the source code. (BitBucket 2022.)

### **3.6.3 GitHub**

GitHub is a Git repository hosting service, but it adds many of its own features. While Git is a command line tool, GitHub provides a web-based graphical interface. It also provides access control and several collaboration features, such as a wikis and basic task management tools for every project. The main function of GitHub is fork, which copy, or clone a repository from one user's account to another. This enables taking a project that don't allow write access to modify it under your own account. If you make changes you'd like to share, you can send a notification called a "pull request" to the original

owner. That user can then, with a click of a button, merge the changes found in your repo with the original repo. (Finley 2012.)

### **3.7 Covid API**

The API used in this project is built by Kyle Redelinguys, which is a free API that provides most of the necessary info to be displayed on the dashboard. This API can also be used to build mobile apps or integrate into other application. The source of data is provided by John Hopkins CSSE.

## 4 DESIGN PHASE

The website takes shape throughout the design stage. At this stage, all visual content is created, including images, photos, and videos. The website layout is the result of the efforts of a developer. It could be a rough drawing or a completed graphic design. The layout's main purpose is to describe information structure, show content, and demonstrate fundamental functions. Colors, logos, and images are included in layouts, which might provide a rough idea of the eventual product. (Gordiyenko 2015.)

To start the design process, the vision of the web can be shown via the rough layout. The layout entails creating a basic draft, which could be graphical, to gain a sense of the website's design. The layout's objective is to give an information structure to the users, allowing them to take a visual tour of the material and basic features. Constructing a base layout for the website, the wireframe designed in the previous step is translated into buttons, tabs, menus, dashboards, color themes, typography, and graphics. (Gordiyenko 2015.)

### 4.1 Planning

This project aims to build an intuitive dashboard to let the user view and interact with the indexes provided, therefore, most important feature and figures will be presented on the main page. Figure 4 shows how a user can interact with the dashboard. The user can view the indexes of the cases, deaths, recover figure, and the graphs that intuitively show all the mentioned data. Furthermore, the user can choose which country to view, and the top 20 countries with the highest confirmed cases. Informative videos can also be viewed on the dashboard providing the user a better view of the pandemic.

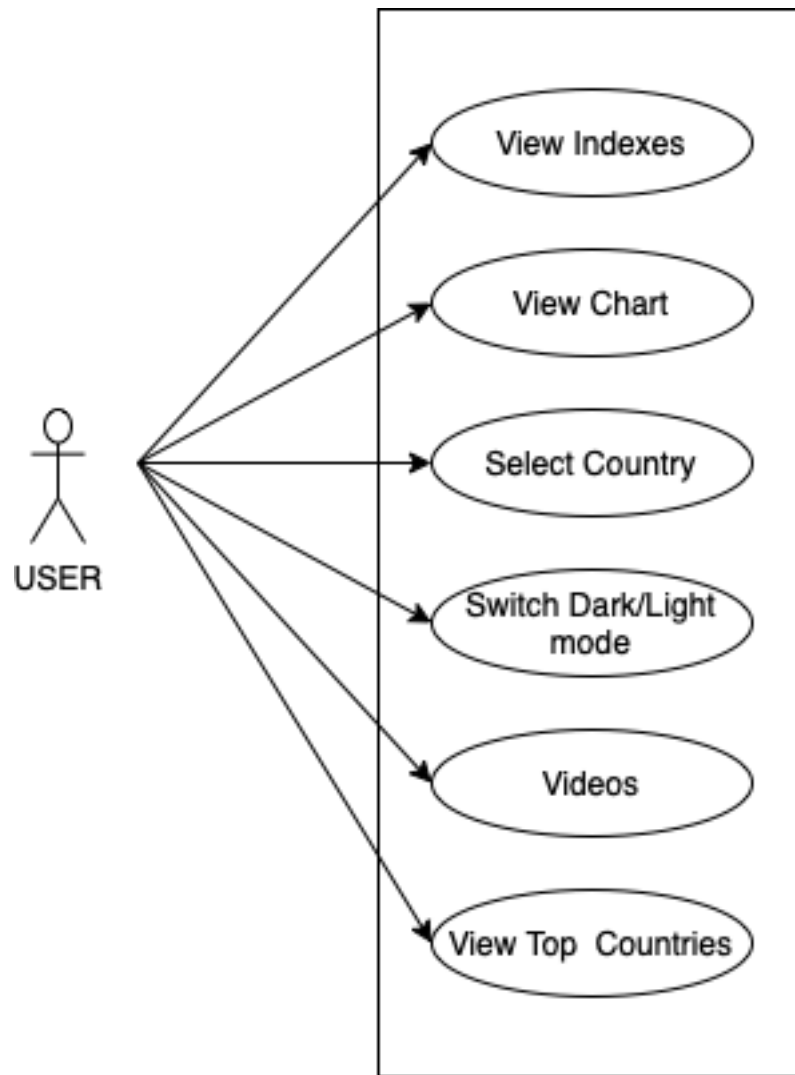


Figure 4 Use-case diagram for the user

## 4.2 Layout

The layout of the website can define its success. With a suitable design, the user can easily navigate through the website without any trouble. With the wrong design and layout, the user can be lost and confused, therefore missing the content or the information. From the start, it is important to get the layout idea right. (Boag P.2020.)

The value of the layout however, extends farther. The design must be suitable for the site's content. The content can shine with the appropriate layout, but with the inappropriate one, the information might seem packed, difficult to read, and unattractive. All websites have a grid structure that sits underlying the design. These columns and rows organize the material and help the user navigate the page. These grids can be used to construct a variety of different approaches. Every website has an underlying grid on which the layout was created by the designer. (Boag P.2020.)

### 4.2.1 Card-based Layouts

In this project, a card-based layout will be used to show the content of the dashboard. Card-based layout is very popular with E-commerce sites due to its flexibility in presenting condensed information on each element, which is easier for the user to make a choice. Card-based layouts also respond well to changes in screen size, with the number of cards in a row gradually decreasing as the available width decreases. (Boag 2020.)



Figure 5 Card-based layout (Boag 2020.)

Following the card-based layout template, the sketch for the dashboard is made to give a vision for the webpage, therefore, in figure 6 a mockup is made. The header contains the title and the switch to change between white and dark mode. In the main content section contains cards to show information about the cases, graphs and videos related to the pandemic. The footer is credited the source of the API.

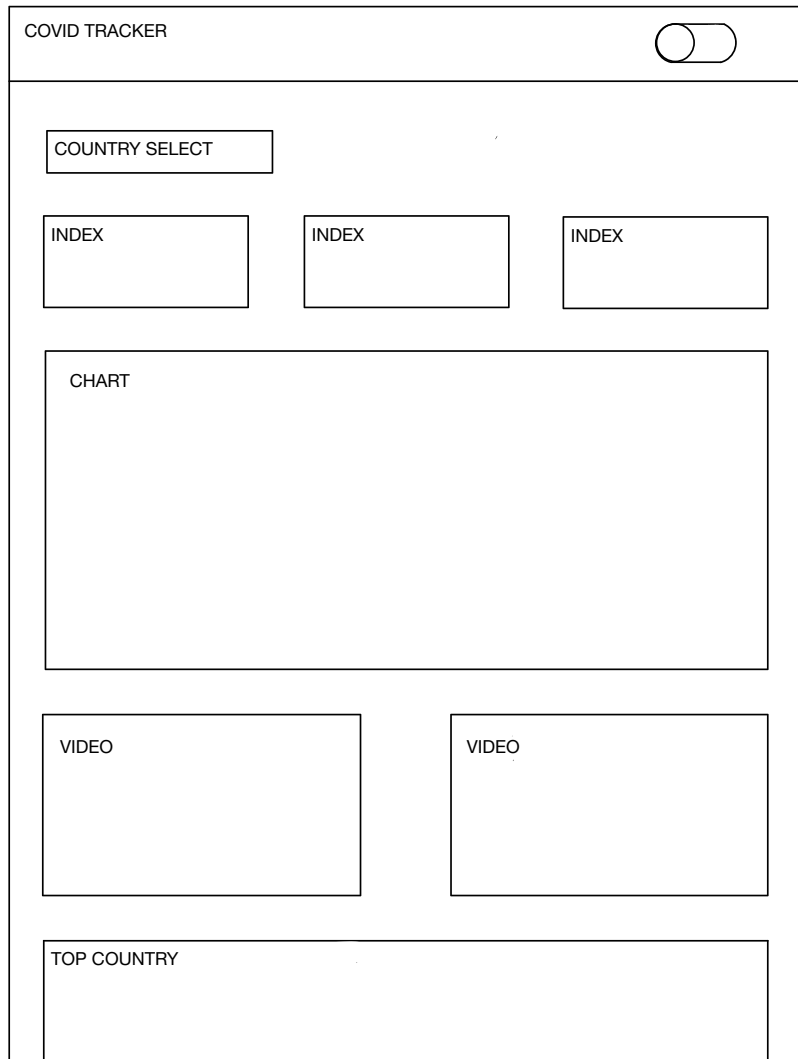


Figure 6 Dashboard Mockup



## 5 WEBSITE IMPLEMENTATION

In this chapter of the thesis, the process of developing the website is presented. Each part of the UI will be explained in detail, and the implementation of HTML, CSS and the data fetched from the Covid19 API is also mentioned in this part. This project is built using HTML, styling with CSS and JavaScript is used to implement the function.

### 5.1 Navbar

The first part of the website is the navigation bar. As in Figure 7, it consists of 2 main parts, the title of the website and a switch to change between dark and light mode. The title is also a hyperlink to go back to the main page. When the switch is clicked, it changes the state of the website from light to dark mode and vice versa.



Figure 7 Navigation bar with 2 modes

The dark and white mode is defined in the `:root` section of the CSS. The `:root` selector is used to select the root element in the web page. In this case it is the HTML part. By using this, all the page elements can inherit from it. In the JavaScript part, a function is used whenever the switch is clicked, which is shown in Figure 9, it will catch the click action and change into dark mode, which is defined in the `:root` part, as in Figure 8.

```

:root {
  --bg-body: #F6F5F5;
  --bg-content: #D3E0EA;
  --bg-hover: #1687A7;

  --color-txt: #276678;

  --nav-height: 70px;

  --shadow: 0 0 30px 0 rgb(82 63 105 / 5%);
}

.dark {
  --bg-body: #191919;
  --bg-content: #2D4263;
  --bg-hover: #C84B31;

  --color-txt: #ECDBBA;
}

```

Figure 8 Declare variables in pseudo :root

```

initTheme = () => {
  let dark_mode_switch = document.querySelector('#darkmode-switch')

  dark_mode_switch.onclick = () => {
    dark_mode_switch.classList.toggle('dark')
    body.classList.toggle('dark')
  }
}

```

Figure 9 JavaScript function

## 5.2 Main content

The main content of the website is consisting of five different parts, in which are country selector for selecting and searching the desired country, the tracking info to show indexes, graph, videos and top countries. The main content is divided into cards, each card contains different elements and data. By using card-based layout, the information can be shown intuitively and clearly.

### 5.2.1 Country Selector

The country selector allows the user to find and select the country they want to view. When a country is chosen, the whole page will refresh and show the chosen country data and graph accordingly. The Figure 10 shows the code when the user input a letter, it will sort out a list of countries that have that character in their names, and then render a list of filtered countries. This is done by taking the input from the user when a key is released, then a filtered list is made and display to the user.

```
// country filter
initContryFilter = () => {
  let input = document.querySelector('#country-select-list input')
  input.onkeyup = () => {
    let filtered = countries_list.filter(e => e.Country.toLowerCase().includes(input.value))
    renderCountrySelectList(filtered)
  }
}
```

Figure 10 JavaScript code to filter the search

### 5.2.2 Tracking counter

The counter section shows three different indexes, which are confirmed cases, recovery cases and death cases. The indexes from these cards are fetched through the Covid-19 API by using GET request. Figure 11 shows the example response.

```
{
  "Global": {
    "NewConfirmed": 100282,
    "TotalConfirmed": 1162857,
    "NewDeaths": 5658,
    "TotalDeaths": 63263,
    "NewRecovered": 15405,
    "TotalRecovered": 230845
  },
```

Figure 11 Example response

In Figure 12, by using Bootstrap, these cards can scale with the screen of the devices. For phones, it will show 3 cards on top of each other, and for tables and PC, it is in the same line. Each card is distinguished by using colors and icons so that the cards are intuitively and easily available for the user to see.

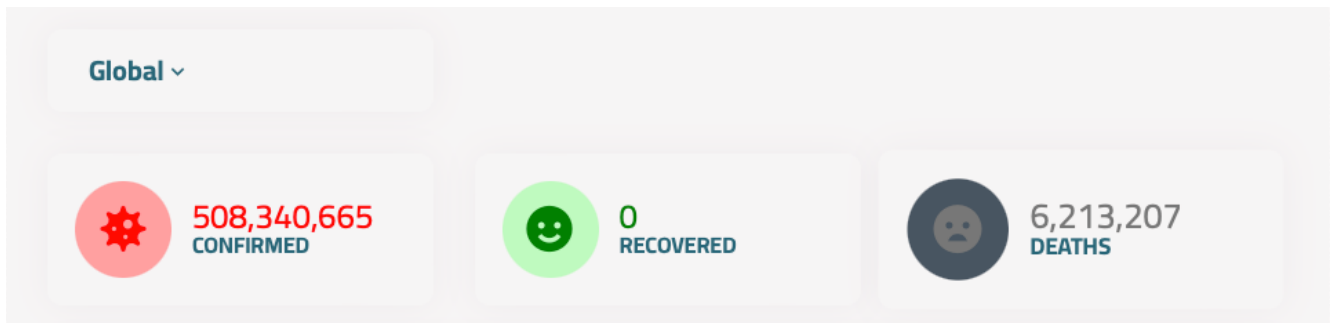


Figure 12 Displaying cards

There are 3 main element in each card, which are the icon, the index and the text. All the elements are contained in a specific `<div>`, shown in figure 13. To get the indexes into those elements, the script in figure 14 is responsible for it, by taking the data from the API and put in the `<div>`.

```

<!-- counter -->
<div class="col-12 col-md-4 col-sm-12">
  <div class="box box-hover">
    <div class="count count-confirmed">
      <div class="count-icon">
        <i class="bx bxs-virus"></i>
      </div>
      <div class="count-info">
        <h5 id="confirmed-total">123,456,789</h5>
        <span>confirmed</span>
      </div>
    </div>
  </div>
</div>
<!-- counter -->

```

Figure 13 HTML code for the counter part

```
showConfirmedTotal = (total) => {
  document.querySelector('#confirmed-total').textContent = numberWithCommas(total)
}
|
showRecoveredTotal = (total) => {
  document.querySelector('#recovered-total').textContent = numberWithCommas(total)
}

showDeathsTotal = (total) => {
  document.querySelector('#death-total').textContent = numberWithCommas(total)
}

loadSummary = async (country) => {
```

Figure 14 JavaScript code for the counter part

### 5.2.3 All time chart

To create a chart in this project, Apex Chart is used, particularly the basic line chart. A simple line chart is plotted with only a single line. A simple line chart shows the relationship between two different variables, in this case the time and the cases. A multiple line chart is a line chart that is plotted with two or more lines, for example in Figure 15, there are three lines. Each lines have a differentiate color like on the cards in Figure 12.

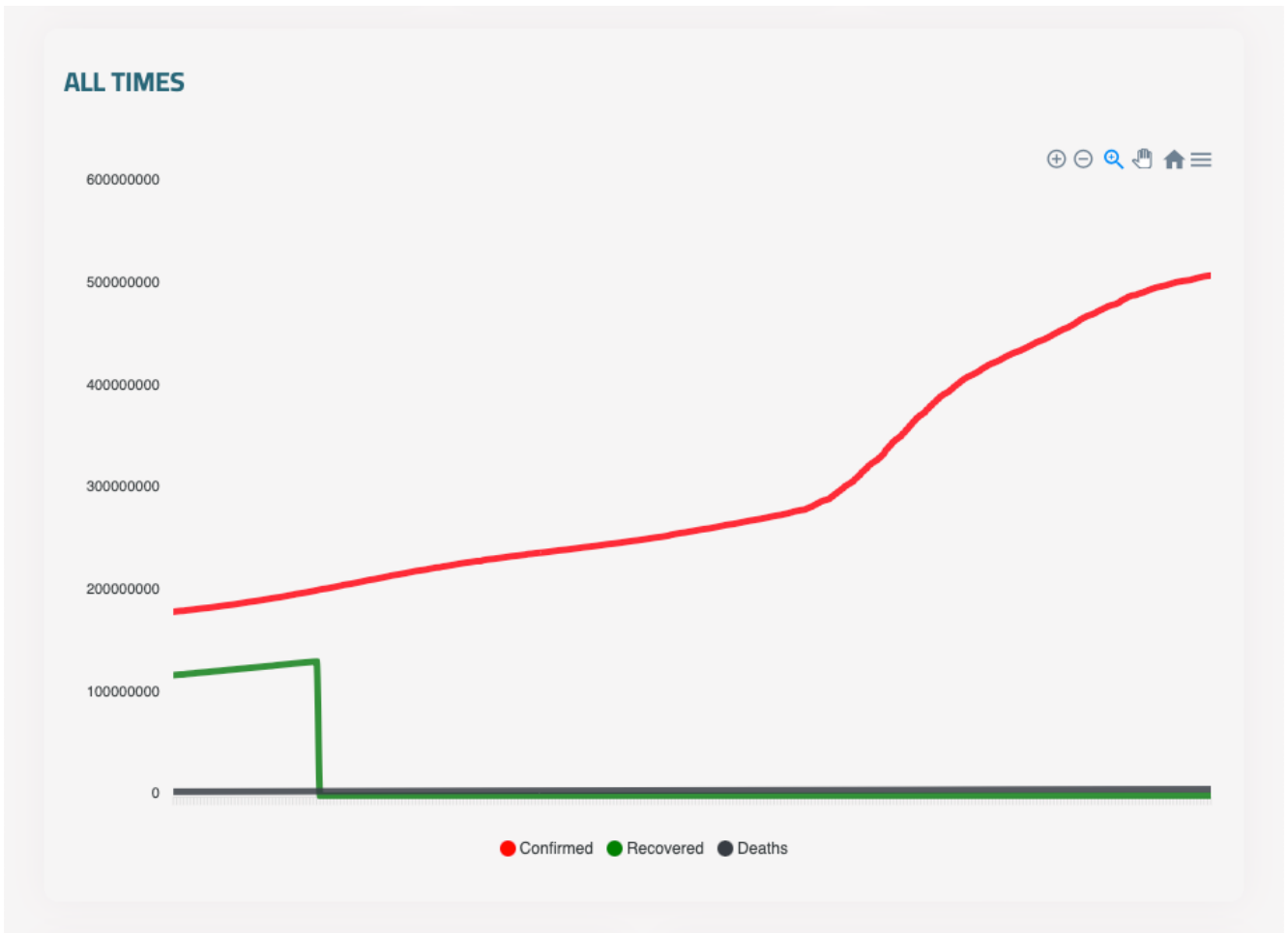


Figure 15 Line chart

Figure 16 described the way data of the chart is fetched from the Covid-19 API. A summary of new and total cases per country updating daily by using GET Summary. To return all live cases by case type for a country after a given date, a GET WORLD WIP syntax is used. These records are pulled every 10 minutes and are ungrouped. Country must be the slug from /countries or /summary. Cases must be one of: confirmed, recovered, deaths. Using GET Day one syntax, it returns all cases by case type for a country from the first recorded case. Country must be the Slug from /countries or /summary. Cases must also be one of: confirmed, recovered, deaths.

```

const covid_api_base = 'https://api.covid19api.com/'

const covidApiEndPoint = {
  summary: () => {
    return getApiPath('summary')
  },
  worldAllTimeCases: () => {
    return getApiPath('world')
  },
  countryAllTimeCases: (country, status) => {
    let end_point = `dayone/country/${country}/status/${status}`
    return getApiPath(end_point)
  }
}

getApiPath = (end_point) => {
  return covid_api_base + end_point
}

```

Figure 16 Data for the line chart

#### 5.2.4 Videos

This section is for the videos. The videos are embedded from YouTube and can be view inside the website. Two videos from World Health Organization and Kurzgesagt explain about the pandemic, the Covid-19 virus and what should be done to avoid getting infected by this virus. These videos can help the user get a general view about the pandemic.

In figure 17 and 18, the code for the video part and its implementation is shown. This part is responsive by using Bootstrap. In the tablet and PC screen size there will be two columns, therefore two videos are side by side. On phone screen size they will be on top of each other so that the videos can be viewed naturally.



Figure 17 Videos

### 5.2.5 Top Countries

Top countries table shows twenty countries that have the most confirmed cases, which is constantly update. The table is done by using descending sorting to get the highest cases to the lowest one. And then based on that order, view the table with the data on it. The figure 18 shows the code for the table, and the figure 19 shows the table itself.

```

let casesByCountries = summaryData.Countries.sort((a,b) => b.TotalConfirmed - a.TotalConfirmed)

let table_countries_body = document.querySelector('#table-countries tbody')
table_countries_body.innerHTML = ''

for (let i = 0; i<20; i++) {
  let row = `
    <tr>
      <td>${casesByCountries[i].Country}</td>
      <td>${numberWithCommas(casesByCountries[i].TotalConfirmed)}</td>
      <td>${numberWithCommas(casesByCountries[i].TotalRecovered)}</td>
      <td>${numberWithCommas(casesByCountries[i].TotalDeaths)}</td>
    </tr>
  `
  table_countries_body.innerHTML += row
}

```

Figure 18 JavaScript code for the table



TOP COUNTRIES AFFECTED			
Country	Confirmed	Recovered	Deaths
United States of America	80,984,914	0	991,254
India	43,060,086	0	522,223
Brazil	30,349,463	0	662,891
France	28,494,054	0	146,097
Germany	24,200,596	0	134,185
United Kingdom	22,106,306	0	173,985
Russian Federation	17,872,625	0	367,366
Korea (South)	16,929,564	0	22,243
Italy	16,136,057	0	162,688
Turkey	15,018,547	0	98,691
Spain	11,736,893	0	103,721
Viet Nam	10,563,502	0	43,013
Argentina	9,060,923	0	128,344
Netherlands	8,119,687	0	22,227
Japan	7,656,600	0	29,305
Iran, Islamic Republic of	7,216,040	0	140,975
Colombia	6,091,343	0	139,778

Figure 19 The top countries table

### 5.3 Footer

The footer contains the credit for the Covid-19 API, which is built by Kyle Redelinguys and the documentation is provided via Postman at: <https://documenter.getpostman.com/view/10808728/SzS8rjbc#4b88f773-be9b-484f-b521-bb58dda0315c>. Figure 21 shows the code for the footer part.

```

<!--footer-->
<div class="footer">
  Source from <a href="https://documenter.getpostman.com/view/10808728/SzS8rjbc#b07f97ba-24f4-4ebe-ad71-97fa35f3b683"
  target="_blank">api.covid19api.com</a>
  <br>
</div>

```

Figure 21 Footer HTML

## 6 CONCLUSION

With a required skillset and an appropriate plan, building a website is a practical and useful project. Step by step, a concept about for the website must decide because of the variety website ideas to think about. Then the user interface and layout need to be plan and design. Then the implementing part, writing the HTML code for each section, style them with CSS and give function by using JavaScript. When the implemented part is done, it goes to a test to check which function is doing well and which one is not, then go back a fix all of the error still remains. After that it can be published and ready to use.

In the past decade, people have been on the Internet more than ever, from social media, video games to news, it is a new way for people to have access to the information right from their palm. Therefore, having a dedicated website where user can easily access and get information to is one way not just to view the infection rate but also raising people awareness about the pandemic.

To conclude, this project has used HTML, CSS and JavaScript to build the front-end part of the dashboard. The API, which is provide by Kyle Redelinghuys, is fetched using JavaScript. The dashboard shows all the require figures, graph and videos, the dark mode switch is functional, and the overall layout is intuitive. The project came out as expected, however there are room for further improvements. With this project, the skill set for building and managing a website is improved, and it plays a huge role in the programmer career.

## REFERENCES

Pellegrino, A. 2020. *What is Product Vision?* Available: <https://blog.tryexponent.com/what-is-product-vision/> Accessed: 06.04.2022

Patel, T. May 2020. *How To Choose A Web Technology Stack?* Available: <https://www.thirdrock-techkno.com/blog/how-to-choose-a-technology-stack-for-web-application-development/> Accessed 01.04.2022

Roznovsky, A. *Choosing A Technology Stack for A Web Application Development.* Available: <https://light-it.net/blog/choosing-a-technology-stack-for-web-application-development/> Accessed 01.04.2022

Hope. 2019. *CSS.* Available: <https://www.computerhope.com/jargon/c/css.htm> Accessed 26.03.2022

Haverbeke, M. 2019. *Eloquent Javascript: A modern introduction to programming.* Third edition. San Francisco: No Starch Press.

BitBucket. *What is Version Control?* Available: <https://www.atlassian.com/git/tutorials/what-is-version-control> Accessed 26.03.2022

Finley K. 2012 *What Exaclty Is GitHub?* Available: [https://techcrunch.com/2012/07/14/what-exactly-is-github-anyway/?guccounter=1&guce\\_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce\\_referrer\\_sig=AQAAAJkAJJqpzdX8GKCNL6WDJOdArfX\\_BFMiBkXdx7jlpq6JmViiQB3Bmoe-suQnhKNs11DVA-KsOy5AjJLq6Zh78-Uq4F0oxqbBepFEffeKThMcxH-TLg0wdu-C3X5ilroX-ldH3JBbOYD9Ex-iKnJP\\_7jws8xcCuJVR1i\\_-UfcBNpJwV](https://techcrunch.com/2012/07/14/what-exactly-is-github-anyway/?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAAJkAJJqpzdX8GKCNL6WDJOdArfX_BFMiBkXdx7jlpq6JmViiQB3Bmoe-suQnhKNs11DVA-KsOy5AjJLq6Zh78-Uq4F0oxqbBepFEffeKThMcxH-TLg0wdu-C3X5ilroX-ldH3JBbOYD9Ex-iKnJP_7jws8xcCuJVR1i_-UfcBNpJwV) Accessed: 26.03.2022

Iliev, B. 2021. *Introduction to Git and Github.* First Edition. DigitaOcean. New York City, New York, USA

Glass, E. 2020. *How to build a website with HTML*. First Edition. DigitaOcean. New York City, New York, USA

Bacinger T. 2022 *What is Bootstrap? A Short Bootstrap Tutorial on the What, Why, and How?* Available at: <https://www.toptal.com/front-end/what-is-bootstrap-a-short-tutorial-on-the-what-why-and-how> Accessed: 10.04.2022

Emadamerho-Atori N. 2020. *Data Visualization with Apex Charts*. Available: <https://www.smashing-magazine.com/2020/11/data-visualization-apexcharts/> Accessed: 10.04.2022

Gordiyenko S. December 2015. *Website Development Process*. Available: <https://xbsoftware.com/blog/website-development-process-full-guide/> Accessed: 10.04.2022

Boag P. Nov 2020. *Website Layouts*. Available: <https://websitesetup.org/website-layouts/> Accessed: 19.04.2022

