Deliverable 4: Use Cases and Requirements Report
SENG3011: THUMBNAILS

Zixiang Lin z5314168
Eric Phung z5234001
Bavan Manamohan z5208542
Saalai Kaaviya Salai Manimudian z5260340
Nitharshni Chennai Kumaravel z5255563

Table of Contents

Github Repo

API

Web Application

Symptom-Disease Matching

Locating Nearby Doctors/Chemists

Home Screen

Side Bar

Github Repo

Our github repo is located at:

https://github.com/DennisLin27/SENG3011_-Thumbnails-

API

Overview: This API is able to search through a database containing disease related information scraped from the CDC website and return the required JSON files that match the user's inputs.

- Feature 1: When date range, location and/or keywords are provided:
  - User Story:
    - As a user, I would like a list of all the disease outbreaks available which match the date range, location and/or keywords that I have provided so that I am able to learn about each particular disease outbreak out there that corresponds to my needs.
  - Use Case and Persona Description:
    - Use Case: User enters the url of our application, then enters the parameters: start_date, end_date, keyterms and location, and '&' between each parameter (e.g. /findstart_date=2010-03-04T00:00:00&end_date=2022-1-05T00:00:00&location=new york&keyterms=deli) The results will be displayed to the user after being entered.
    - Persona Description: Jason, aged 25 is an office worker living in Shanghai who is aware of the Covid-19 situation in his area. Jason needs to make sure his aged family is staying away from outbreak areas so they can stay safe from Covid-19. Hence Jason is able to to use the location of outbreak areas, the duration and time of the outbreak to locate the outbreak areas and gather news about Covid-19.
  - Final Implementation:
    - When provided with a date range, location and the keyword of Listeria, the API was able to generate a request URL, get the matching information and present it in the response body as shown below:

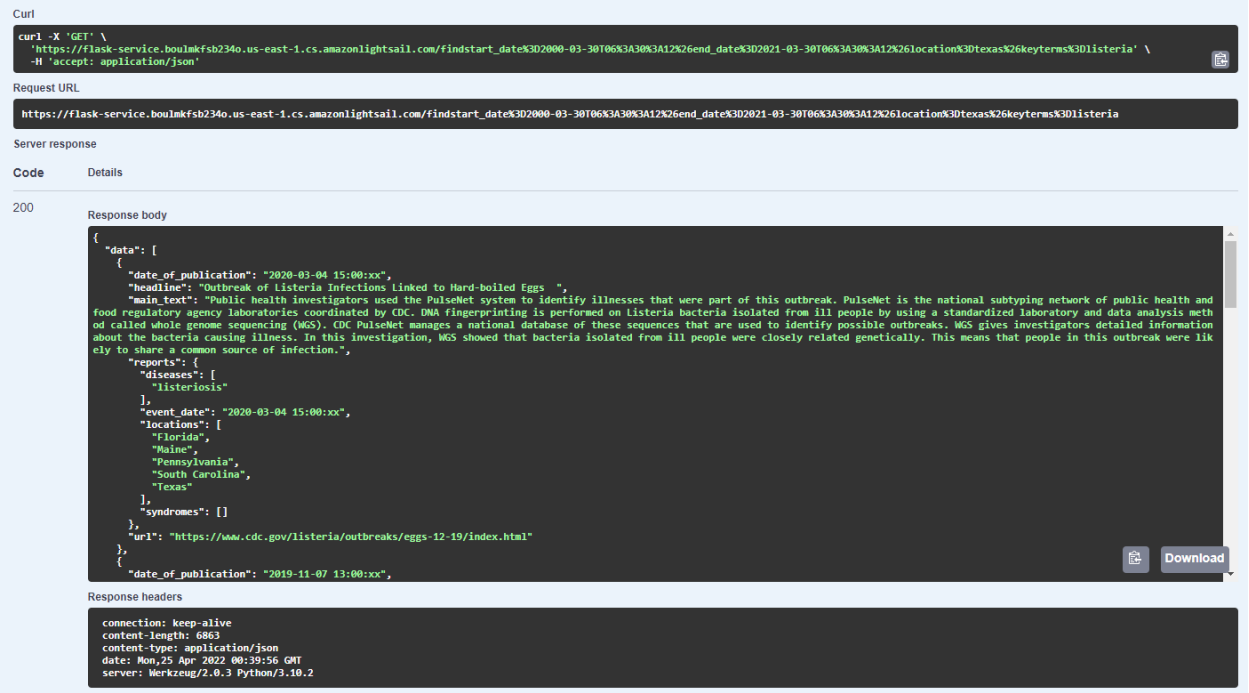

Figure 1: Displaying Feature 1

- Feature 2: When no parameters are provided
  - User Story:
    - As a user, I would like a list of all the disease outbreaks available so that I can get to know about the types of diseases around the world.
  - Use Case and Persona Description:
    - Use Case: User enters the url of our application in the browser, then follows by /findAll. No parameter needs to be entered. A list of disease reports will be displayed to the user.
    - Persona Description: Raymond is a hospital worker who needs to know about currently outbreak diseases. Raymond is able to view a list of disease outbreaks to get to know the types of diseases that he needs to pay attention to.
  - Final Implementation:
    - When no parameters are given, the API creates a request URL to find everything available in the database and display it in the required format in the response body.

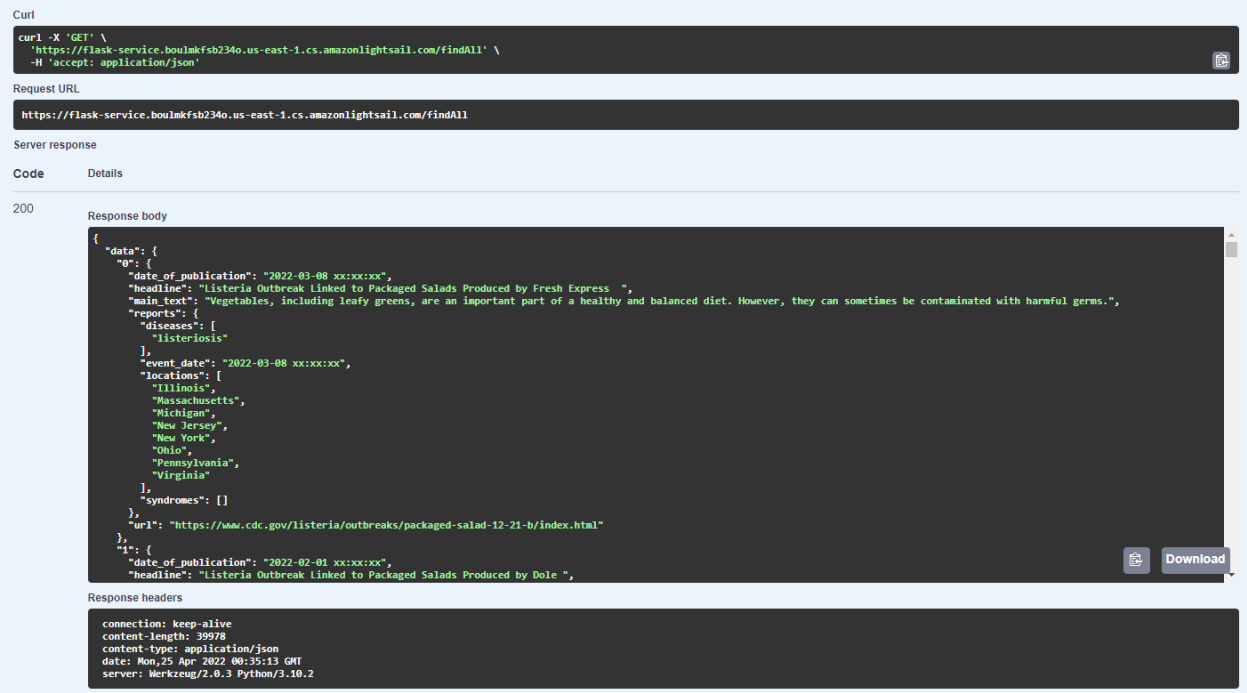

Figure 2: Displaying Feature 2

- Feature 3: When only location is provided
  - <u>User Story:</u>
    - As a user, I would like a list of all the disease outbreaks that are present in a particular location so that I am aware of the diseases I may encounter if I ever decide to go there.
  - <u>Use Case and Persona Description:</u>
    - <u>Use Case:</u> User enters the url of our application in the browser, followed by /find/location, and then followed by the location they want to find information about. A list of disease outbreaks will be displayed based on the location.
    - <u>Persona Description:</u> Jason (mentioned above) is able to get an idea of what diseases are currently in outbreak around the city of Shanghai so he can prepare and make sure that he is safe to go out.
  - <u>Final Implementation:</u>
    - When only the location is provided, the API should create a request URL and return all the matching reports from that location. Unfortunately this is not working properly in our API at the moment, but was working during initial submission.

- Feature 4: When only key terms is provided
    - User Story:
        - As a user, I would like a list of all the disease outbreaks available based on the specific key words so that I can learn about the different outbreaks around the world that are associated with them.
    - Use Case and Persona Description:
        - Use Case: User enters the url of our application in the browser, followed by /find/keyterms, and then followed by keyterms. When there are more than one key term, they are separated by ",". A list of disease outbreaks will be displayed based on the keyterms.
        - Persona Description: Raymond (mentioned above) needs to know about all the news and outbreak reports of Covid-19, Therefore, he entered Covid-19 as a key term and received information all about Covid-19.
    - Final Implementation:
        - When the user enters only key terms the API should create a request URL and return all the matching reports for that key term. Unfortunately this is not working properly in our API at the moment, but was working during initial submission.

- Feature 5: When only date range is provided:
    - User Story:
        - As a user, I would like a list of all the disease outbreaks that were present within a given date range so that I can find out if I would have had any exposure to them while travelling during that time.
    - Use Case and Persona Description:
        - Use Case: User enters the url of our application in the browser, followed by /find/date, and then followed by start_date=<date1>&end_date<date2>, where date1 is the beginning of the date range and date2 is the end of the date range. A list of disease outbreaks will be displayed based on the location.
        - Persona Description: Olivia is a 21 yr old student studying at UNSW. For her summer break, she backpacked across Europe. Just to make sure, she wants to check if there were

any disease outbreaks she may have come in contact with during her holiday.

- Final Implementation:
  - When provided with a date range, the API was able to generate a request URL with this specific time period and gets all the information stored in the database that corresponds to it.

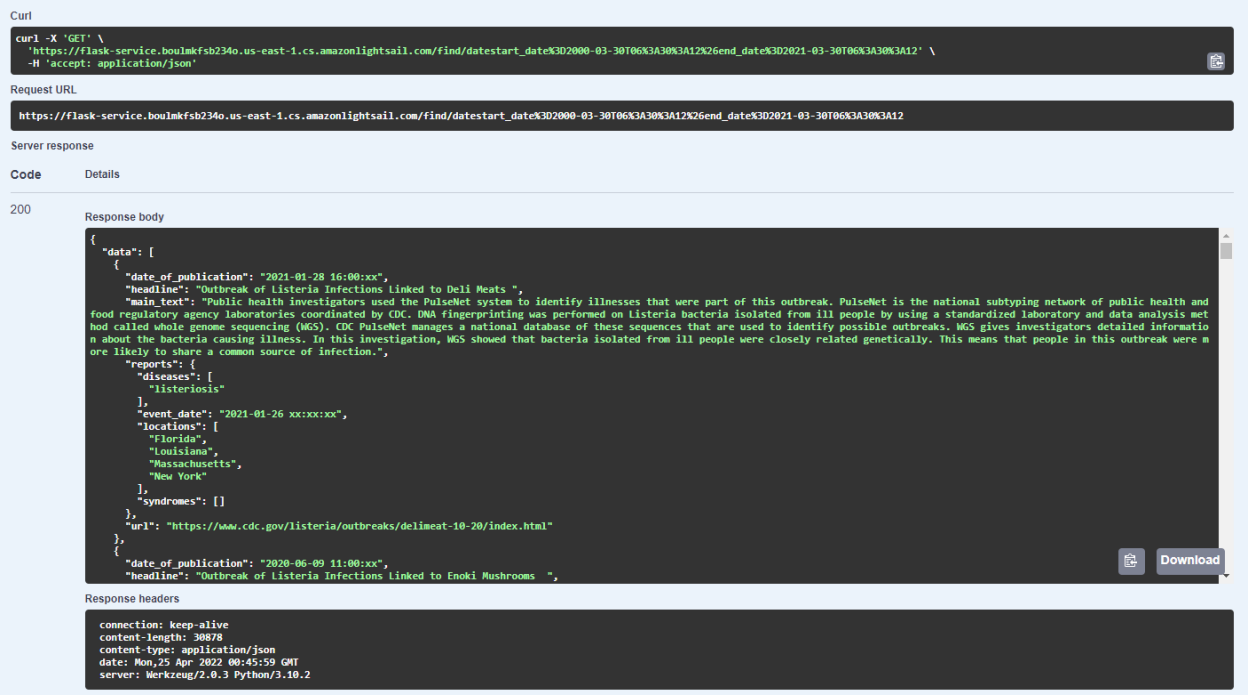

Figure 3: Displaying Feature 5

Web Application

Symptom-Disease Matching

Overview: This feature is essentially a form with a list of questions that collects information such as age group, symptoms, location, and additional information about the user. The app then collates this all together and returns the possible diseases or illnesses they might have and displays it from most to least likelihood.

- Feature 1: Questionnaire
  - User Story:
    - As a user, I would like a questionnaire that will collect all the information that is necessary from me so that it can

find out the possible diseases/illnesses that I may have from most to least likelihood.

- Use Case and Persona Description:
  - Use Case:
    - User opens our application via URL
    - Select Symptom Checker on the Home Page.
    - User will then be directed into a 6 question questionnaire. In the first page, age group buttons are displayed (0-10 years, 10-20 years, 20-35 years, 35-60 years, above 60). The user picks one of these and clicks on the right arrow button at the bottom right to go to the next page.
    - The second page is select country page, there is a drop down box and the user needs to select one of the correct countries from it, then click on the right arrow button to go to the next page.
    - The third page is similar to the second where the only difference is selecting the city instead of country.
    - The fourth page displays a list of symptoms with checkboxes on the right. The user can select multiple symptoms by ticking the checkbox on the right, then click on the right arrow to go to the next page.
    - In the fifth page, the user selects one of the five symptom lengths, then clicks on the right arrow to go to the next page.
    - The sixth page displays a textbox for additional information where the user can enter additional info and click on the submit button. This directs the user to the final page which is the result page.
    - Users are able to view their progression via progression bar at the bottom on each page where each dot represents one question being answered.
  - Persona Description: George is 50 years old and is worried about his health. George enters his age, location, any symptoms he has and additional notes about the symptoms. He receives a health report of diseases that he may have and how likely they are, and is relieved to find that it is most likely indigestion.
- Final Implementation:
  - When the user clicks on the ‘Symptom Checker’ button on the homescreen, they will be directed to this questionnaire,

which will collect all the required information from the user. The progression through the questionnaire is also displayed at the bottom using the progression bar.

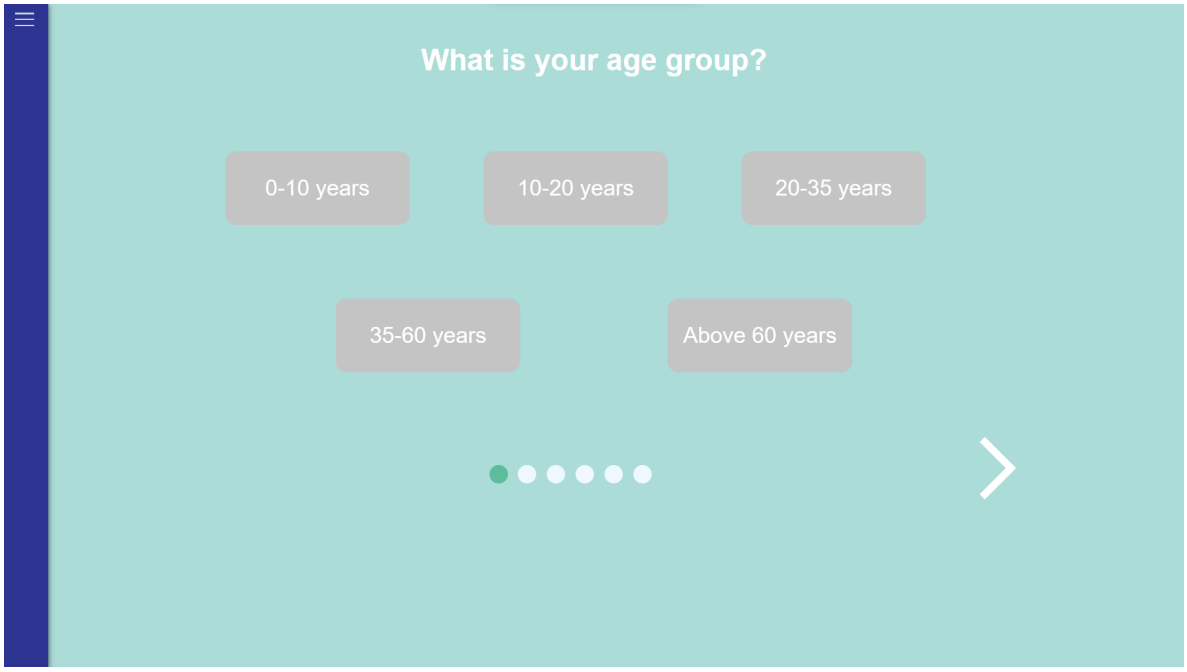

Figure 4: Displaying 'Age' page

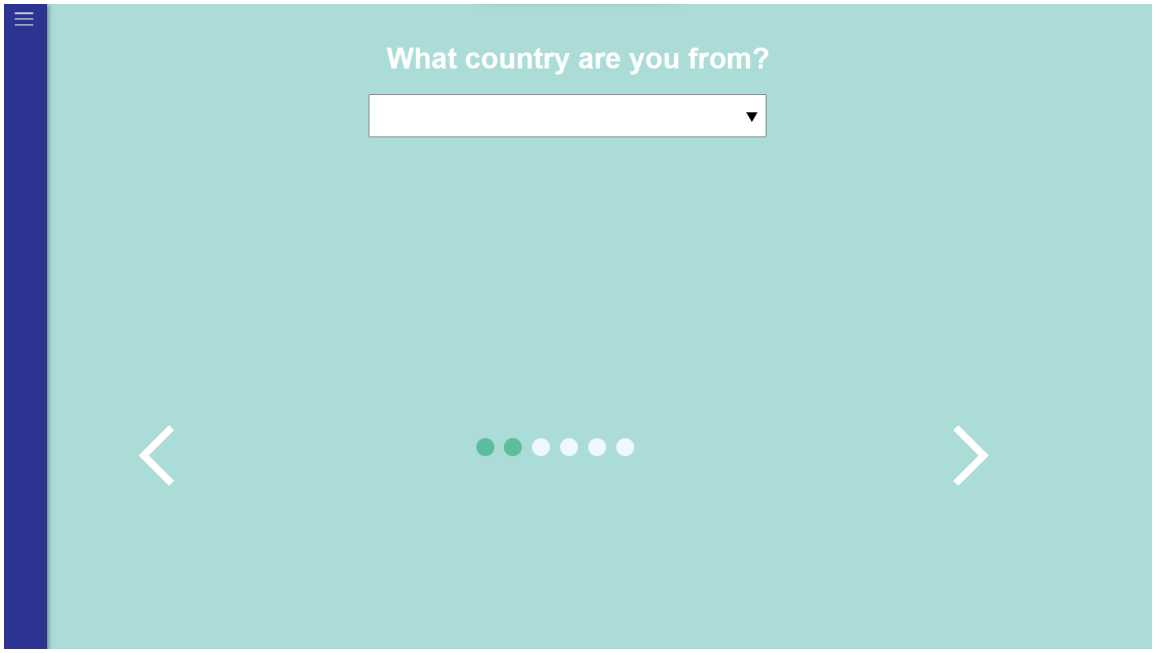

Figure 5: Displaying 'Country' page

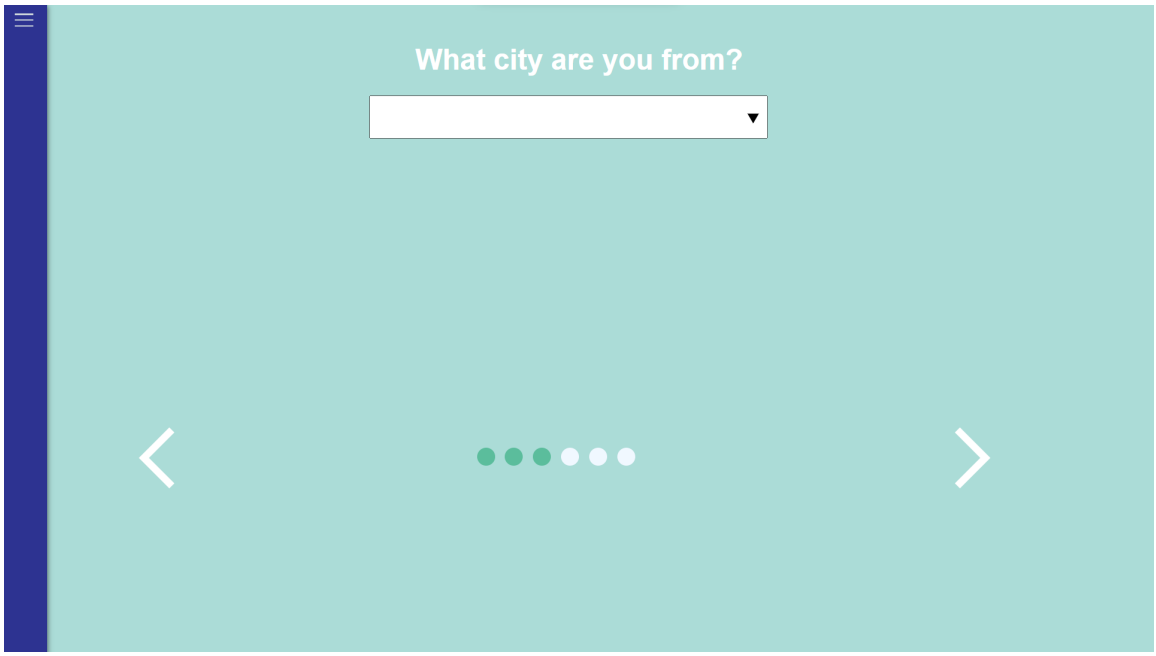

Figure 6: Displaying 'City' page

Figure 7: Displaying 'Symptoms' page

Figure 8: Displaying 'Symptoms Length' page

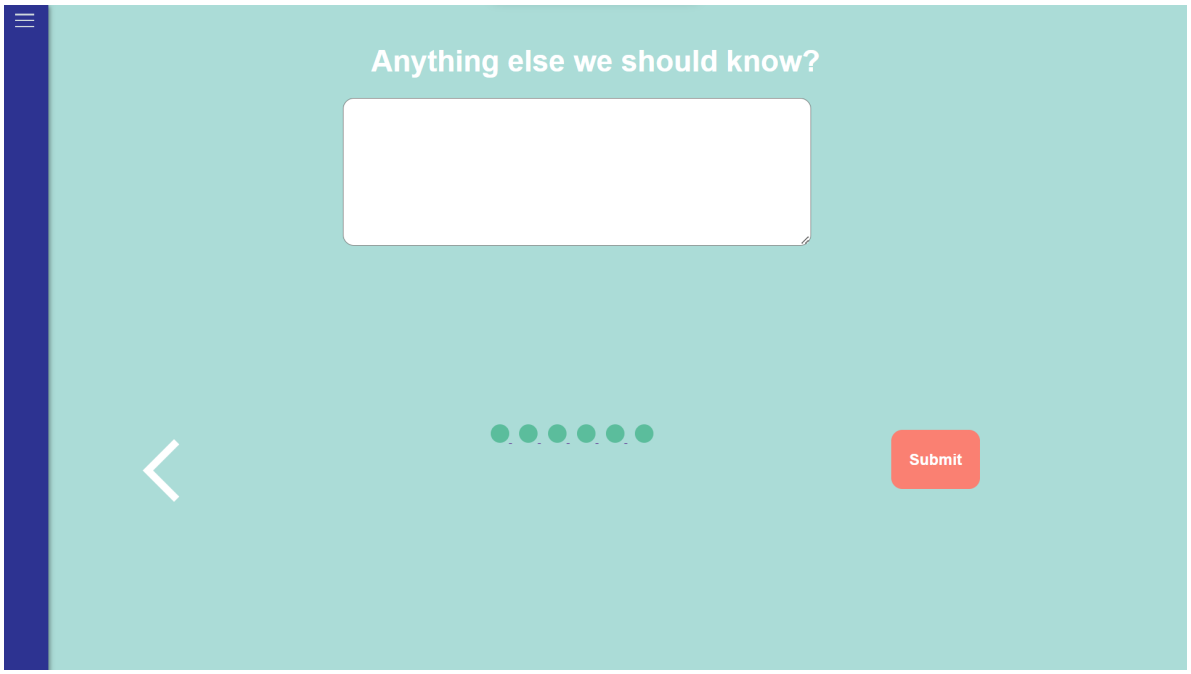

Figure 9: Displaying the 'Additional Info' page

- Feature 2: Report Page:
  - <u>User Story:</u>
    - As a user, I would like a Report Page that lists out all the different illnesses I may have so that I am able to find out

about treatments, remedies and if I need to see immediate help from a health professional.

- Use Case and Persona Description:
  - Use Case: After entering all the questions from above, the user will be displayed with this result page. By clicking on the possible symptoms, there will be a collapsable table which displays details about the disease, symptoms, treatments etc.
  - Persona Description: Based on the results, George (mentioned above) is able to identify diseases he might have that he urgently needs to see a health professional.
- Final Implementation:
  - This is how the matching diseases and illnesses will be shown to the user. When they click on an option, they will be able to access information on that type of disease and even select website links that will redirect them to the WHO website to find out more information if need be.

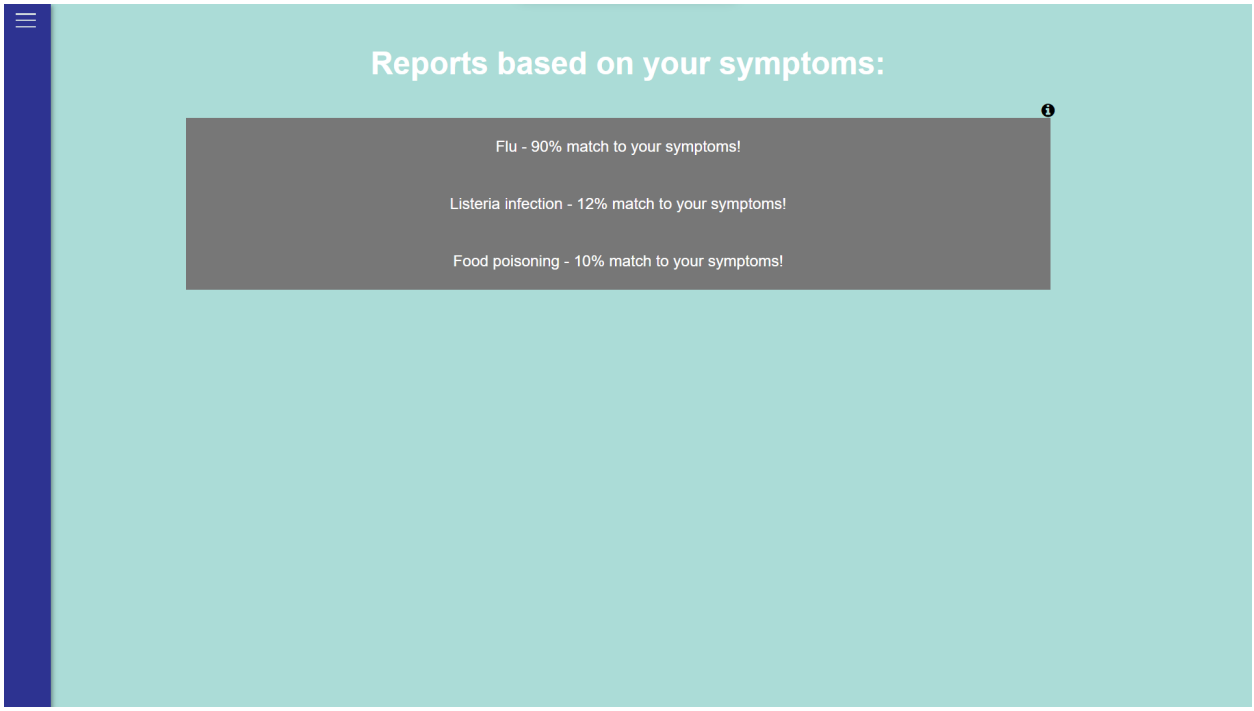

Figure 10: Displaying generated disease reports

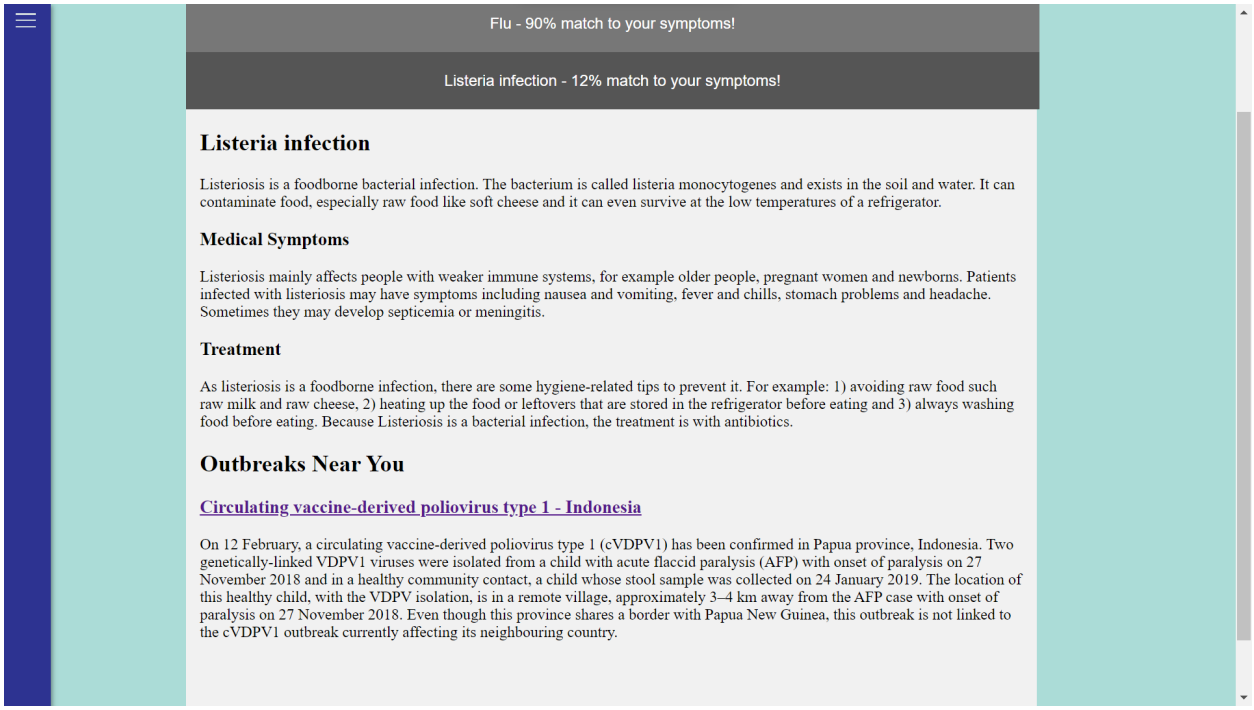

Figure 11: Displaying individual disease report

Figure 12: Displaying URL redirected

Locating Nearby Doctors/Chemists

Overview: These two, separate features allows users to see the nearby doctors or chemists in their area, so that they are able to see which ones is the most convenient option and go there when need be.

- Feature 1: Map
    - User Story:
        - As a user, I would like a map view that shows all the doctors/chemists to me in a visual manner so that I can easily see the different clinics/shops within 3km radius distance from my current location.
    - Use Case and Persona Description:
        - Use Case: The user can either click on ‘Find a Doctor’/’Find a Chemist’ button on the home page, or by selecting the button on the hamburger menu. The User will be directed to the page and by entering a location in the search box, the map will show nearby doctors/chemists with red markers on them. The User can zoom in/out of the map by using ctrl + scroll wheel, or clicking on the +/- button. There is a full screen button on the top right that by clicking, enlarge the map to full screen.
        - Persona Description: George (mentioned above) wants to know where the nearby doctors/chemists are so he can get to one in the shortest amount of time.
    - Final Implementation:
        - When “George Street” is inputted, the website provides a visual display of all the doctors/ chemists (depending on which feature the user is using) within a 3 km radius from that location.

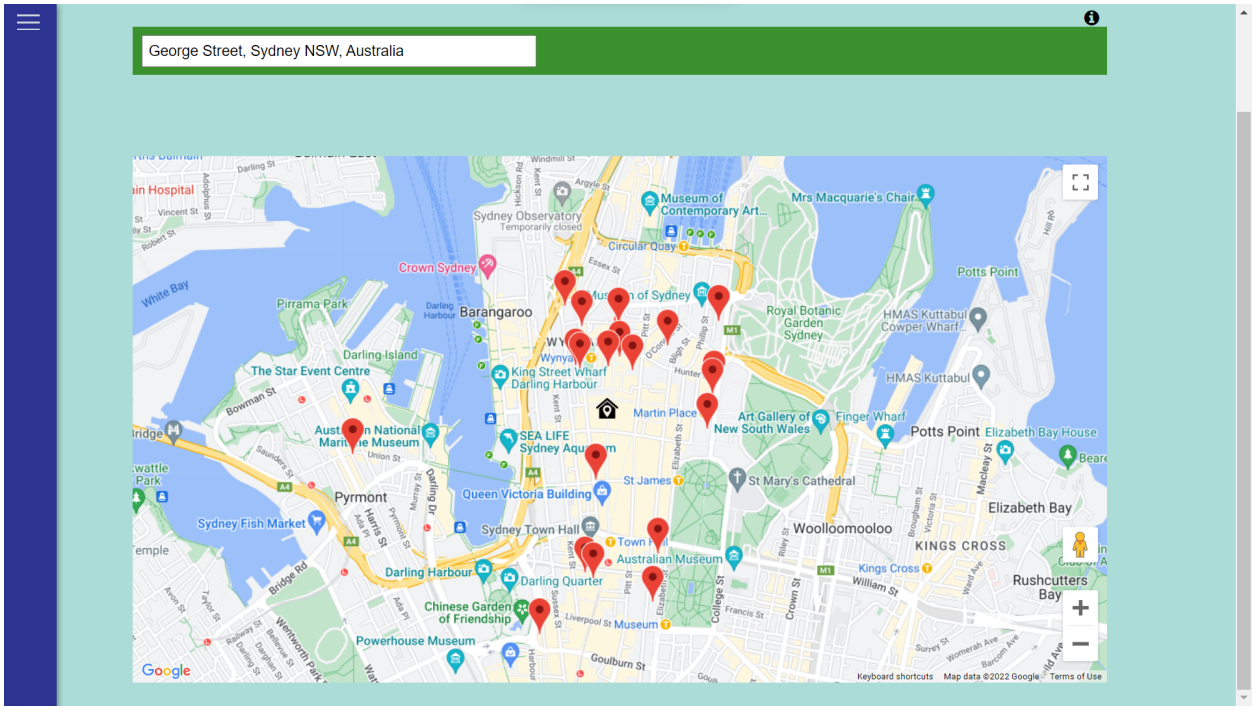

Figure 13: Displaying Location tags on map

- Feature 2: Location Tags
  - <u>User Story:</u>
    - As a user, I would like to see location tags on the map view visualisation so that I am able to select and and look at specific information including the name and address of the location.
  - <u>Use Case and Persona Description:</u>
    - <u>Use Case:</u> By clicking on one of the red markers, the name and address of the doctor/chemist will pop up. The user is able to close the window by clicking on the cross.
    - <u>Persona Description:</u> George (mentioned above) is able to see all the locations of doctors/chemists, he clicks on one of them and is able to get the name and the address so that he can search up their number to book an appointment.
  - <u>Final Implementation:</u>
    - When a particular location tag is selected from the many that is displayed within the desired area, the name and the address of that location is displayed for the user.

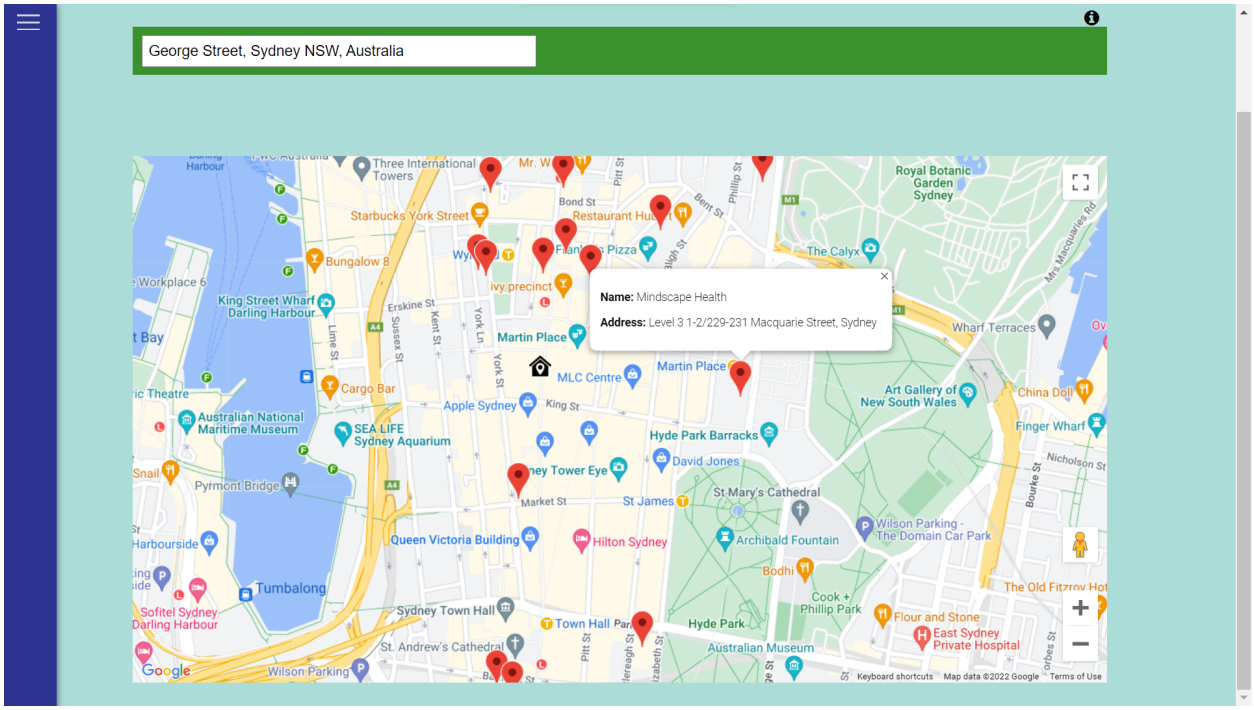

Figure 14: Displaying Location tags on map

- Feature 3: Street View
  - <u>User Story:</u>
    - As a user, I would like a street view feature available in the map visualisation so that I am able to see how the store that I want to go to looks and easily recognise it when I go there.
  - <u>Use Case and Persona Description:</u>
    - <u>Use Case:</u> On the same page of Find a Doctor/Chemist, by clicking on the yellow person icon, the user is able to see the street view of the location entered. The red marker of the location will also be displayed in the street view mode. Users are able to look around the area by dragging with their mouse.
    - <u>Persona Description:</u> George (mentioned above) doesn't know what the clinic/store looked like, so he uses Street View and is able to recognise it when he arrives.
  - <u>Final Implementation:</u>
    - When the little yellow person icon is selected from the bottom right corner of the map, dragged and placed on the location tag of interest, this will provide a street view of that location:

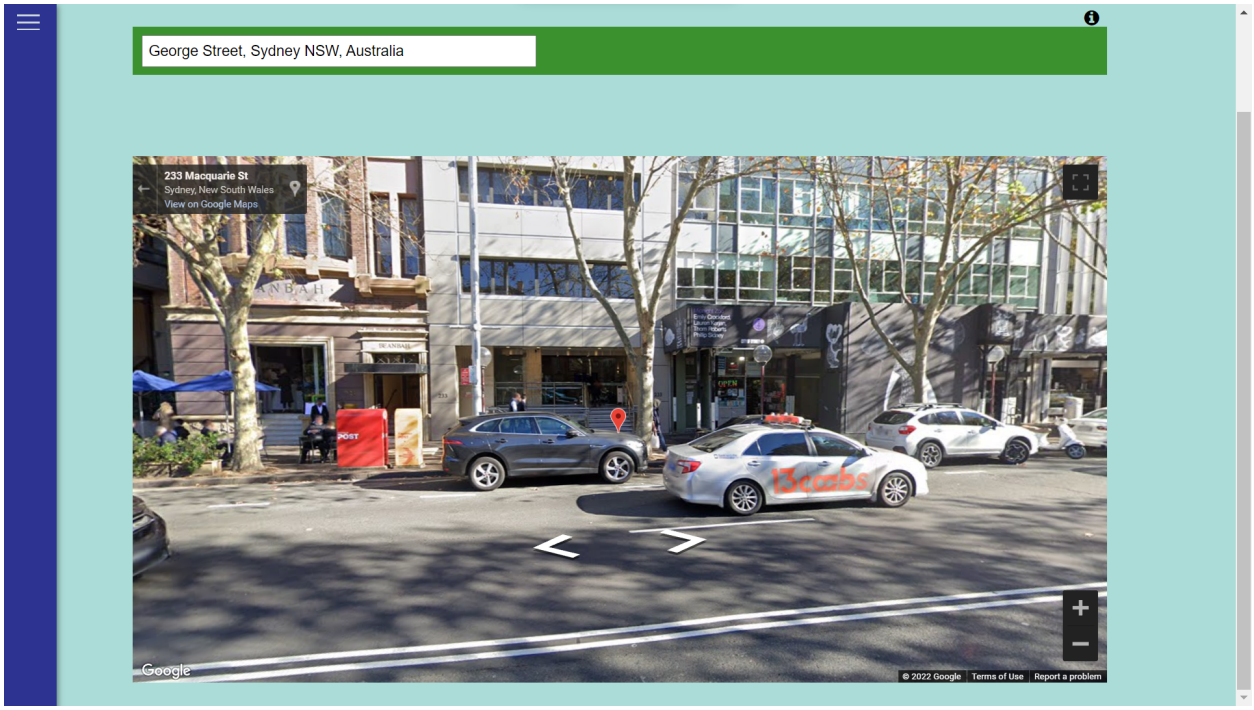

Figure 15: Displaying Street view

Home Screen

Overview: This is the main screen that users will see when they open our web application. It will display all the feature out application contains

- Feature 1: Home Screen
    - User Story:
        - As a user, I would like a home screen so that I am able to see all the features that this application contains and use them
    - Use Case and Persona Description:
        - Use Case: This is the first page after the user opens our application via url. By selecting one of the three buttons on Home Page, the user will be directed to the page related.
        - Persona Description: Raymond wants to see what are the available features in the app so he can decide which one to use.
    - Final Implementation:
        - When the web application is opened, the user will be able to see this screen so that they can select the feature of their desire.

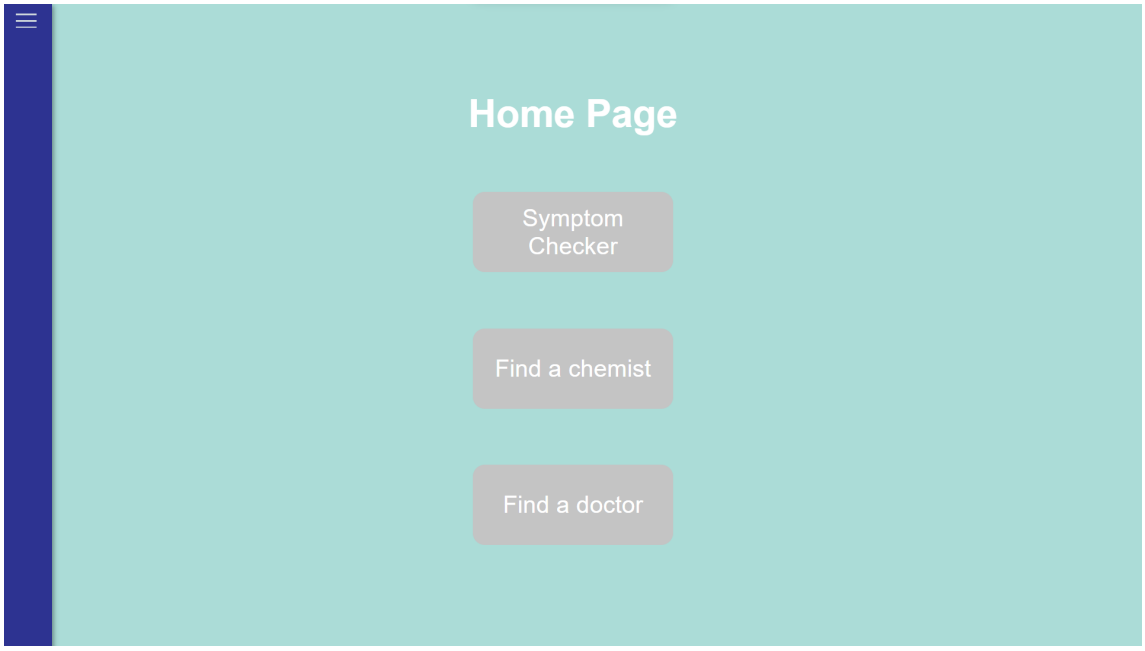

Figure 16: Displaying Home Page

Side Bar

Overview: This feature displays all the features available on the right side all the pages of our web application

- User Story:
  - As a user, I would like a sidebar available on all the webpages so that I can easily move from one feature to another without having to go back to the homepage every single time.
- Use Case and Persona Description:
  - Use Case: By clicking on the hamburger menu on the left, buttons that navigate to pages are displayed. Clicking on one of the buttons, the user will be directed to that page. The user can close the menu by clicking on the cross once opened.
  - Persona Description: George (mentioned above) wants to go back to the home page in the middle of a survey, so he uses the sidebar to navigate to the home page.
- Final Implementation:
  - On each page of the web application, the blue section in the far left of this page will be displayed. Hence when need be, they will be able to easily move from one page to another

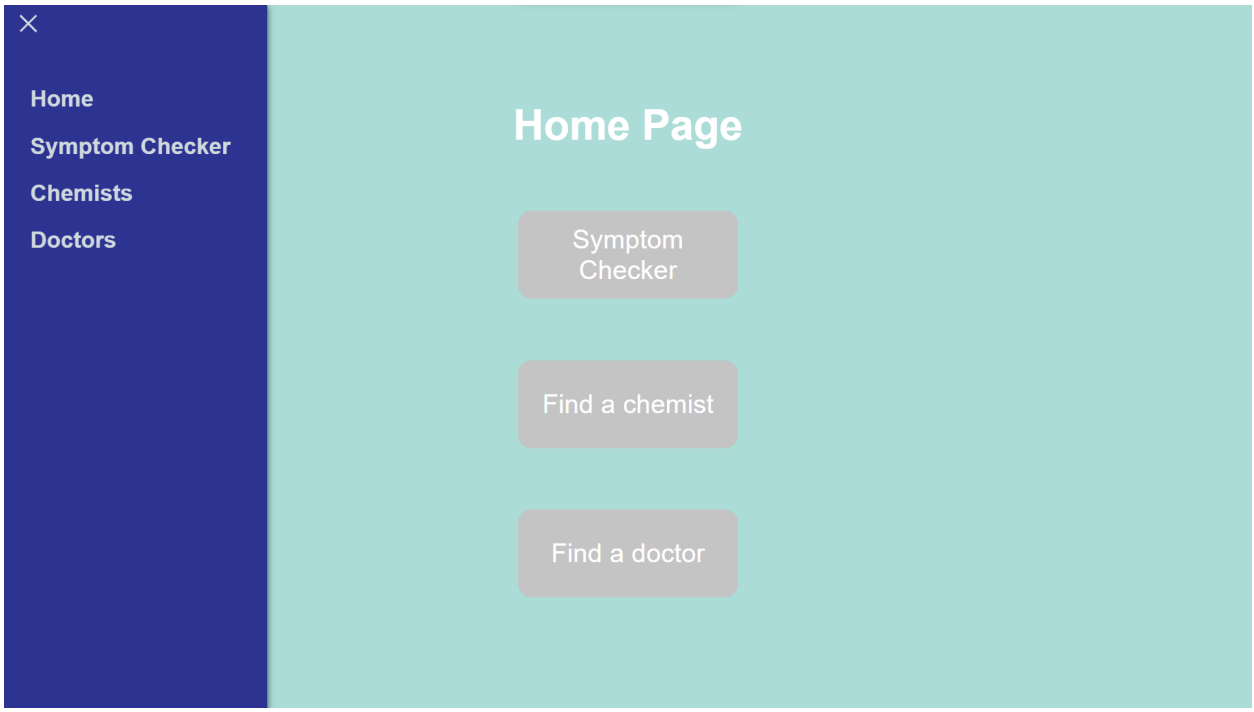

Figure 17: Displaying Home Page and Menu