VIETNAM NATIONAL UNIVERSITY – HO CHI MINH CITY
INTERNATIONAL UNIVERSITY
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DATA VISUALIZATION FOR COVID-19

By

Ngo Quang Trung

A thesis submitted to the School of Computer Science and Engineering in partial fulfillment of the requirements for the degree of Bachelor of Computer Science

Ho Chi Minh City, Vietnam
Year 2022-2023

DATA VISUALIZATION FOR COVID-19

APPROVED BY:

Le Hai Duong

THESIS COMMITTEE

ACKNOWLEDGMENTS

First and foremost, I would like to thank Dr. Nguyen Van Sinh, the Dean of Computer Science Department, who gave me an opportunity to practice and learn in the international environment of Technology.

Second, I express my thanks to my friends, who helped and suggested to me how to think and opened my mind about the idea of data visualization with recommendations in reality.

And most importantly, I sincerely express my regards and respect to Dr. Le Hai Duong who sided with me for 4 years at International University and gave me opportunities to learn modern web applications such as MEVN stack to participate in a real project. Moreover, he is always encouraging me to explore my full potential to complete the work in the most efficient way during my study period.

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Definition
API	Application Programming Interface
CSS	Cascading Style Sheets
DOM	Document Object Model
ESS	Explained Sum of Squares
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IT	Information Technology
IP	Internet Protocol
IU	International University
JS	JavaScript
JSON	JavaScript Object Notation
REST	Representational State Transfer
RSS	Residual Sum of Squares
TSS	Total Sum of Squares
SVG	Scalable Vector Graphics
UI	User Interface
URL	Uniform Resource Locator
WCSS	Within Cluster Sums of Square

ABSTRACT

Many users, especially the elderly, as well as those who are not too tech-savvy, are currently facing problems when managing raw and complex data. It is difficult and confusing to deal with and deal with large amounts of data on disease cases of individual countries and territories over time. To solve this, presenting graphical and interactive user interfaces is essential for users who want to understand those data. By applying the data visualization method, an attractive application will impress more users with visual aids; faculty or departments can keep tracking more effectively of cases of infection globally and specific to each province in each country to make judgments about the complicated situation of the epidemic in a territory, country, or locality quickly and in the most general way, thus taking appropriate prevention measures and raising awareness and spreading the virus. Their understanding of everyone gives them the most general overview of the epidemic situation, creating a solid foundation for self-awareness of epidemic prevention. Besides, with the special epidemic situation, the pandemics develop abnormally with many mutant variants that surpass the antibodies of the vaccine, so there is a risk of new pandemic outbreaks at any time. Therefore, for medical students and researchers, it is necessary to store visual data and predict the future epidemic situation and the epidemic situation of groups of countries with similar epidemics. Through the application of popular machine learning algorithms to solve the above problems, the application of covid data visualization will help the above target users solve the need of observing and predicting the epidemic seemingly situation.

CHAPTER 1: INTRODUCTION

1.1 Background

While history, the pandemic [1] is a great danger to not only the economy but also affects human life such as cholera, plague, Covid-19. Until now, thousands of new deaths are recorded every day in Japan and South Korea and show no signs of abating. China abolished the zero covid policy, causing this country to have a dizzying increase in the number of infections along with the policy of opening its customs gate when the country appeared a new variant BF.7. Along with that development, right in Vietnam this year 2023, just recorded infections with the XXB variant that is considered more dangerous than the previous variants. The global pandemic of Covid-19 with many new strains that are resistant to vaccines, when Covid-19 is still not considered an endemic disease as well as thousands of new infections and deaths globally continue to be recorded. Facing big data information about the epidemic situation is one of the big problems of users. One of the most effective ways is to raise awareness of each person, update the specific case number so that they know, and limit travel to reduce the risk of a high outbreak. Existing Covid-19 applications have some shortcomings compared to the needs of users. Users want to search and filter out information they need to know about the specific epidemic situation not only in the country but also in other parts of the world through newspapers or traditional search sites. In addition, these pages lack charts that show enough information for users to observe and compare visually, as well as do not have a smart disease forecast feature for users to easily arrange travel schedules. Not only during the Covid-19 pandemic, with the use of Vue framework as a component and organization of APIs in the form of import and export, the application of this thesis will also be well reused for the upcoming pandemic by replacing the suitable APIs of that pandemic.

1.2 Problem Statement

Facing inadequacies and dangers from the pandemic in general and Covid-19. Then an application that apparently displays the data is a necessity and above all, the application can be flexibly reused for new pandemics in the future, when new variants of covid are constantly mutating and are resistant to vaccines.

As a user who is interested in health and a healthy lifestyle, having to follow through the press and media channels is a big disadvantage because it is possible to miss the developments of the epidemic at any time. The issue of the accuracy of the data source is also something that users care about.

Besides, not only epidemic researchers or medical students, but families with children also studying abroad, or those who like to travel are still very interested in the application that can predict the situation of diseases. Future infections as well as schedules in a particular country so that they can easily arrange they're to those countries or conduct research on them.

In addition, when the epidemic is abnormal with many mutated virus strains, it is useful to show countries with similar epidemic statuses based on the number of infections and deaths. Spread of these strains.

As worried users who are in a complicated situation with the epidemic, they want to know the current situation of the Covid-19 epidemic. For example, the number of new cases, the number of deaths, is the number of recoveries in the country they live in or more specifically, in the province or city in that country. Moreover, they want to see a visual map or globe of the world when they want to know which country is holding the leading position in the number of most serious cases compared to the rest. This helps them to have a more general view of developments around the world to raise awareness of themselves to propagate to relatives and society.

1.3 Scope and Objectives

The thesis will investigate the processes of visualizing data and the demo web application when applying this technique. Also, this thesis hopes to achieve the following main targets:

- Provide a graphical country dashboard with a good user experience.
- Provide a graphical worldwide dashboard with a worldwide map
- Allow filtering on visual charts (Change the date range, Change case type,)
- Allow users to search for a country to change the graphical country dashboard (Filtering, sorting on search)
- Allow user to manipulate on prediction feature (covid cases, deaths)
- Detect user IP browser for the first loading country dashboard
- Recommend a list of countries with the same epidemic situation to the country that users are viewing.
- Produce the Vaccine efficiency section of each country.
- Produce countries comparison in its continent section.
- Deploy Web-App for engaging many users.

1.4 Structure of thesis

The thesis will be divided into 6 chapters:

- **Chapter 1:** This section will introduce the problems faced by all users in the face of big and complex data and inadequacies of current covid applications, in the context of the extremely complicated evolution of pandemics in different countries with many new and more dangerous strains.

- **Chapter 2:** This section will analyze and introduce the list of libraries and frameworks to be used.

- **Chapter 3:** This part will cover user requirements analysis, System design, data structures with user interface design, basic data visualization steps, and an introduction to machine learning algorithms applied in this thesis

- **Chapter 4:** This section will cover the implementation of the application, the file structure in the application and the selection of the appropriate chart types and the resulting output and the appropriate features.

- **Chapter 5:** It will discuss the possible situations when using the Covid-19 data visualization application and the reusability of this application for future new pandemics, as well as SWOT analysis.

- **Chapter 6:** This last section will show a list of future works and conclusion for the Covid-19 data visualization web app.

CHAPTER 2: LITERATURE REVIEW / RELATED WORK

Before going to explain how the implementation works, this chapter will introduce a list of core libraries and frameworks to be used in the project and explain the mechanisms behind them.

In data visualization, UI is a must for every user so that they can see the data and interact with it. The software may be developed for desktop, mobile, or web applications that are suitable for specific cases. This thesis will apply web applications and mainly focuses on Vue 3, Highchart.Js with many other libraries in order to support these works and make a comparison between them.

2.1 Vue Framework

Vue [2] is a JavaScript framework made by Evan You after working for Google using AngularJS in a number of projects. He later summed up his thought process: "I figured, what if I could just extract the part that I really liked about Angular and build something really lightweight." The first source code commit to the project was dated July 2013, and Vue was first released the following February 2014 . Vue is used to build user interfaces with the development of single-page applications. It is also providing the code to open source so that can bring a friendly ecosystem to developers, and they can create additional libraries to support such as Vue Router and Vuetify. There are a few main concepts that Vue provides to the developers:

- **Element:** The building block of the Vue application, renders and displays the content on the browser.

- **Component:** A JavaScript class or function that returns a DOM element via render function. Also, it can return other components (nested components)
- **Props:** Basically, they are the data inputs of a component, props are similar to HTML attributes, and they cannot be changed in its component. Props can be used by calling the attributes from their parent component.
- **State:** In the Vue component, a state is an object that represents the parts of the app that can be changed when the content is affected by user interactions.
- **Life cycle of a component** contains four phases: creating, mounting, updating, and destroying [3]. They are important that can affect the user interface, user experience, and user interaction when using the app. For more information, please take a look at Figure 1.

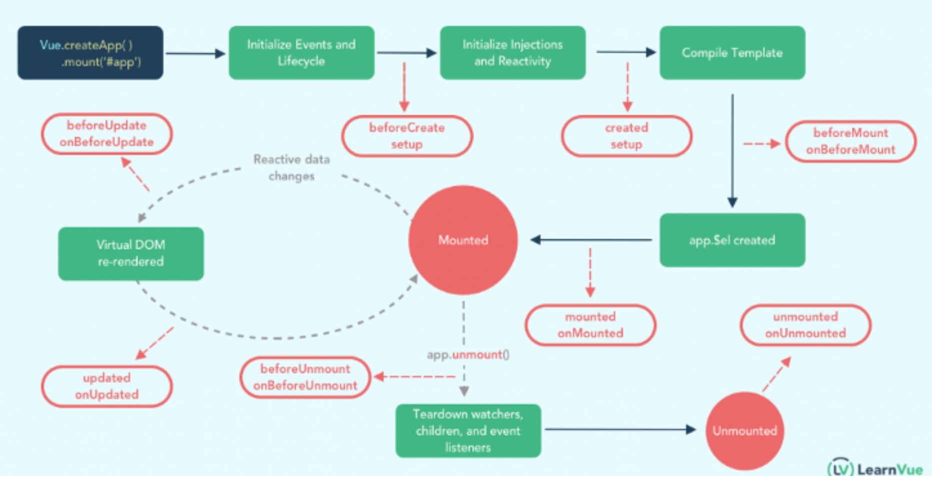


Figure 1: Life cycle methods of a Vue component

In data visualization, Vue is a great tool not only can handle state when dealing with complex data but also help developers to style the component logically like CSS.

2.1.1 Vuex

Vuex is a state management [4] pattern When there is a shared state across multiple components, especially components that are siblings or in very different views, managing delivery and updating of the state is a challenge. Vuex allows the management of shared states between components in a structured and maintainable way, by creating a global state tree that can be accessed and updated by every component in a structured way. It serves as a centralized store for all components in an application, with rules proving that the state can only be mutated predictably and that all components can access this central data store without having to pass from parent component to child component as in the traditional way. When a web page has many components and adds complexity, you need to organize how state changes are made. This is an ideal situation where Vuex would support the project well. (see Figure 2).

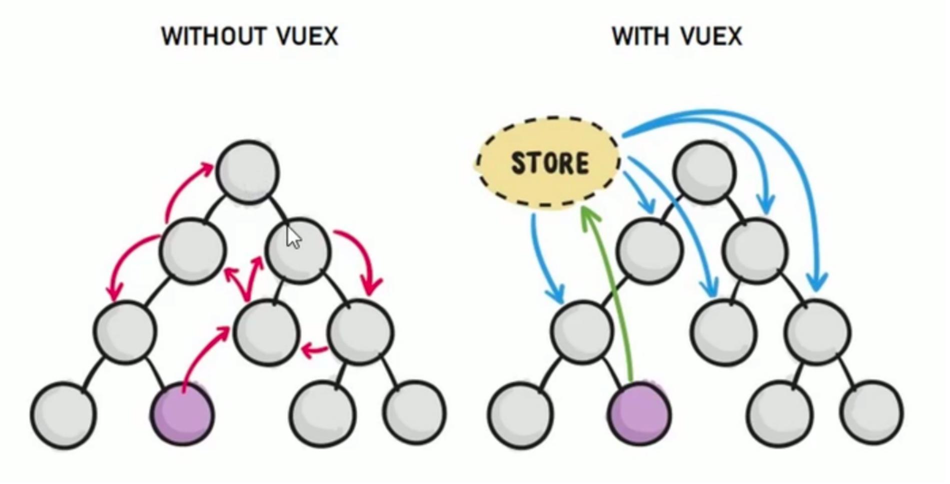


Figure 2: State management comparison with Vue (left) and Vuex (right)

In order to understand the core concept of this library when combined with Vue, there are four keywords when updating a state: Actions, Mutations, Getters, and State (see Figure 3).

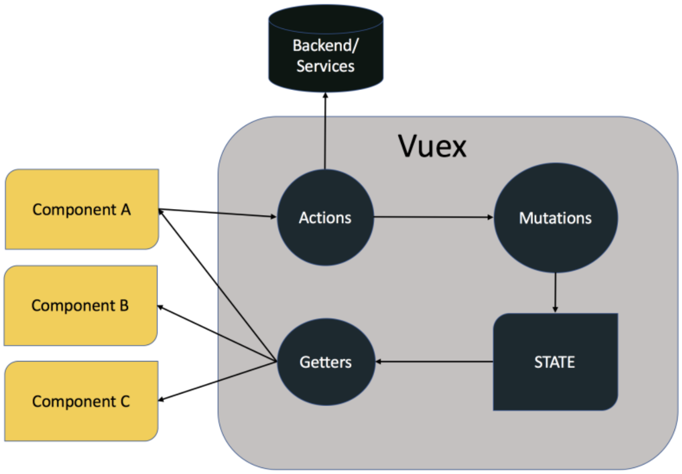


Figure 3: Mechanic of state management in Vuex

First, when a user triggers an event such as clicking or hovering, an “Action” will be dispatched into “Mutations”. Next, “Mutations” will handle and update the store (or global states), and finally, the store will update which means the “States” values in that store are handled, therefore they can be used for rendering directly or through the “Getters” ". The component will receive new props to re-render.

Second, both “States” and “Getters” can be used for rendering, imagine that “States” stores the raw data values but in some cases, we need the raw ones to be filtered before rendering, therefore they have “Getters” in Vuex.

Finally, the “States” can only be updated by “Mutations” but why do we need “Actions” to handle the events when “Mutations” can do that by itself? Because “Mutations” processes data synchronously, while APIs need asynchronous methods to handle requests from the client side to the server side quickly. So we need to call requests from "Actions" on "Mutations" to process them asynchronously.

2.2 Highchart (Highchart.js)

Highcharts [5] is an SVG-based pure JavaScript charting library first released in 2009 that is suitable for developers to create responsive, interactive, and accessible charts. The library includes all standard chart types and more. Highcharts library comes with all the tools to create reliable and secure data visualizations. Built on JavaScript and TypeScript, all our charting libraries work with any back-end database or server stack. We offer wrappers for the most popular programming languages (.Net, PHP, Python, R, Java) as well as iOS and Android, and frameworks like Angular, Vue, React, and more. Simple options structure allows for deep customization and styling that can be done via JavaScript or CSS

CHAPTER 3: METHODOLOGY

3.1 User requirement analysis

As mentioned in section 1.2 problem statement in chapter 1, users want an application that displays statistical data all information about the number of daily and overall cases on a single page with a comfortable interface, easy to use so that everyone, including children or the elderly, can grasp official content from reliable sources about the epidemic situation quickly and simply. Instead of having to keep track of numbers long and not concisely.

To make the application attractive with many interesting features to reach more users such as people who love to travel, international students, or medical students. Machine learning application features such as predicting the number of new infections or deaths in countries on any date in the future or a system that suggests countries with similar epidemic status. Besides, the comparison between countries around the world or countries in each continent by categories like the number of cases per day or the total number of cases so far and much more should also be developed in the app.

To represent this information apparently, the project will use many different types of charts suitable for different purposes. Besides, depending on the responsive application that deploys on multiple platforms compatible with different operating system browsers or both desktop and mobile platforms.

The project will have 2 pages, including an overview of the global epidemic situation and a specific epidemic overview in each country.

3.1.1 The global epidemic situation requires

In this section, the user wants to have an overview of the disease worldwide. There can be a map divided by color level from light to too dark to easily identify which countries are the hardest hit by the Covid-19 pandemic. Displays an overview according to the user's navigation into each given country of the total number of cases at the time the user is looking, as well as the deaths and recovered cases. At the same time, it will show the most prominent countries in the world in terms of the complexity of the epidemic so far. The following tasks below need to be achieved:

- Each country will have its country name and represent how many cases have been confirmed, the total number of deaths, and completely recovered when the user hovers over that specific country in the form of a world map. Users can see the Covid-19 cases distribution of in every country in the world through a world map divided by color level from light to dark, the darker, the more seriously affected in that country.
- Users want to know immediately the total number of cases so far and the number of cases today about the disease's evolution, as well as the trend chart of the last 7 days
- Users, want to know the disease's evolution according to the time chart of the daily epidemic parameters
- Need a time chart of various kinds of daily epidemic parameters in terms of percentage of infections, deaths, and recoveries for users to track the change and severity of the epidemic
- request a time map according to the total number of cases accumulated over the days so that users can see the large fluctuations of the epidemic

- Comparing continents together as well as comparing countries in that continent in an interactive way is also what users want.

3.1.2 Specific disease situation in each country requires

When it comes to the situation of the Covid-19 epidemic in a specific country, in addition to displaying in the form of a graph the required information about the total number of infections, the total number of recoveries, active cases and deaths in that country. It will show recent reports of new cases and new deaths in that chosen country. The requirement must be qualified by the following goals:

- Require suitable charts to display key data in that country from the time the outbreak appeared to the current time the user is accessing.
- Users want to get reports from charts of their choice
- Provide suggestions for countries with similar translation status for users' convenience.
- The function of predicting the number of new infections after a period of time compared to the current time will be what users expect in this application.
- UI must be simple, clear, and friendly to users, and make users interact in a short amount of time
- The performance of loading and processing data must be fast and optimized when users manipulate.
- Filter and search for other countries for an easier look, as country sorters by different categories are available.

- Assessing the stability and effectiveness of vaccines in a country based on the number of infections and deaths is what users expect for an application in the period after a strong outbreak of the epidemic.

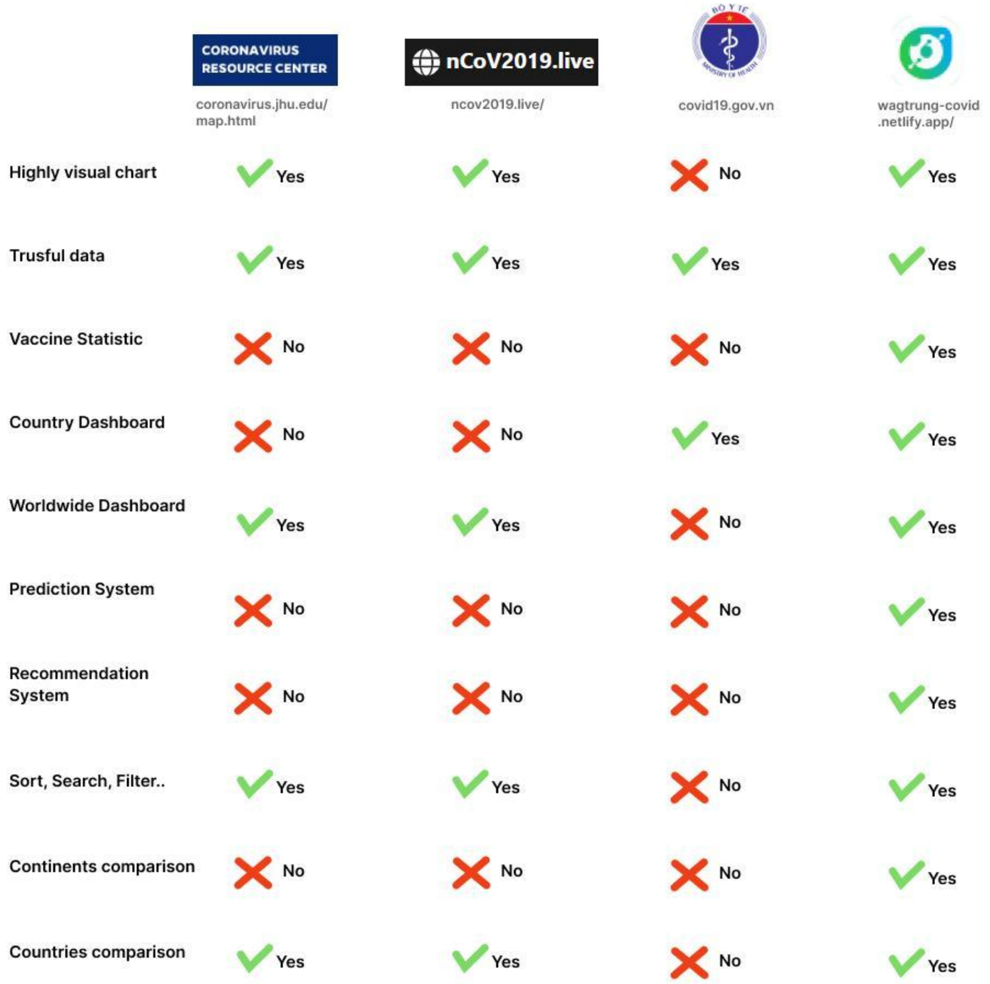
3.1.3 Analysis of current applications

Before starting to develop this application, the market research and evaluation is indispensable.

The difference and highlights of the application in this thesis are:

- The application of machine learning algorithms to make predictions and the country recommendation system.
- Official data source, updated daily. Add to that the synergies and coverage of worldwide and country-specific content.
- In addition, sections such as evaluating the effectiveness of vaccines and comparing countries in the same continent are also noteworthy.
- In addition, code reuse for upcoming pandemics is also calculated by managing and organizing files and API suitable for application reuse.
- Finally, it is an eye-catching, modern interface design that is easy for users to interact with charts and full of information for a high user experience.

There are already a few apps like ncov2019 , coronavirusJHU , BoYTe , ect. But it is not really complete and provides a good experience for users for their search and disease needs in both the present and the coming future (see Figure 4).



	CORONAVIRUS RESOURCE CENTER coronavirus.jhu.edu/map.html	nCoV2019.live ncov2019.live/	covid19.gov.vn	wagtrung-covid.netlify.app/
Highly visual chart	✓ Yes	✓ Yes	✗ No	✓ Yes
Trusful data	✓ Yes	✓ Yes	✓ Yes	✓ Yes
Vaccine Statistic	✗ No	✗ No	✗ No	✓ Yes
Country Dashboard	✗ No	✗ No	✓ Yes	✓ Yes
Worldwide Dashboard	✓ Yes	✓ Yes	✗ No	✓ Yes
Prediction System	✗ No	✗ No	✗ No	✓ Yes
Recommendation System	✗ No	✗ No	✗ No	✓ Yes
Sort, Search, Filter..	✓ Yes	✓ Yes	✗ No	✓ Yes
Continents comparison	✗ No	✗ No	✗ No	✓ Yes
Countries comparison	✓ Yes	✓ Yes	✗ No	✓ Yes

Figure 4: Competitor analysis

3.2 System design

After receiving the requirements from users, it is necessary to design the system architecture. Having analyzed and implemented the use case diagrams, the user flows are shown in Figures 5 and 6 below, respectively.

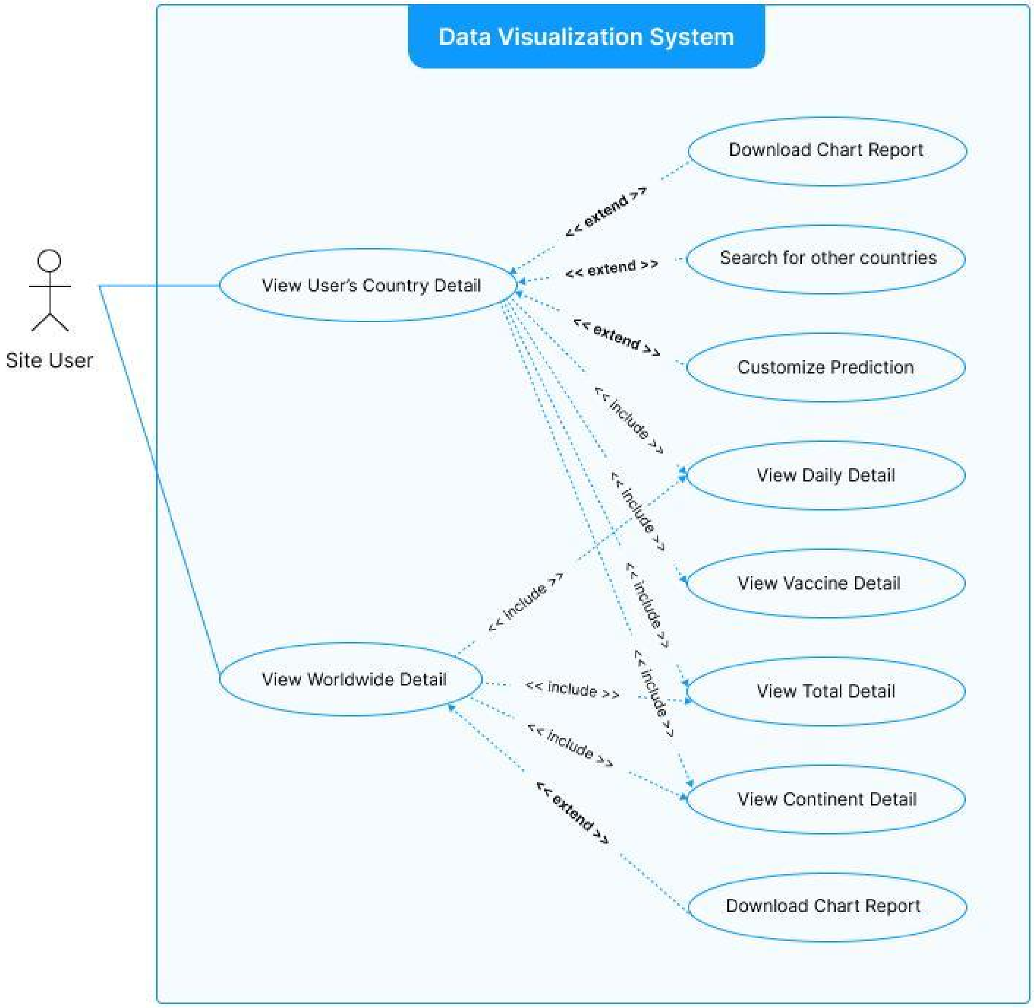


Figure 5: Use case Diagram for visualization application

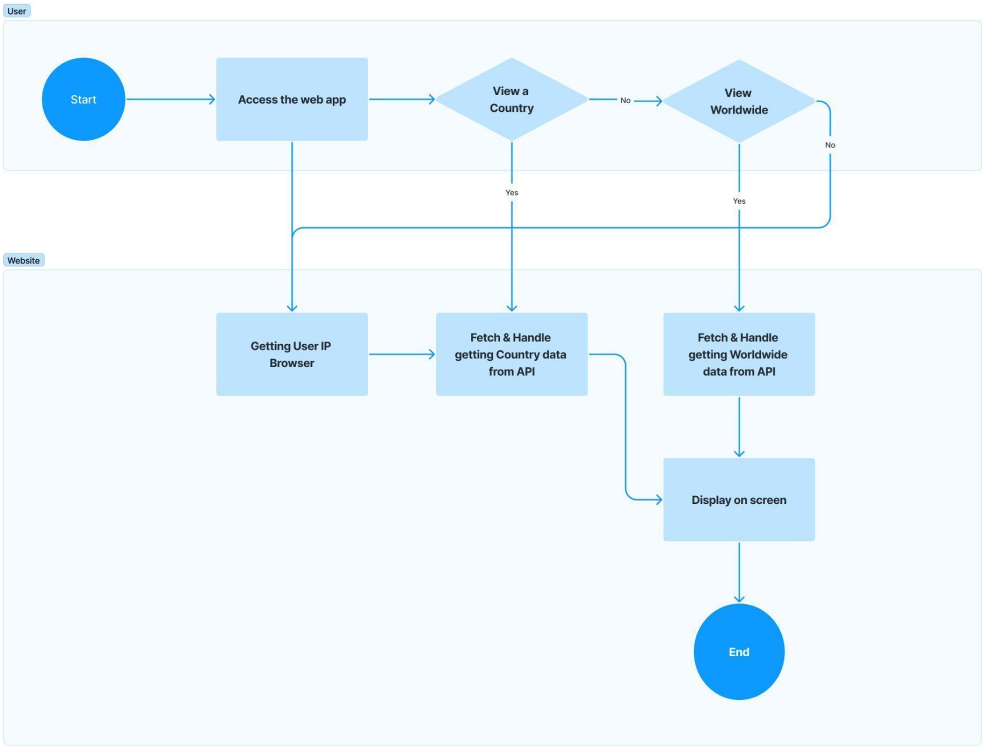


Figure 6: User Flow Diagram for visualization application

3.3 User Interface design

To match the requirements, and design system above. This thesis will use Figma tool to sketch the draft versions before going to implement. Note that the draft versions will not represent UI exactly as the final build.

In Covid-19 worldwide overview UI design, the layout is split into 3 parts:

- **Top side** : includes the most prominent parameters that users are interested in, including: total number of infections so far, number of deaths so far, number of recoveries so far. The trend of the epidemic in the last 7 days. Finally, the number of new cases for today.

- **Main side:** a large map in chroma that allows users to interact intuitively, change the disease parameters they want to see
- **Bottom side:** the information sections include a graph of the number of active cases by day, the number of cases is calculated incrementally. Comparison between countries and continents

Covid-19 National Overview is the same as Global Overview with some new features. The layout will be divided into three parts.

- **Top side:** will also include the top prominent epidemic metrics that users want to see. In which the general data so far and today's data are noteworthy data. In addition, the search for countries should also be placed in the right place to make it easier to manipulate the selection on a list of countries that can be sorted by different categories.
- **Main side:** A suitable design part for the list of countries with similar epidemic status suggested from the system. Highly interactive display for the predictive function of the number of infections.
- **Bottom side:** will include charts of epidemic developments by sections: daily section, total timeline section, vaccine section and comparison between countries in the same continent.

There doesn't seem to be much change in the two main Layouts, the overview in the world and the overview in specific countries. Country data has the ability to predict upcoming disease developments and provide suggestions for countries with similar epidemic status. Therefore, users only need to be interested in a single application and can look up and refer to most of the disease parameters in a general and logical arrangement of related subsections, without the need for additional information to have to refer to too many websites to aggregate and compare data across countries.

3.4 State design management

To let the web renders data like UI design, there is a state design phase in development will be required before implementation when scaling the application. In this design, it will include main state worldwide and country.

The data retrieved from the API (blue boxes in figure 7and Figure 8) includes:

- **" AllCountries** ": overview of all countries is stored in an array.
- **" UserCountry "**: The parameter is obtained and determined from the public IP of the user.
- **" HistoryCountry "**: The total number of cases or deaths or recoveries accumulated by the portal day by day, recorded as a chronological array
- **"VaccineCountry"** : The total number of vaccines administered by that country cumulatively over each day is recorded as a chronological array.
- **"ContinentTotal"** : Statistics for the epidemic overview of the continent that the user is viewing.

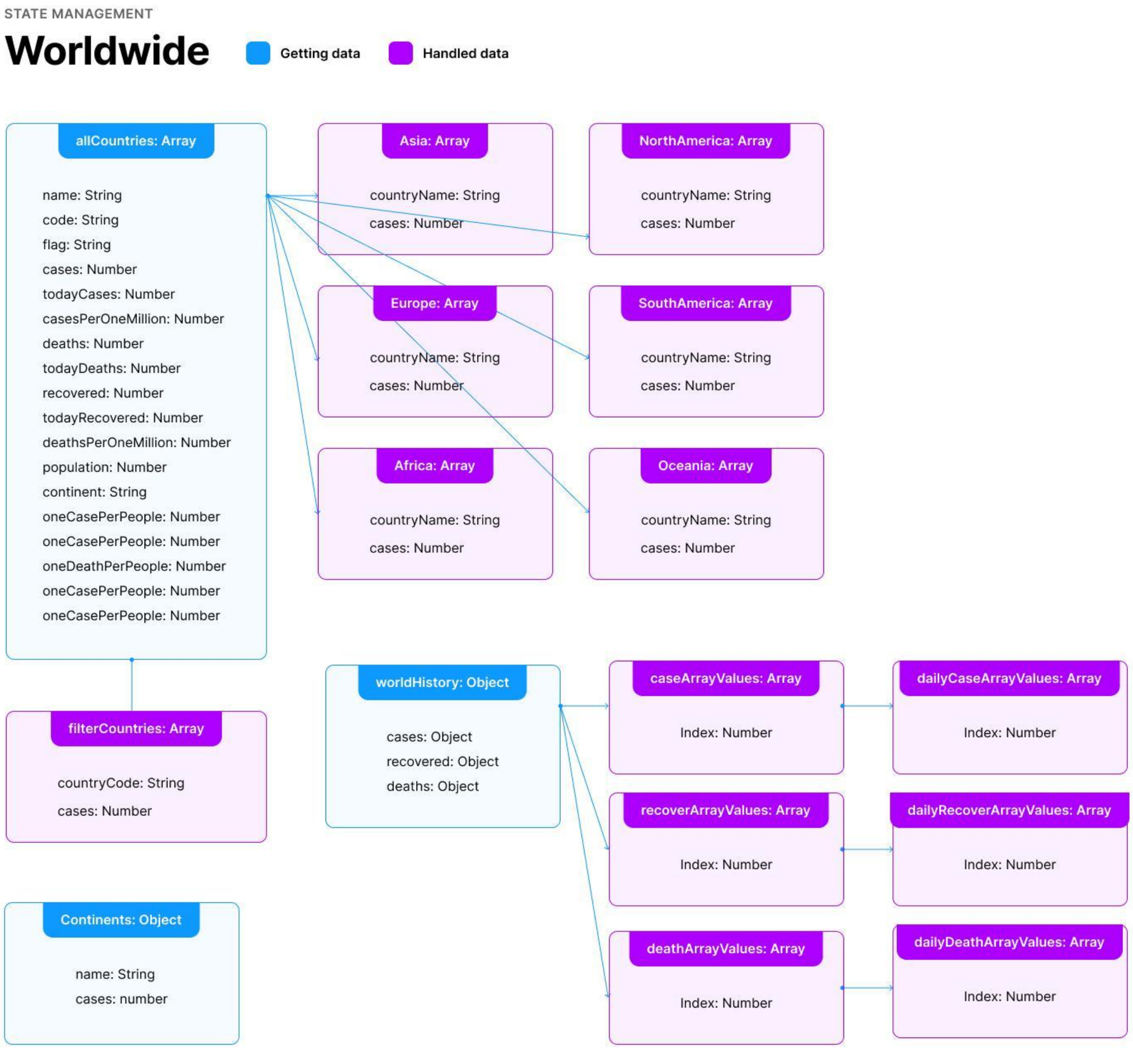


Figure 7: State design in Covid-19 Worldwide Overview

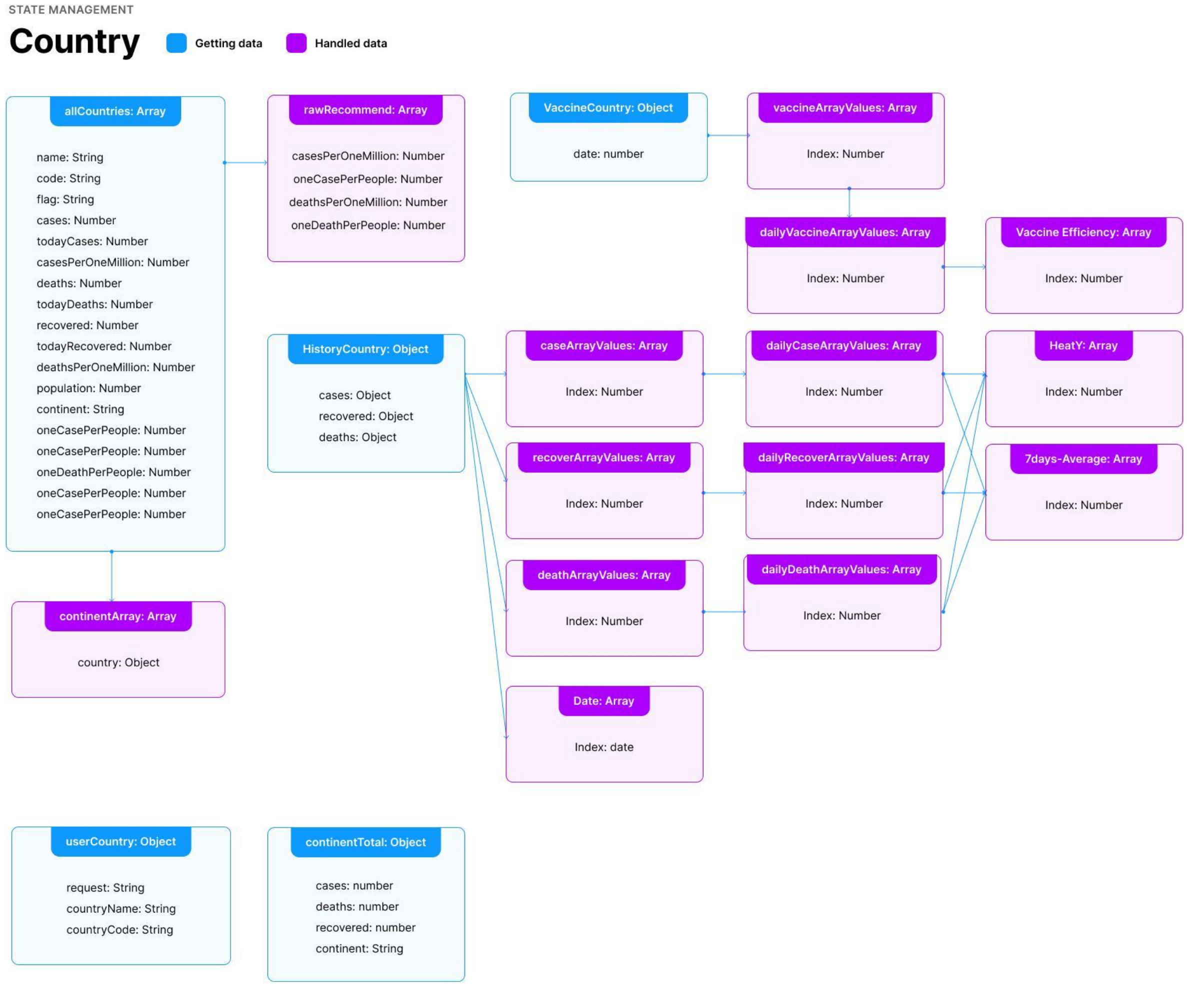


Figure 8: State design in Covid-19 National Overview

Remaining, all other data used to develop this application has to be processed and manipulated into data suitable for different types of requirements and graphs. The processing of data obtained from the API will be carried out in Chapter 4 of this thesis.

3.5 API management and fetching method in REST API

Besides, When requesting data to backend server via REST API, there are 4 phases of state (see Figure 9): start request, pending request, request success and request failed. If there is any error when fetching, client will get an error message and written in state. This is necessary to prevent errors and also improve UX when a server is down.

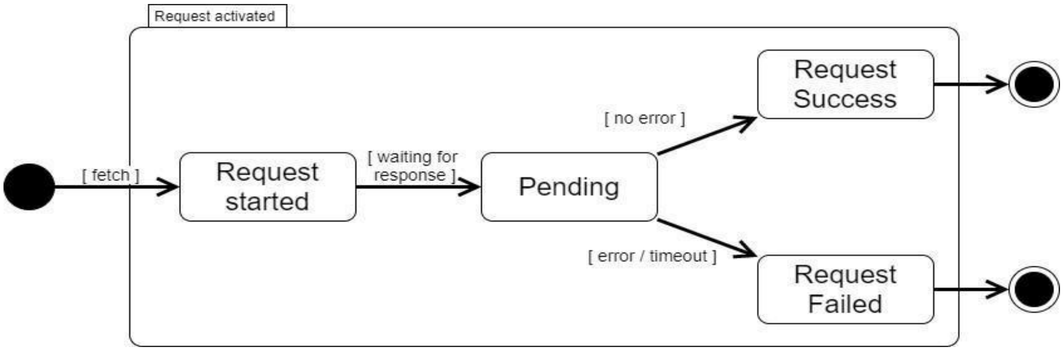


Figure 9: Request state when fetching

3.5.1 Get regular data with fetch or Axios

We use the Get method of Rest API to get data from the provided API source. If the operation is successful, we get back a Promise in the form of HTTP Requests from which we can access that data in JSON form using the response.json() method. Moreover, when using the Axios library with Get instead of using the built-in fetch in JavaScript, it will eliminate the step of converting from HTTP Requests to JSON format.

3.5.2 Asynchronous processing while retrieving data

Since the data is returned as a promise, there is a problem of asynchronous execution in JavaScript [6]. So there will be two common methods used for asynchronous processing: using then() method or using async and await.

When you need the code to be executed in sequence when working with promises, there are two ways depending on the intended use, including:

- **In turn:** promises will be executed in sequence, fetch requests written first will be executed first, and requests must wait until the previous one completes.
- **Parallelism:** all requests will be called and executed at the same time, minimizing user waiting time.

In this application, use both methods mentioned above. The fetch method will be called for the first time in turn to get the overview data of all countries. Then from getting the country code from the user request or automatically from the user's IP browser. The application will conduct a parallel fetch of in-depth data specific to the selected country, such as data on the cumulative total of daily cases, data on the overview of the continent to which the country belongs, and data on the total number of cases. the progressive vaccine that the country has used. Accordingly, we avoid asynchronous and also reduce the waiting time of the user.

3.6 Methods to update data in Vue

In Vue lifecycle, in addition to using mounted and created for interacting with the API by calling the function before rendering the elements to the interface. In addition, updating the data when there is any interaction from the user that changes the value of the data variables is also very important. Specifically, here is changing the country that the user wants to see, it will require a new fetch of in-depth data about the selected country. If used with lifecycle Updated() after each change in the value of any variable, the function will be called over and over again and waste and increase the user's waiting time. The solution of this application is to work with watch() in Vue, the watch method will only observe the change in the specified variables and call the function to interact with the API to avoid the same function being called multiple times waste of time.

Besides, the Vue lifecycle hook in the interaction between parent components and child components is also something to keep in mind when developing this application when async occurs.

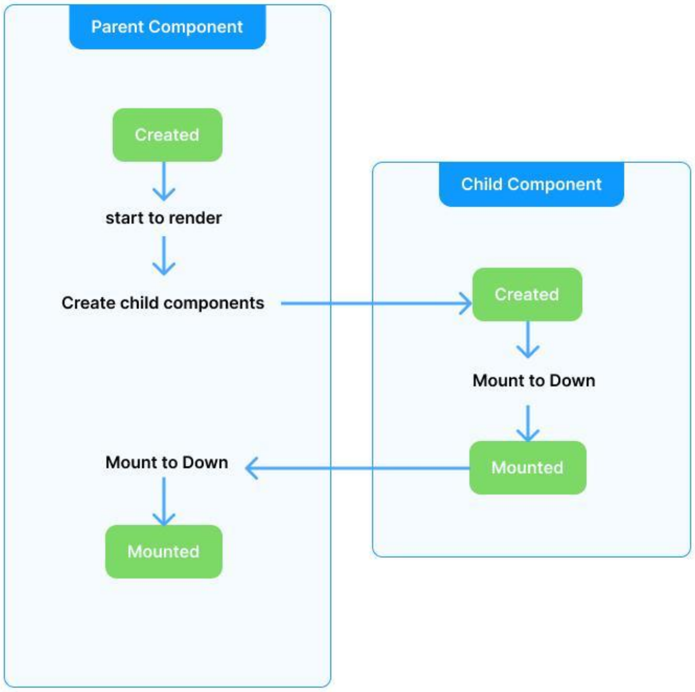


Figure 10: Vue parent and child lifecycle hooks

3.7 Regression algorithm for prediction feature

It belongs to the group of supervised learning in machine learning [7] Supervised learning is a method of predicting outcomes based on known (input, outcome) pairs from which to make predictions about outcomes with user input. This subgroup includes two algorithms, Regression and Classification.

The difference between Regression and Classification in supervised learning. While Classification is that the label or input variables can be grouped into finite groups, Regression is the input variables that cannot be grouped, but they must be specific real values.

- **Application of Classification:** This method is widely used in recognizing human faces in images, recognizing handwriting, identifying gender through images, ...
- **Application of Regression:** this method works well mostly in numerical form, so it is often applied to predict the price of houses, cars, ...

3.7.1 Regression Algorithm

There are many different types of regression algorithms suitable for certain types of models, such as: linear regression, polynomial regression, logistic regression, exponential Regression, and more. In conclusion, in this application to predict the number of cases or deaths per day with many fluctuations up and down, polynomial Regression is the right choice for the project.

Polynomial Regression [8] is a representation of the predictor variables of the function with an order of 2 or more. So when order is 1, we have Linear Regression, which is a special form of polynomial Regression.

How polynomial Regression works in the application to predict disease developments after any period of time. The mathematical formula behind this algorithm is shown in Figure 11. Where a_m is the values to find for the predicted system of equations, n is the amount of input data , and m is the degree of the equation to be found.

$$\begin{bmatrix} \sum_{i=0}^{n} x_i^{2m} & \dots & \sum_{i=0}^{n} x_i^{(m+2)} & \sum_{i=0}^{n} x_i^{(m+1)} & \sum_{i=0}^{n} x_i^{m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sum_{i=0}^{n} x_i^{(m+2)} & \dots & \sum_{i=0}^{n} x_i^{4} & \sum_{i=0}^{n} x_i^{3} & \sum_{i=0}^{n} x_i^{2} \\ \sum_{i=0}^{n} x_i^{(m+1)} & \dots & \sum_{i=0}^{n} x_i^{3} & \sum_{i=0}^{n} x_i^{2} & \sum_{i=0}^{n} x_i \\ \sum_{i=0}^{n} x_i^{m} & \dots & \sum_{i=0}^{n} x_i^{2} & \sum_{i=0}^{n} x_i & n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ .. \\ .. \\ a_m \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} y_i x_i^{m} \\ .. \\ \sum_{i=0}^{n} y_i x_i^{2} \\ \sum_{i=0}^{n} y_i x_i \\ \sum_{i=0}^{n} y_i \end{bmatrix}$$

Figure 11: Polynomial Regression formula

As we can see, for different orders, there will be a corresponding predictor function to suit different model types (see Figure 12).

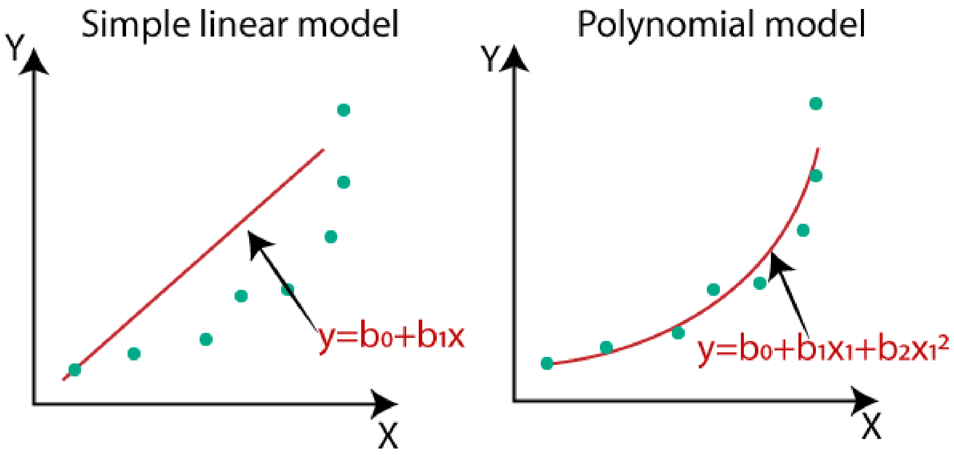


Figure 12: Regression Models corresponding to different degrees

3.7.2 Choose the optimal polynomial order for the model

Ways to choose the right model to give stable outcomes (low bias and low variance), as well as avoid underfiting or overfiting [9] (see Figure 13).

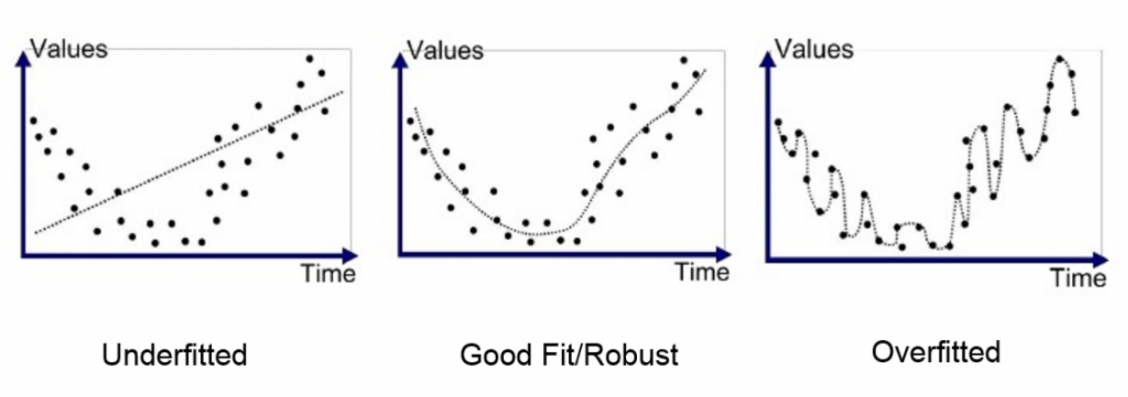


Figure 13: Under fitting and overfitting in regression

- **Under fitting phenomenon:** when the predicted values deviate much from the given training data set, the harmful effect causes high bias.
- **Overfitting phenomenon:** when the prediction values try to fit at most with the given training data set, the harm causes high variance (performs very well on the training data set, but fails when predicting on the unknown values. see)

The method using Mean Square Error [10] is calculated from the average of the RSS (sum of the squares of the residuals, see Figure 14) from all the input data points for the mth order polynomial.

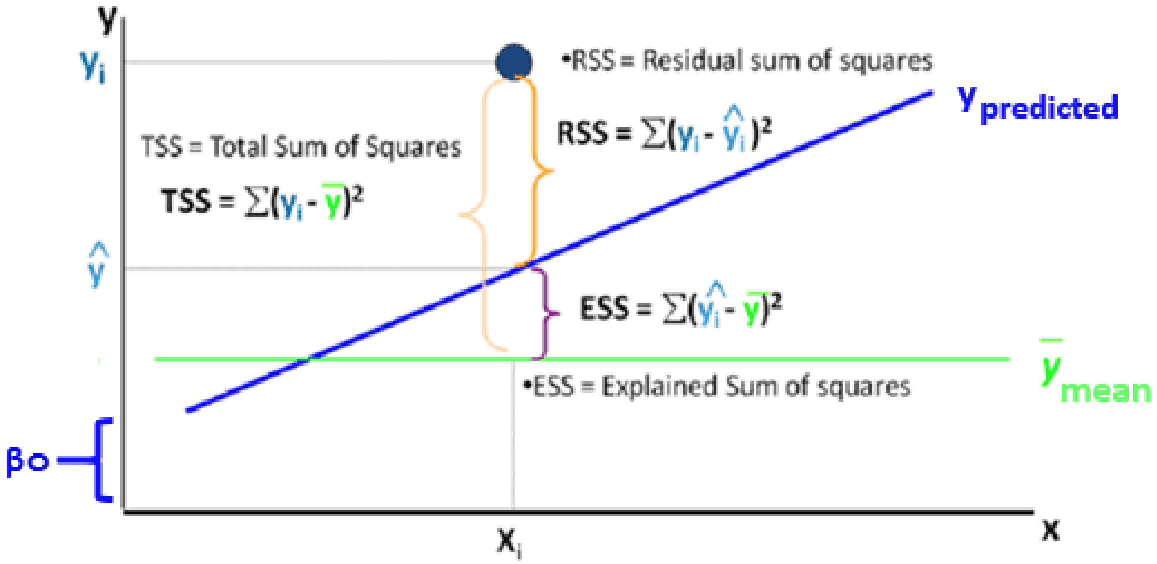

y
y_i
•RSS = Residual sum of squares
$y_{predicted}$
TSS = Total Sum of Squares
$TSS = \sum(y_i - \bar{y})^2$
$RSS = \sum(y_i - \hat{y}_i)^2$
$\hat{y}$
$ESS = \sum(\hat{y}_i - \bar{y})^2$
$\bar{y}_{mean}$
•ESS = Explained Sum of squares
βo
X_i
X

Figure 14: RSS, ESS and TSS in regression model

Values from Mean Square Error's mth orders, the closer any value is to 0, the better the model's fit. However, increasing the model's order also increases its complexity, so to avoid overfitting, we should choose the order with the highest volatility that will give the appropriate mode. In the example in Figure 15, order 2 will be assumed to be best fit the regression model.

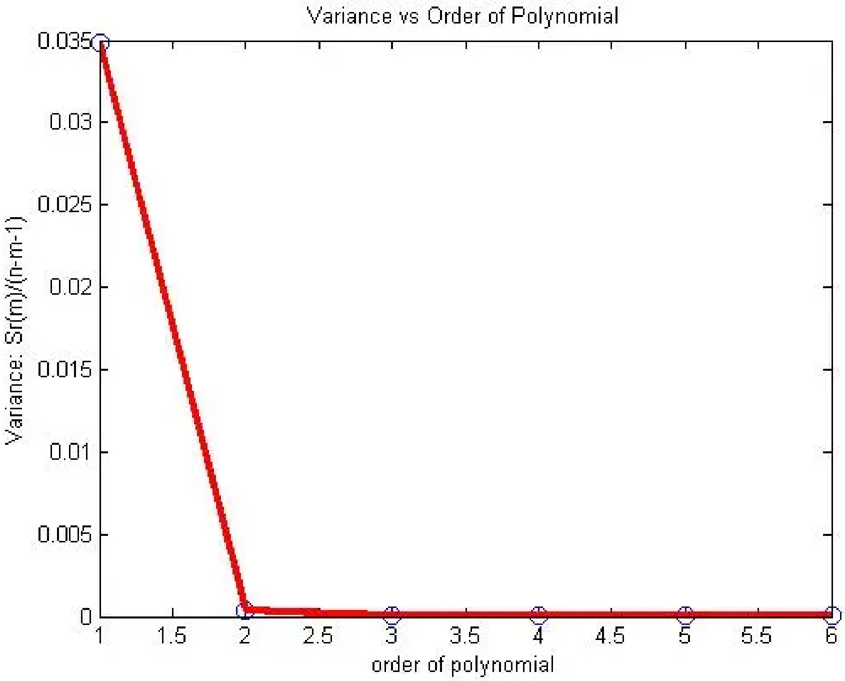


Figure 15: Optimal order regression with Mean Square Error example

Another evaluation method is based on the Coefficient of determination (R Square - R ^{2}) variable to assess the fit of the model [11]. R ^{2} has a formula calculated from RSS (sum of the squares of the residuals), so this is also chosen as an option to consider the appropriateness of the model for the application of predicting the number of Covid-19 cases. The higher the R Square value, the better the explanation for the predicted values.

$$TSS = RSS + ESS$$

$$But, R^2 = 1 - \frac{RSS}{TSS}$$

Figure 16: R Square formula

We see that the R^2 is inversely proportional to the RSS, but they are correlated with each other, so the use of the R^2 or Mean Square Error methods as mentioned above are equally meaningful in choosing the appropriate regression model. .

3.8 K-Means Algorithm for Countries Recommendation feature

The application includes a feature that gives suggestions about countries with similar epidemic status to the country the user is viewing. This feature is developed on a content-based recommendation system with the use of the K-Mean algorithm in machine learning.

There are two main types of recommendation systems [12], including:

- **Content-based recommendation system** : provides suggestions from clustering data by features or categories. Make content suggestions that are similar in features and do not need to be based on the reviews of other users.
- **Collaborative Recommendation System** : makes recommendations based on reviews of previous users. The Covid-19 application cannot have the assessment of the disease status of a country from the user, so this suggestion system is not selected.

3.8.1 K-Means Algorithm

K-means algorithm belongs to the group of unsupervised learning [13] of machine learning (see Figure 17). In addition to K-Means, DB Scan is also the algorithm that is said to be more optimal in data clustering (no need to choose the number of clusters, automatically removing noise variables, works well on many models) [14]. However, to be able to make predictions, K-means is the algorithm of choice for this application because it does not remove the noise variable, so all variables are clustered to recommend into different lists. Avoid data not belonging to any cluster.

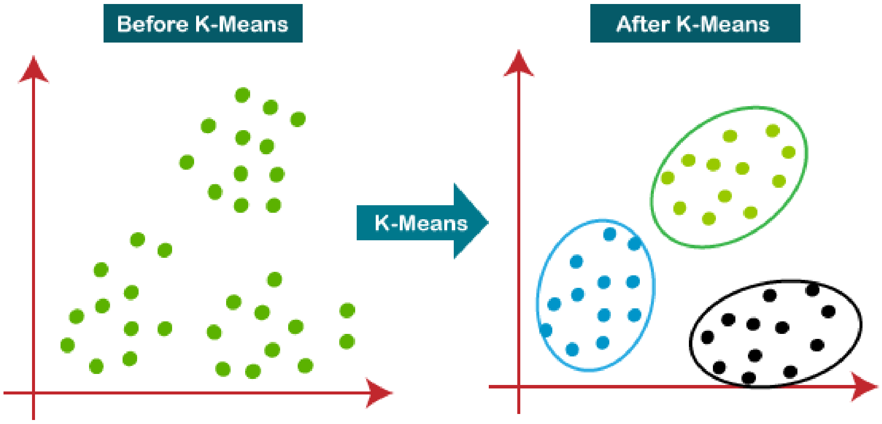


Figure 17: Clustering apply K-Means algorithm

Translation features are applied to characterize the translation situation in a country that the application considers appropriate to be included as input values for the data points before input into the K algorithm. Means includes:

- **"casesPerOneMillion"** : number of infections per 1 million people
- **"oneCasePerPeople"** : 1 case will appear for every number of people
- **"deathsPerOneMillion"** : number of deaths per 1 million people
- **"oneDeathPerPeople"** : 1 death for every number of people

The above epidemic parameters were chosen to be representative of the characteristics of each country because they do not bring about differences in population and territory sizes of different countries, making them more correlated when compared epidemic similarities

The way the K-means algorithm works is illustrated by the cycle diagram below in Figure 18.

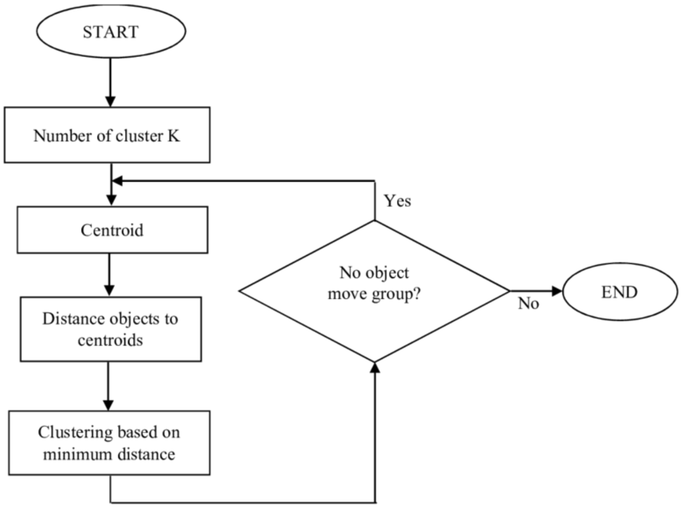

START
Number of cluster K
Centroid
Distance objects to centroids
Clustering based on minimum distance
No object move group?
Yes
No
END

Figure 18: K-Means Algorithm operation process

In the application about suggesting countries with similar epidemic status. The steps will include:

1. Option 10 Clusters (K=10)
2. Give random 10 centroids
3. Calculate the Euclidean distance from the centroids to all data points in turn
4. Reassign the points to one of the nearest centroids to form K clusters

5. Recalculate the mean of the centroids and continue from step 3 until there is no longer a change in the clusters when assigning points to the centroids
6. Form 10 clusters (each cluster is a list of countries with similar translation)

After the clustering of countries with similar disease outbreaks is completed, the implementation of suggestions for users will be conducted in chapter 4 of this thesis.

3.8.2 Choose the optimal number of clusters in K-Means

Method for choosing the optimum number of clusters K in K-Means. We can use Elbow Method [15] This method tells us how the fitness of each K-value correlates with the WCSS Value (Within Cluster Sums of Square) aka Distortion. The number of cluster with the largest variation would be the right choice for the number of clusters needed, for example the image of the position of the elbow on our arm (see Figure 19).

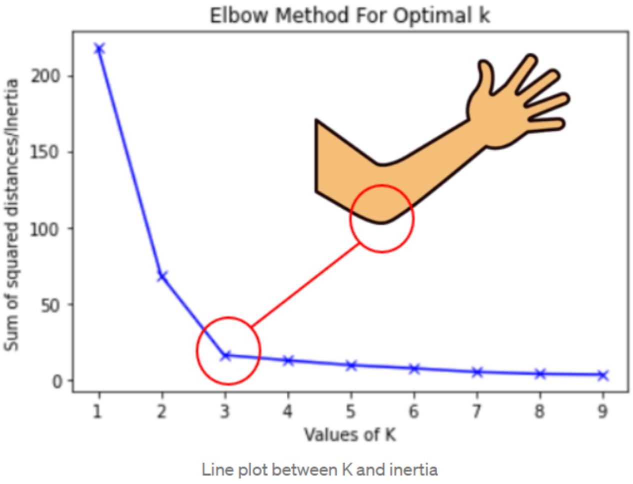
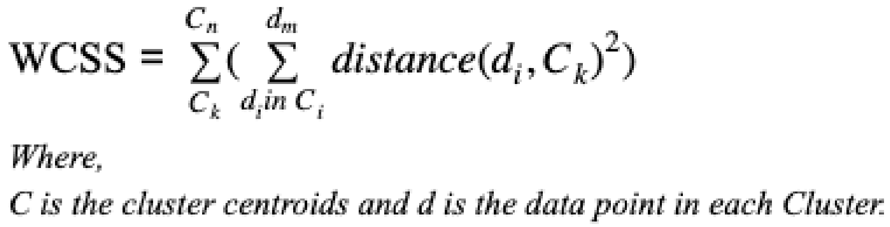


Figure 19: Elbow method for optimal k

Where WCSS Value: is calculated as the average sum of the Euclidean distances of the data points to its principal centroids with the formula in Figure 20.



$$\text{WCSS} = \sum_{C_k}^{C_n} \left(\sum_{d_i \, in \, C_i}^{d_m} distance(d_i, C_k)^2 \right)$$

Where,

C is the cluster centroids and d is the data point in each Cluster.

Figure 20: Formula to calculate WCSS

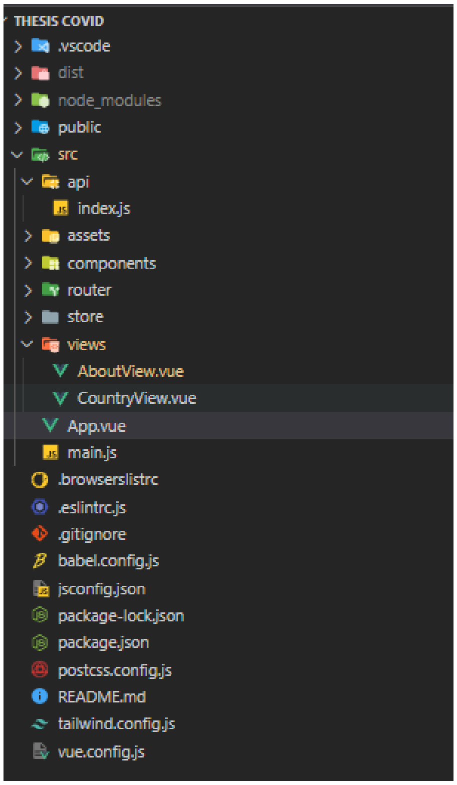
CHAPTER 4: IMPLEMENT AND RESULTS

4.1 Implement

This chapter will carry out implementation and development into a complete application, as well as specific analysis steps. Finally, solutions and results.

4.1.1 File structure

Organize file management in the application as shown in Figure 21. Files are closed and called into folders with their own functions suitable for reuse using import/export methods or components when working with JavaScript and Vue Framework.



```
THESIS COVID
> .vscode
> dist
> node_modules
> public
v src
  v api
      index.js
  > assets
  > components
  > router
  > store
  v views
      AboutView.vue
      CountryView.vue
    App.vue
    main.js
  .browserslistrc
  .eslintrc.js
  .gitignore
  babel.config.js
  jsconfig.json
  package-lock.json
  package.json
  postcss.config.js
  README.md
  tailwind.config.js
  vue.config.js
```

Figure 21: File structure management

4.1.2 Choosing suitable types of chart

Working with the Highcharts charting library, allows us to familiarize ourselves with many types of charts, but only a few charts are best suited to the application of the epidemic situation to represent the most intuitive parameters.

- **Spark line chart :** shows an up or downtrend over a certain period of time, showing how the current total number of cases has changed with the previous days.
- **Mix chart between Scatter and Line :** this duo will be used in predicting the number of future cases. If you replace the line chart with a bar chart for the prediction parameter, it will hide the scatter elements, so the line chart is the most suitable option.
- **Pie Chart:** represents the parameters as a percentage, convenient for users to compare the difference in the number of cases between the current and the previous day.
- **Bar chart and line chart:** shows the number of daily shifts and compares with the weekly average.
- **Country Map:** country map by states or provinces with chroma by magnitude of variable
- **Heat chart:** The chart is arranged according to the appropriate days of the week to observe and compare the outstanding epidemic developments on the days of the week.
- **Stack chart:** a chart that integrates 3 components in the form of percentages, helping users to identify the division of new infections, deaths, and recoveries per day
- **Area Chart:** is the chart chosen to display vaccine information and the correlation of deaths per day
- **Spline Area Chart:** similar to the area chart, but the interface is softer and more wavy
- **Column Group Chart:** column chart standing in groups of 3 columns per country, each column represents a translation parameter for users not only to compare countries, but

also to compare the correlation of parameters in each country. countries on that continent.

- **Bubble chart:** is chosen to show the general situation of cases worldwide, so that users can immediately identify which country is the most heavily affected compared to the rest.
- **World Map:** a hierarchical worldwide map of the world that can be adjusted to zoom in and out for each geographical territory and change observations according to the total number of cases, total deaths and total recoveries.
- **Bar chart with drill down:** 6 continents are displayed as horizontal not vertical because when manipulating each horizontal bar the user will see all the countries under that continent. If displayed in columns, there will not be enough room to display all countries.

4.1.3 Process data from API for compatibility with charts and usage purposes

Because not the data put into the charts is just taken from the api. Most of them must be processed to suit the intended use and user requirements before rendering the application.

- Generates daily case count data on a timeline from the cumulative total of infections per day, and reuses the code for getting the number of cases per day of other data such as the number of deaths per day, the number of recoveries per day or the number of vaccines administered per day.
- Process the average value for the last 7 days based on the number of infections per day processed and outlined above.
- Process the data for the vaccine efficiency section from the total used vaccine daily data for 1 country. Break down the data into periods by comparing the population of that

country. Over the period, the correlation of the average number of cases and the number of deaths will also be calculated next.

- Transform the data accordingly to include in the Heat chart for the display of daily shift data by day of the week. The data of the number of shifts per day is stored in the form of a linear array, so when entering this chart form, we need to process the logic of dividing the data into small arrays for 7 days and then saving them all into a common array.
- From the data of the entire country, we grouped countries by continent to make comparisons between countries. By relying on the continent parameter, we loop through the countries in turn and then map the countries with the continent name into different arrays, we get 6 new arrays corresponding to 6 continents.
- World map or country map is a part of a map of countries that is distinguished by the color tones of regions, so the data it requires is aggregated in the form of hc-key (country code name or province code name, state of a country) and the numeric variable is the total number of infections or the total number of deaths and recoveries. All must be processed and saved as a common array.
- The Column drill down chart requires different arrays of 6 arrays for all countries by continent, and 1 array for an overview of the total number of cases for each continent, all of the above data generated from the method iterates over the data "allCountries" retrieved from the previous API.
- In the list of countries with epidemics in the search section, it is necessary to conduct 2 main parts including sorting the countries in descending order by different categories such as today's cases, total infections, and more. In addition, process the search feature by country name with the filter function in JavaScript.

4.1.4 Implement the feature of predicting the epidemic situation

The input data is the daily number of infections or the number of deaths per day that is taken from the data per day after processing in the previous section. The input data is given as an array with outcome(y) being integer values greater than or equal to 0, and input(x) being numeric values starting from 0 like consecutive days to current time. All input points from this (x, y) value pair will be represented on the graph as a scatter.

After applying and implementing the Regression.js library, the result is a prediction equation according to the selected order. Next, We save it as an array with the predicted values plotted into a corresponding system of equations to represent on the chart as a line chart.

The next steps will be to choose any time after the current date so that the system relies on the previous equation and calculates the prediction that the user wants to see.

In addition, the application also allows changing the number of input data points or change the order value of polynomial regression so that the user can choose the prediction model suitable for the help of the R Square value.

4.1.5 Implement the feature of similar country suggestion system on epidemics

The input data is a large array in which there are small arrays corresponding to each country with the epidemic. In the sub-array will include 4 elements as mentioned in the previous section, including the number of infections per million people, the number of deaths per million people, how many people will appear 1 infection and how many people will 1 death occurred. The above data elements are considered by this thesis to be suitable for classifying countries because it ignores information about the population and territory of each country.

After processing the input data source representing each country. We continue to apply K-means library to cluster data with 10 default clusters. As a result, after executing the algorithm, we get a large array with numeric elements (ranging from 1 to 10), these numbers are in the correct position of the array of countries in the input data.

So we continue to iterate through the countries again, if the cluster value of any country is equal to the value of the cluster that the user's country is viewing, we aggregate and save those countries as a new array. And that is the network of countries that have translation similarities with the country the user is viewing, from which the suggestion system is complete.

4.2 Results

After conducting the complete project code, we will have the results as below with the required features included in 2 main pages including: Country Overview and Worldwide Overview.

4.2.1 Country Overview

1. Highlight country statistics: outstanding statistics about a country's pandemic that most users are most interested in (see Figure 22).

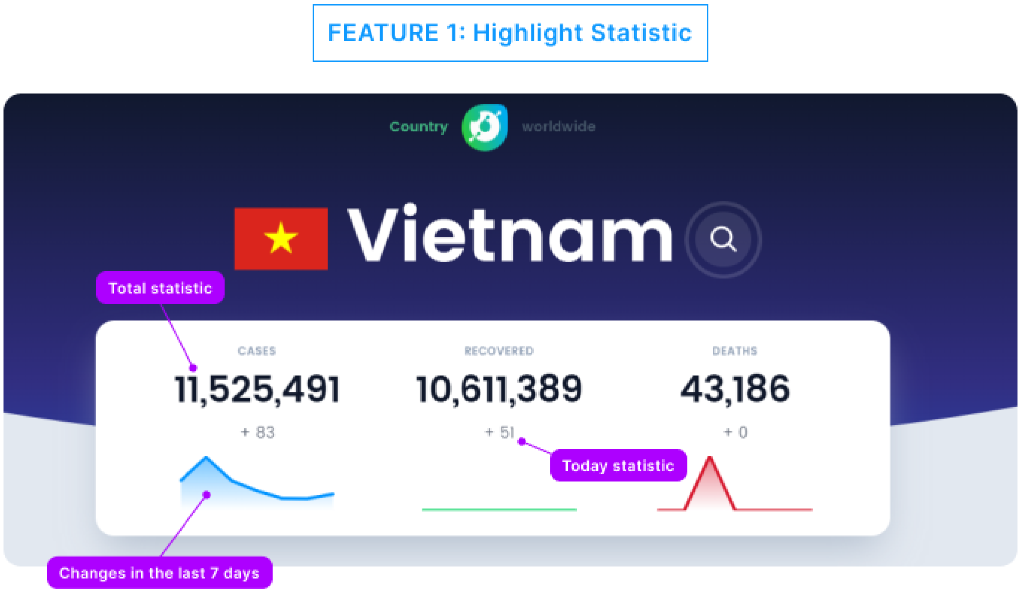
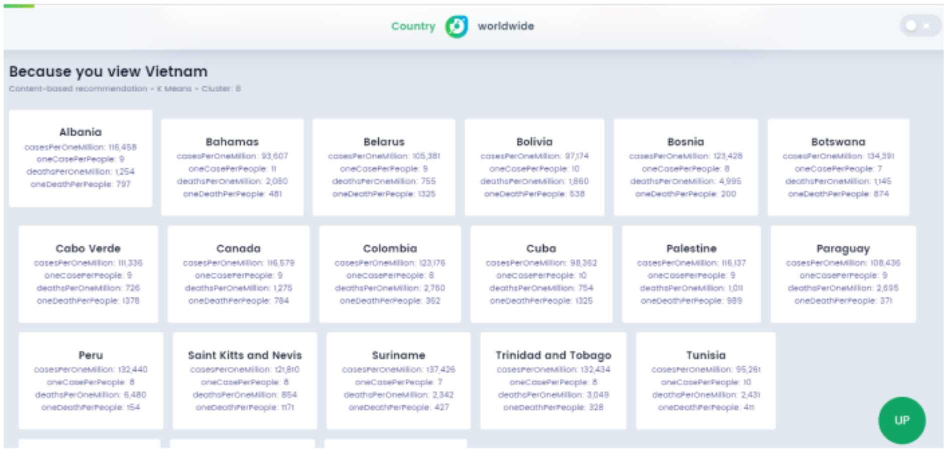


Figure 22: Highlight country statistics

2. Countries recommendation: A list of countries with similar epidemic status to the country the user is viewing, after being clustered by the K-Means algorithm, will be displayed to suggest to the user (see Figure 23).

FEATURE 2: Recommendation



Country worldwide

Because you view Vietnam

Content-based recommendation - K Means - Cluster: 8

Albania
casesPerOneMillion: 116,458
oneCasePerPeople: 9
deathsPerOneMillion: 1,254
oneDeathPerPeople: 797

Bahamas
casesPerOneMillion: 93,607
oneCasePerPeople: 11
deathsPerOneMillion: 2,080
oneDeathPerPeople: 481

Belarus
casesPerOneMillion: 105,381
oneCasePerPeople: 9
deathsPerOneMillion: 755
oneDeathPerPeople: 1325

Bolivia
casesPerOneMillion: 97,174
oneCasePerPeople: 10
deathsPerOneMillion: 1,860
oneDeathPerPeople: 538

Bosnia
casesPerOneMillion: 123,428
oneCasePerPeople: 8
deathsPerOneMillion: 4,995
oneDeathPerPeople: 200

Botswana
casesPerOneMillion: 134,391
oneCasePerPeople: 7
deathsPerOneMillion: 1,145
oneDeathPerPeople: 874

Cabo Verde
casesPerOneMillion: 111,336
oneCasePerPeople: 9
deathsPerOneMillion: 726
oneDeathPerPeople: 1378

Canada
casesPerOneMillion: 116,579
oneCasePerPeople: 9
deathsPerOneMillion: 1,275
oneDeathPerPeople: 784

Colombia
casesPerOneMillion: 123,176
oneCasePerPeople: 8
deathsPerOneMillion: 2,760
oneDeathPerPeople: 362

Cuba
casesPerOneMillion: 98,362
oneCasePerPeople: 10
deathsPerOneMillion: 754
oneDeathPerPeople: 1325

Palestine
casesPerOneMillion: 116,137
oneCasePerPeople: 9
deathsPerOneMillion: 1,011
oneDeathPerPeople: 989

Paraguay
casesPerOneMillion: 108,436
oneCasePerPeople: 9
deathsPerOneMillion: 2,695
oneDeathPerPeople: 371

Peru
casesPerOneMillion: 132,440
oneCasePerPeople: 8
deathsPerOneMillion: 6,480
oneDeathPerPeople: 154

Saint Kitts and Nevis
casesPerOneMillion: 121,810
oneCasePerPeople: 8
deathsPerOneMillion: 854
oneDeathPerPeople: 1171

Suriname
casesPerOneMillion: 137,426
oneCasePerPeople: 7
deathsPerOneMillion: 2,342
oneDeathPerPeople: 427

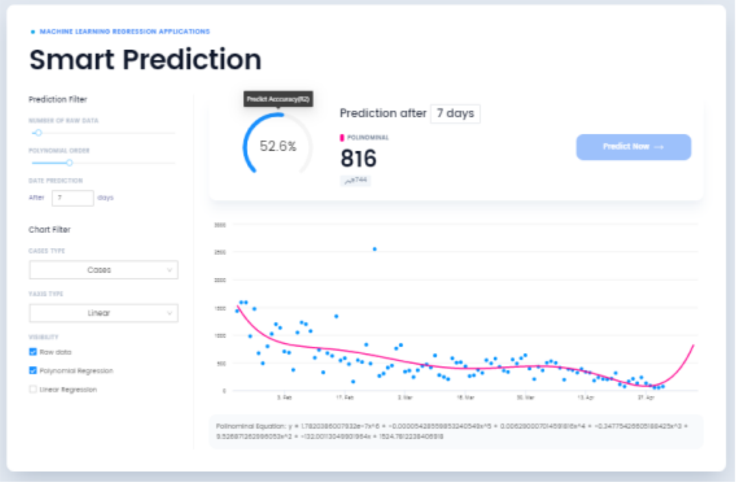
Trinidad and Tobago
casesPerOneMillion: 132,434
oneCasePerPeople: 8
deathsPerOneMillion: 3,049
oneDeathPerPeople: 328

Tunisia
casesPerOneMillion: 95,261
oneCasePerPeople: 10
deathsPerOneMillion: 2,431
oneDeathPerPeople: 411

Figure 23 : Countries with similar epic situation

3. The function to make predictions about the disease situation: users can adjust the values and choose a time in the future for the system to make predictions according to the system of equations obtained from the Regression algorithm (see Figure 24).

FEATURE 3: Prediction



MACHINE LEARNING REGRESSION APPLICATIONS

Smart Prediction

Prediction Filter

NUMBER OF RAW DATA

POLYNOMIAL ORDER

DATE PREDICTION

After 7 days

Chart Filter

CASES TYPE

Cases

YAXIS TYPE

Linear

VISIBILITY

☑ Raw data

☑ Polynomial Regression

☐ Linear Regression

Predict Accuracy(R2)

52.6%

Prediction after 7 days

POLINOMINAL

816

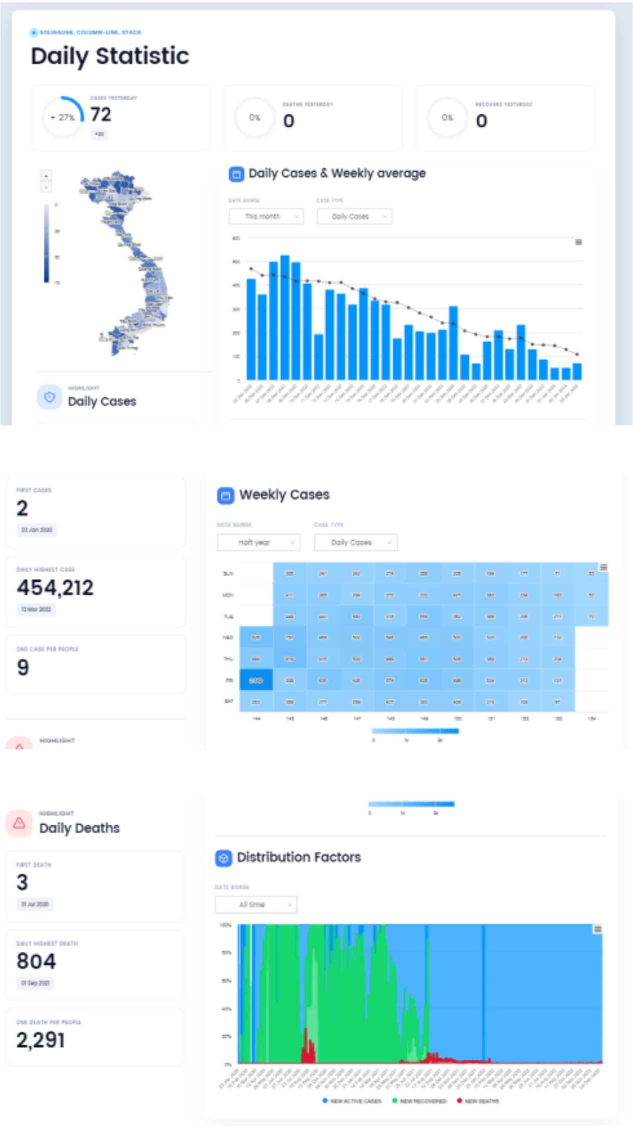
Predict Now →

Polinominal Equation: y = 1.7820386007932e-7x^6 + -0.00005428559853240549x^5 + 0.006290007014591816x^4 + -0.34775426605188425x^3 + 9.526871262996053x^2 + -132.00113049901964x + 1524.7812238406918

Figure 24: Predicting future growth of infections and deaths

4. Daily Statistic: The data of daily infections, after being processed from the total number of cases per day, will be presented in different charts such as weekly daily chart, stack chart, .. Other outstanding national translation parameters country and a map of that country (see Figure 25).

FEATURE 4: Daily Statistic



SOLIDGAUGE, COLUMN-LINE, STACK

Daily Statistic

CASES YESTERDAY 72 +20

DEATHS YESTERDAY 0

RECOVERS YESTERDAY 0

Daily Cases & Weekly average

DATE RANGE: This month

CASE TYPE: Daily Cases

HIGHLIGHT

Daily Cases

FIRST CASES 2 23 Jan 2020

DAILY HIGHEST CASE 454,212 12 Mar 2022

ONE CASE PER PEOPLE 9

Weekly Cases

DATE RANGE: Haft year

CASE TYPE: Daily Cases

HIGHLIGHT

Daily Deaths

FIRST DEATH 3 31 Jul 2020

DAILY HIGHEST DEATH 804 01 Sep 2021

ONE DEATH PER PEOPLE 2,291

Distribution Factors

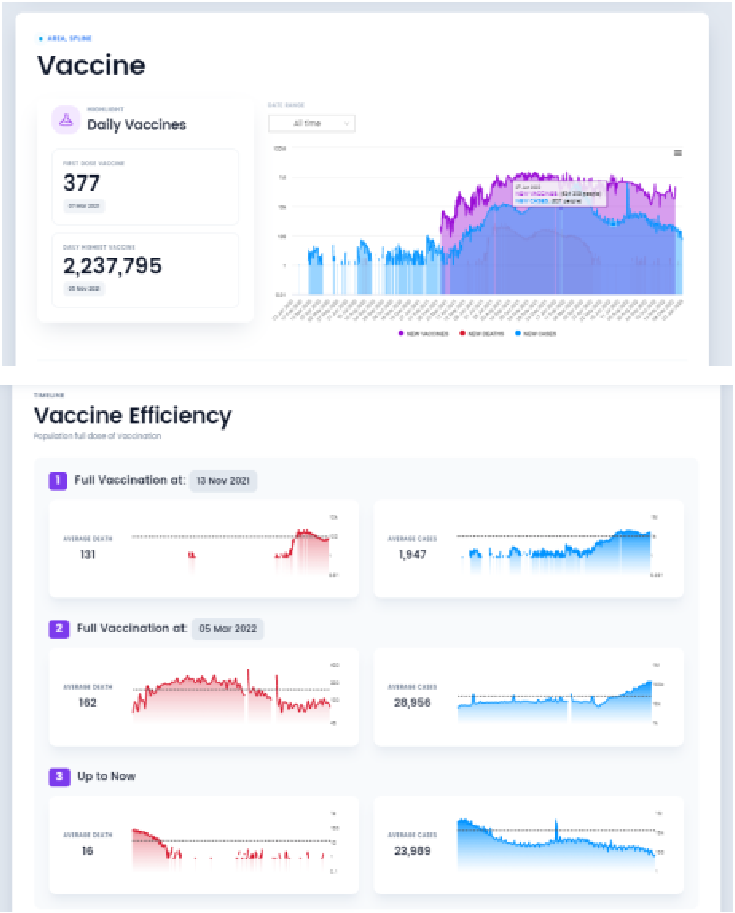
DATE RANGE: All time

Figure 25: Number of infections per day in the country

5. Vaccine statistic and vaccine efficiency: The amount of vaccine used per day in the country the user is viewing and the correlation between it and the number of deaths or

new infections per day. From there, we look at the effectiveness of the vaccine used in that country (see Figure 26).

FEATURE 5: Vaccine Statistic



AREA, SPLINE

Vaccine

Daily Vaccines

FIRST DOSE VACCINE
377

DAILY HIGHEST VACCINE
2,237,795
05 Nov 2021

DATE RANGE
All time

TIMELINE

Vaccine Efficiency

Population full dose of vaccination

1 Full Vaccination at: 13 Nov 2021

AVERAGE DEATH
131

AVERAGE CASES
1,947

2 Full Vaccination at: 05 Mar 2022

AVERAGE DEATH
162

AVERAGE CASES
28,956

3 Up to Now

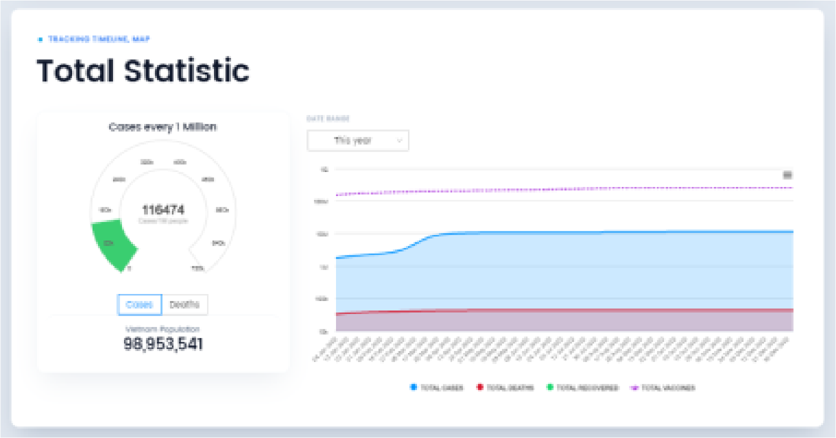
AVERAGE DEATH
16

AVERAGE CASES
23,989

Figure 26: The amount of vaccine used per day and the effectiveness of the vaccine

6. Total Statistic: Overview of pandemic developments according to the cumulative total of the country the user is viewing. Users can change the date range to suit their needs (see Figure 27).

FEATURE 6: Total Statistic


TRACKING TIMELINE, MAP

Total Statistic

Cases every 1 Million

116474

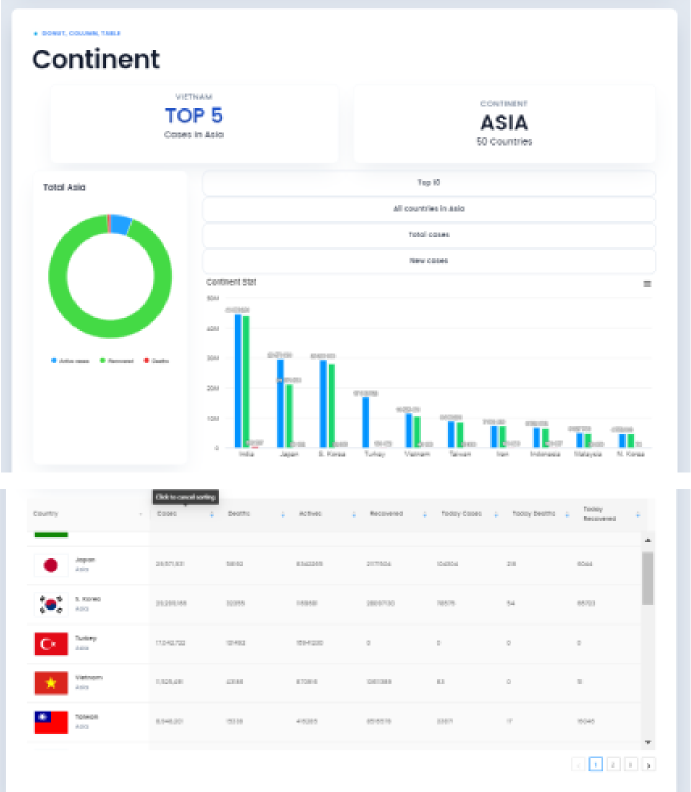
Vietnam Population
98,953,541

This year

Figure 27: Epidemic overview by total

7. Comparing countries in the same continent includes an overview of the continent of the country the user is viewing, in addition can see where that country is in comparison to the rest of the country in its continent (see Figure 28).

FEATURE 7: Continent Statistic


DONUT, COLUMN, TABLE

Continent

VIETNAM
TOP 5
Cases in Asia

CONTINENT
ASIA
50 Countries

Total Asia

Country	Cases	Deaths	Actives	Recovered	Today Cases	Today Deaths	Today Recovered
Japan Asia	23,571,831	58152	8342269	2171504	104304	218	8064
S. Korea Asia	29,239,168	32355	1169581	28097130	78575	54	85703
Turkey Asia	17,242,722	101492	15941230	0	0	0	0
Vietnam Asia	11,526,491	43188	870816	10613488	83	0	18
Taiwan Asia	8,946,201	15338	415283	8515579	33371	17	15040

‹ 1 2 3 ›

Figure 28: Correlation comparison between countries on the same continent

8. The search engine and sorting the list of countries with epidemics allows searching by country name and sorting the list of countries based on the number of infections from high to low (see Figure 29).

FEATURE 8: Country Search

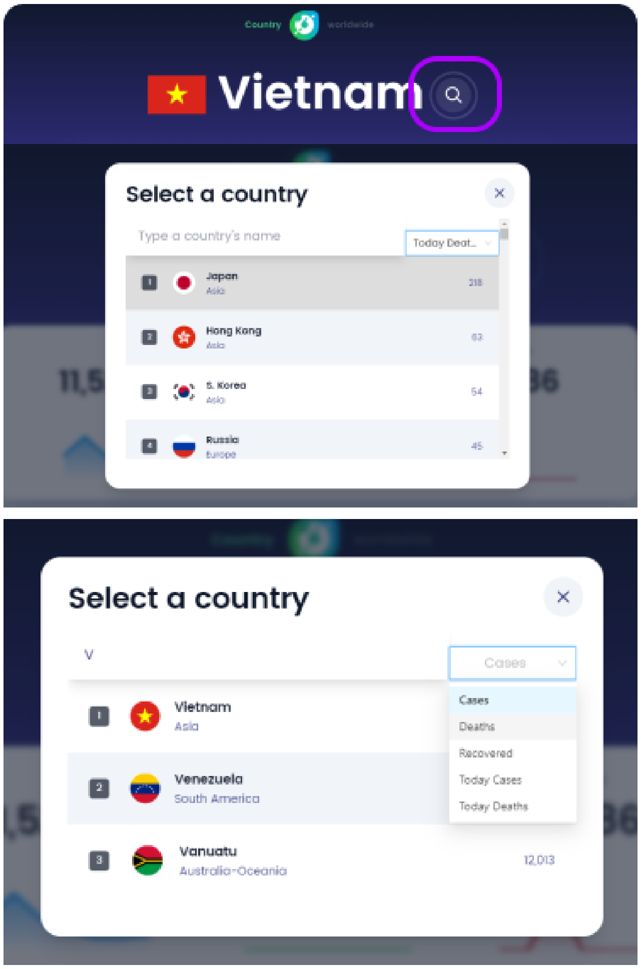
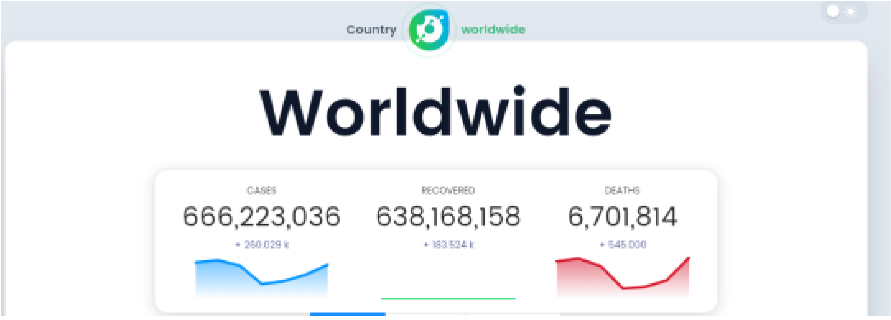


Figure 29: Country search filter

4.2.2 Worldwide Overview

1. Worldwide highlights: the outstanding general data that users care about the pandemic situation around the world (see Figure 30).

FEATURE 1: Worldwide Highlight



Country worldwide

Worldwide

CASES
666,223,036
+ 260.029 k

RECOVERED
638,168,158
+ 183.524 k

DEATHS
6,701,814
+ 545.000

Figure 30: Highlights for the global epidemic overview

2. Worldwide daily statistic: general data on the number of cases per day calculated on a global scale (see Figure 31).

FEATURE 2: Worldwide Daily Statistic

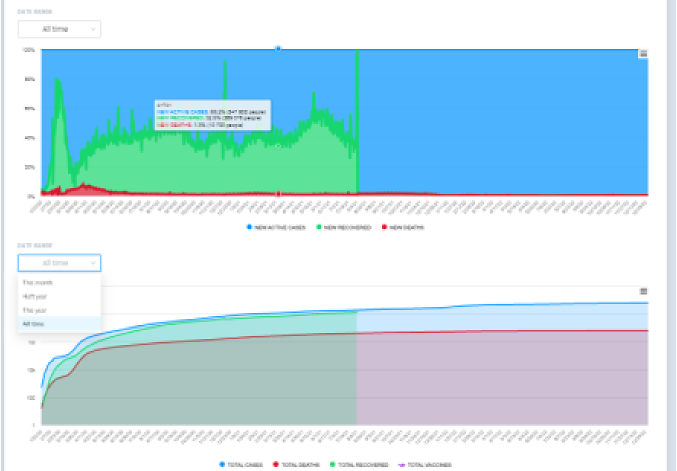
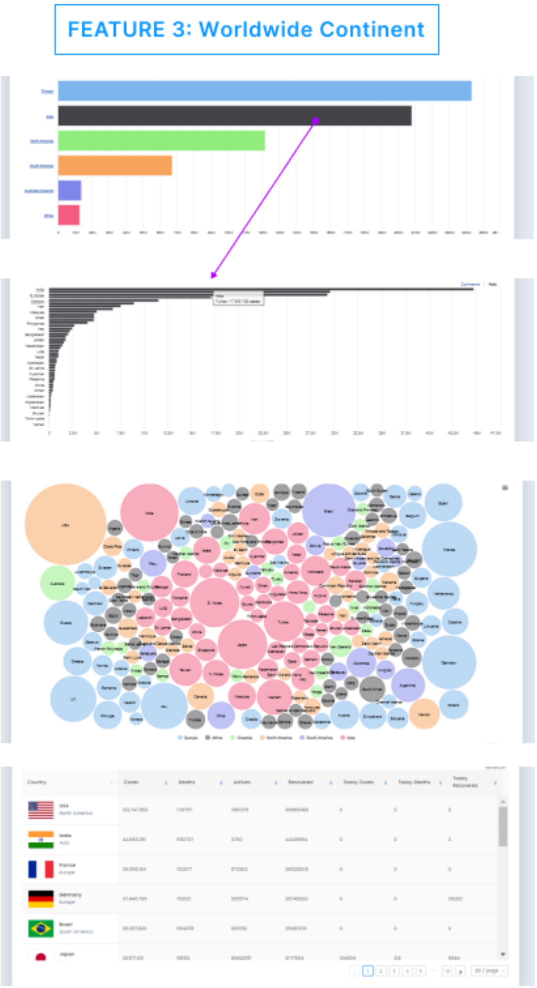


Figure 31: Daily global epidemic situation

3. Compare global continents: allows interaction for easy comparison between continents and comparisons between countries within each continent (see Figure 32).



Country	Cases	Deaths	Actives	Recovered	Today Cases	Today Deaths	Today Recovered
USA North America	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0
India Asia	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0
France Europe	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0
Germany Europe	[illegible]	[illegible]	[illegible]	[illegible]	0	0	[illegible]
Brazil South America	[illegible]	[illegible]	[illegible]	[illegible]	0	0	0
Japan	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Figure 32: Worldwide continental comparison feature

4. Map of worldwide epidemic developments in chroma: allows users to search and view different countries around the globe according to their geographic location, and clearly display data on the total number of infections or the total number of cases mortality and recovery (see Figure 33).

FEATURE 4: Worldwide map

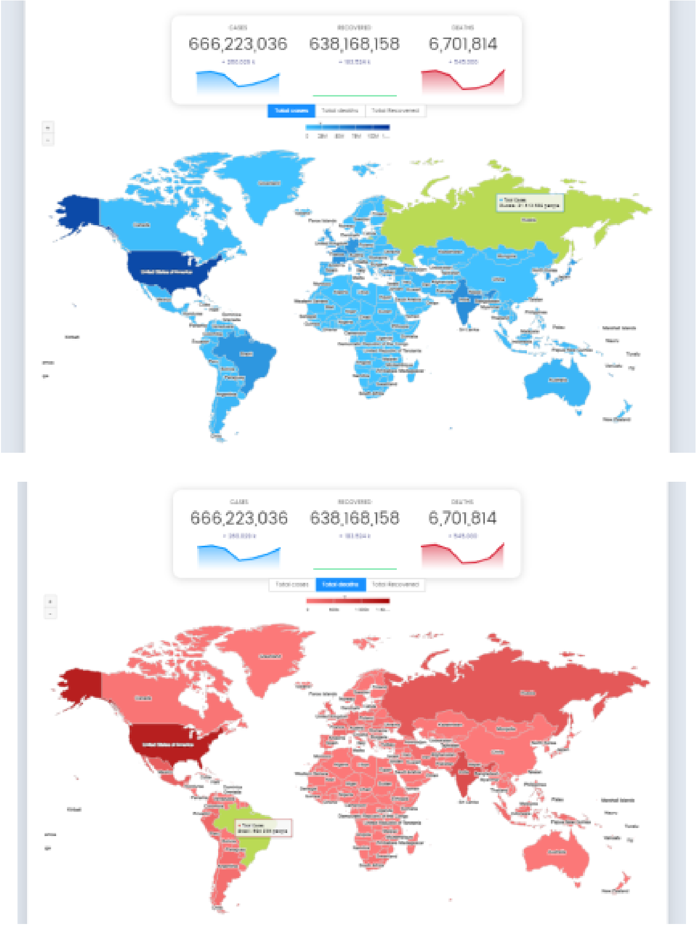


Figure 33: Worldwide world map

4.2.3 Deployment

After the above development steps, the App is moved to production and deployed with third parties including Github (hosting) and Netlify (domain) so that users can access it directly on the devices they use available as laptop, smartphone or tablet at this address: wagtrung-covid.netlify.app/

CHAPTER 5: DISCUSSION AND EVALUATION

5.1 Discussion

After making this web application, the author of this thesis has achieved such a great experience in Web development. Besides, the reusable coding for another upcoming pandemic. In this application, organize the compiled API into a JS file and imported when used. Therefore, we can take advantage of this advantage to replace other APIs with similar properties for future pandemics, not just stopping at the Covid-19 Pandemic. And the calculation operations as well as the code, because they are written in the form of components and arranged in the appropriate file structure, will be completely reused up to 90% for the preparation of the upcoming explosive pandemics, despite the pandemic. The Covid-19 epidemic is not completely over and is not considered an endemic disease globally, which still has thousands of deaths every day.

In addition, the data Visualization for Covid-19 application has built and developed complete charts that match user needs in the most intuitive and accessible way. Besides, it also meets intelligent features thanks to the application of machine learning algorithms that have brought fun and visualization suitable for a wide range of potential users of all ages who are interested in health, people who love to travel, international students and families with children studying abroad or are medical students and pandemic researchers for the effectiveness of vaccines or countries with similar epidemics for concern about the spread of mutations.

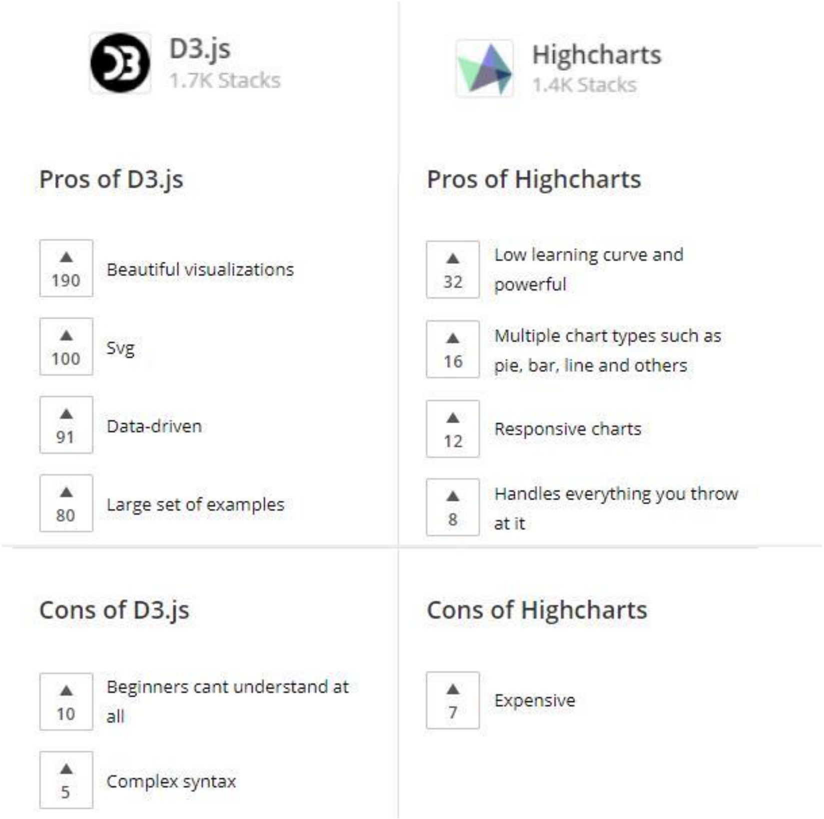
5.2 Comparison

5.2.1 Data visualization and raw data

In reality, a visualized data provides much better information to users which is easy to read and understand. Also, developing data visualization applications will enable users to interact with data dynamically rather than read only. Comparing with some other tools that working with raw data like Excel, it can save much time and cost for making software, yet this can make users waste lots of time to learn the functions to analyze data. In short, visualization is essential for people not only in IT related fields but also other majors.

5.2.2 Highchart.js and D3,js

After the implementation stage, there are significant differences between two DOM libraries:



D3.js
1.7K Stacks

Highcharts
1.4K Stacks

Pros of D3.js

▲ 190 Beautiful visualizations
▲ 100 Svg
▲ 91 Data-driven
▲ 80 Large set of examples

Pros of Highcharts

▲ 32 Low learning curve and powerful
▲ 16 Multiple chart types such as pie, bar, line and others
▲ 12 Responsive charts
▲ 8 Handles everything you throw at it

Cons of D3.js

▲ 10 Beginners cant understand at all
▲ 5 Complex syntax

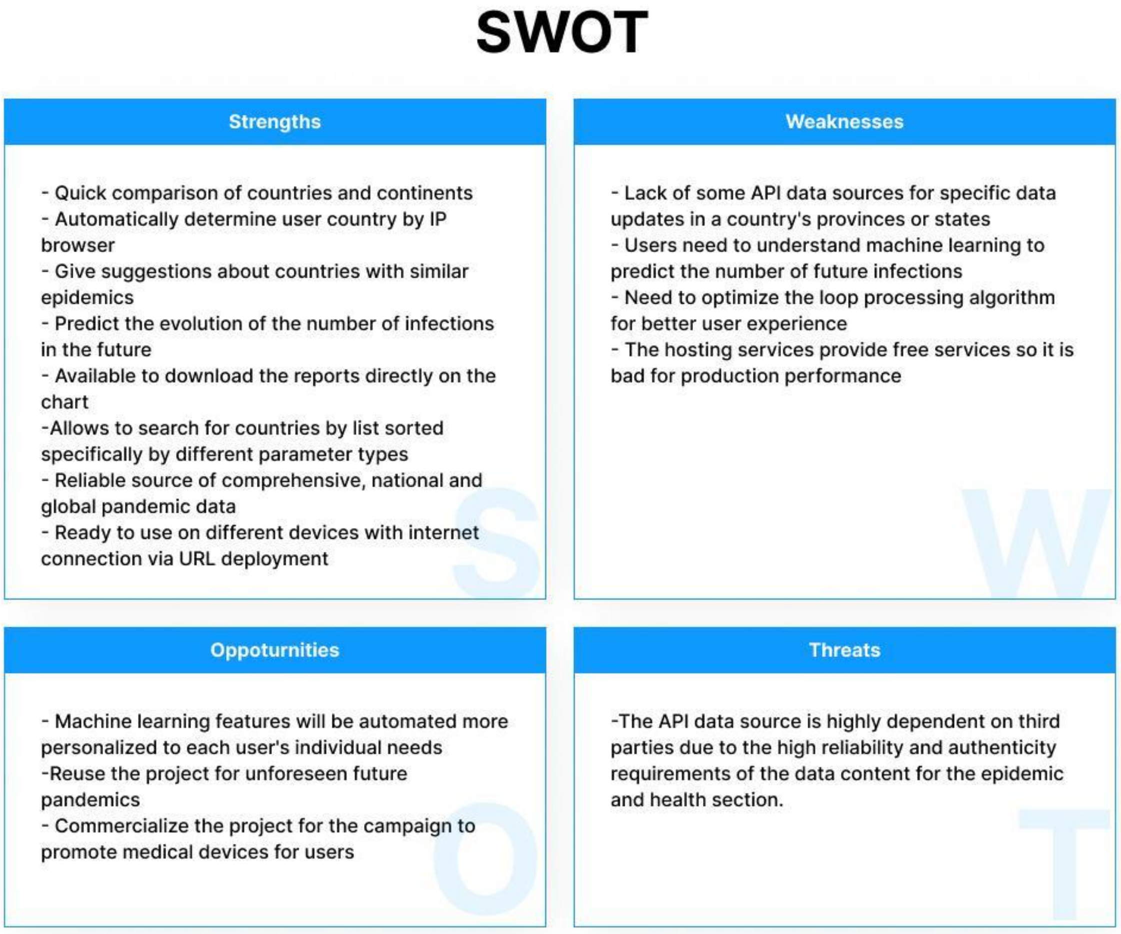
Cons of Highcharts

▲ 7 Expensive

Table 1: Highcharts.js and D3.js pros and cons

5.3 Evaluation

Below is the table for SWOT analysis of the overall system that has been developed, this provides the base for future work improving the application.



SWOT

Strengths	Weaknesses
- Quick comparison of countries and continents - Automatically determine user country by IP browser - Give suggestions about countries with similar epidemics - Predict the evolution of the number of infections in the future - Available to download the reports directly on the chart -Allows to search for countries by list sorted specifically by different parameter types - Reliable source of comprehensive, national and global pandemic data - Ready to use on different devices with internet connection via URL deployment	- Lack of some API data sources for specific data updates in a country's provinces or states - Users need to understand machine learning to predict the number of future infections - Need to optimize the loop processing algorithm for better user experience - The hosting services provide free services so it is bad for production performance
Opporturnities	**Threats**
- Machine learning features will be automated more personalized to each user's individual needs -Reuse the project for unforeseen future pandemics - Commercialize the project for the campaign to promote medical devices for users	-The API data source is highly dependent on third parties due to the high reliability and authenticity requirements of the data content for the epidemic and health section.

Table 2: SWOT Analysis for Covid-19 data visualization application

CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1 Conclusion

With data visualization, the thesis demonstration application has achieved the basics of user requirements as well as objectives that combine both Vue and Highcharts to mutate DOM elements. Besides, it is necessary to understand and apply machine learning algorithms to apply in daily life to come up with intelligent features such as predicting pandemic developments and zoning countries with epidemic similarities. Two main features, Country Overview and Worldwide Overview, are unique separated for specific purposes; one for national concern and one for global situation. This graphical presentation technique requires the knowledge of a web designer for styling, a front-end developer for logical interactions, and a data scientist for analyzing data. Visualization may create a learning curve for people who are in IT-related fields, yet it can provide friendly interfaces to users who cannot process and imagine complex data.

6.2 Future works

Since this thesis project is a part of the Covid-19 Dashboard, it has so much potential to bring more features to people who are concerned about the Covid-19 epidemic about people who are looking to learn and research this disease.

With the features on this application, it has basically solved the shortcomings of the visual information display and the necessary features compared to the current applications in terms of users' needs and concerns about the pandemic. Not only about Covid-19 but also upcoming pandemics, so this application needs to be optimized to prepare for download and to respond to upcoming pandemics.

Besides, when the pandemic breaks out on a global scale, the need to purchase equipment such as masks or alcohol is essential for human needs, so in the future the application will be able to have more applications. Advertising products and services of third parties to commercialize the project when the number of users accessing the application is increasing day by day.

In addition, optimizing the processing of code logic to make the application run faster and reduce the user's waiting time is also a concern in the application development process in the future.

Finally, continuously research the user experience to change and improve UI elements on the application to provide a good and suitable user experience.

REFERENCES

[1] Khan, M., Adil, SF, Alkhathlan, HZ, Tahir, MN, Saif, S., Khan, M., & Khan, ST (2020). COVID-19: a global challenge with old history, epidemiology and progress so far. Molecules, 26(1), 39.

[2] Vyas, R. (2022). Comparative Analysis on Front-End Frameworks for Web Applications. International Journal for Research in Applied Science and Engineering Technology, 10(7), 298-307.

[3] Options: Lifecycle | Vue.js. (nd). https://vuejs.org/api/options-lifecycle.html

[4] Nelson, B. (2018). State Management. In Getting to Know Vue. js (pp. 127-147). Apress, Berkeley, CA.

[5] Kuan, J. (2012). Learning Highcharts. Packt Publishing Ltd.

[6] Parker, D. (2015). JavaScript with Promises: Managing Asynchronous Code. "O'Reilly Media, Inc.".

[7] Nasteski, V. (2017). An overview of the supervised machine learning methods. Horizons. b, 4, 51-62.

[8] Peckov, A. (2012). A machine learning approach to polynomial regression. Ljubljana, Slovenia, URL: http://kt. ijs. si/theses/phd_aleksandar_peckov. pdf.

[9] Guyon, X., & Yao, JF (1999). On the underfitting and overfitting sets of models chosen by order selection criteria. Journal of Multivariate Analysis, 70(2), 221-249.

[10] Allen, DM (1971). Mean square error of prediction as a criterion for selecting variables. Technometrics, 13(3), 469-475.

[11] Nagelkerke, NJ (1991). A note on a general definition of the coefficient of determination. biometrika, 78(3), 691-692.

[12] Das, D., Sahoo, L., & Datta, S. (2017). A survey on recommendation system. International Journal of Computer Applications, 160(7).

[13] Mahesh, B. (2020). Machine learning algorithms-a review. International Journal of Science and Research (IJSR).[Internet], 9, 381-386.

[14] Chakraborty, S., Nagwani, NK, & Dey, L. (2014). Performance comparison of incremental k-means and incremental dbscan algorithms. arXiv preprint arXiv:1406.4751.

[15] Syakur, MA, Khotimah, BK, Rochman, EMS, & Satoto, BD (2018, April). Integration k-means clustering method and elbow method for identification of the best customer profile cluster. In IOP conference series: materials science and engineering (Vol. 336, No. 1, p. 012017). IOP Publishing.

APPENDIX A: INSTALLATION GUIDE

This section only guides Windows users; macOS or Linux users can use other terminals to run. Note that when starting the Covid-19 data visualization application.

Software requirements : Node JS, VS Code Application

Project setup steps :

1. Visit and download the project at the link below https://github.com/wagtrung26/covid-wagtrung
2. At the downloaded folder, open the terminal and run the bellow code to set up the npm environment: **npm install**
3. Open VS Code Application, if you want in the application development environment, then type the code: **npm run serve** to start the project on the browser at localhost
4. Or if you want to see the application in production mode then type the code: **npm run build**

Project deployment :

The current application has been fully deployed at the address below https://wagtrung-covid.netlify.app/, so users can fully view the project directly without having to install it as mentioned above.