FIT3162: Advanced Computer Science Project
Final Project Report

Project Supervisor: Dr. Soon Lay Ki

Group: MCS15
Ooi Yi Sen – 30720699
Chan Wai Han - 31555373
Nawwaf Ali - 31261949
Yeonsoo Kim - 29584612

Word Count: 9900 words

Table of Contents

1.0 Introduction

Since the outbreak of COVID-19, scientists have carried out experiments and research on COVID-19. Vaccines, antibodies, oral antiviral medicine, and cell-based treatments were among the focus areas of their research. The outcome of their research has been sought by the public as well as the governments to control the outbreak. In addition, the general public has not shied away from contributing their thoughts and experiences too. These have resulted in a large burden for the readers seeking to maintain up-to-date knowledge on COVID-19. In fact, some of the information provided is understandable by the general public. This project was hence motivated to design and implement a Question-Answering system in the form a COVID-19 chatbot.

The objectives for this project were to provide:

1. Software codes of the system, including both the front-end website UI and the back-end system.
    - All the logic and algorithms used to process the user's queries and provide a simple user-friendly interface for consumers to easily access the COVID Q&A system.

2. Database server system
    - To store all the data from our dataset in the database which will be used to answer the user's queries.

3. Design Rationale documents
    - To help other developers to know the logic of our program and to explain our choices in selecting the algorithms & software.

Our implementation of the COVID-19 chatbot is in the form a website, with multiple pages consisting of the home page, the actual COVID-19 chatbot, a self browsing repository so that the user can look through the questions in our database themselves rather than querying and finally an about us page. The back-end for this website (the database) was implemented using azure sql database.

2.0 Project Background

For the past few years, the coronavirus had become an important and urgent threat to global health. After increasing by 13-fold in the number of cases outside of its country of origin, China, as well as tripling the number of affected countries, the WHO Director-General publicly announced that the coronavirus would be characterized as a pandemic (*WHO Director-General's opening remarks at the media briefing on COVID-19 - 11 March 2020*, 2020). Coronavirus disease is an infectious disease caused by the SARS-CoV-2 virus. Many people might not have the appropriate knowledge, acceptance, and perception of Covid. Despite professionals such as health workers or researchers publishing resources online, or the public sharing their experiences of such encounters, the amount of information is considerably overwhelming. Therefore, our project aims to develop a system that can compile and automatically answer the various types of questions asked by the user that is related to coronavirus. Not only will this assist the scientific and medical communities with their queries, but also the general public in terms of staying up-to-date with the latest information regarding coronavirus.

3.0 Literature Review

Q&A system

3.1 Introduction

There are so many kinds of Question Answering Systems Nowadays. In this literature review, we will focus on how the current automatic question answering systems are working so that our team can get the basic ideas on how to construct our own question answer system for the project. As all the question answering systems have their own data processing and answering methods that suit its purpose, we will focus on the data processing and answering methods that are closely relevant to our project.

Moreover, as our project is also related to the front end to be used by users, we also investigated how to create a good UI/UX. No matter how good back-end programs and algorithms are used, if users who use the website feel uncomfortable and cannot easily use the program, good back-end programs and algorithms are meaningless.

Due to the fact that many scientific terms and technical terms are used in the articles, we will simplify the terms and the detailed methods/algorithms so that we can easily understand and identify how the programmes work.

3.2 Question Answering System Approaches

To understand what approaches we should use for our automated Q&A system, we decided to have a look for the Q&A system approaches. According to the article by Dwivedi, S. K., & Singh, V. (2013), there are 3 main approaches in automated- Q&A systems.

Discriminative Model Approach

This approach would be the way this article 'Linguistic kernels for answering re-ranking in question answering systems.' by MOSCHITTI (Moschitti, 2021) writes about moving away from typical approaches that use unsupervised methods that involve computing the similarity between query and answer in terms of leical, syntactic,semantic or logic representation (Moschitti, 2021). Instead they studied supervised discriminative models that learn to select answers from examples of question and answer pairs, where the representation of the pair is implicitly provided by kernel combinations applied to each of its components (Moschitti, 2021).

They found evidence to support the exploitation of advanced linguistic information by using powerful discriminative models such as SVMs and effective feature engineering techniques such as kernel methods in challenging natural language tasks.

A drawback to this approach if we follow Moschitti's article is that we would require a question to have multiple ordered candidates answers assigned to a question as training instances.We can take this article into account should we find it a more feasible approach then the others.

Linguistic Approach

Linguistic Approach is a question answering system that requires understanding of natural language text, linguistics and common knowledge (Dwivedi, S. K., & Singh, V.,2013). So, this Q&A system is often based on Natural Language Processing (NLP) logic and knowledge base due to its characteristics. Linguistic techniques are used in this approach, such as, tokenization, POS tagging and parsing, to pre-process the user's query and get the related answers from the database.

An advantage of the linguistic approach is that it can provide a situation-specific answer to the user. However, this approach has many limitations as it is very time-consuming to build an appropriate and sufficient amount of knowledge bases. Moreover, this approach cannot deal with the domain that is out-of-bound of the system if the knowledge is not stored in the structured database. So, most of the limitations are due to its NLP-based logic.

Statistical Approach

Statistical Approach refers to the use of statistics to learn from examples. It means to collect observations, study and digest them in order to infer general rules or concepts that can be applied to new, unseen observations (*Statistical pattern recognition*., 2019) and this approach is independent of structured query languages and can formulate queries in natural language form. (Dwivedi, S. K., & Singh, V., 2013)

The biggest advantages of this approach is that it can produce the best results between other approaches once it has a sufficient amount of data to train the Q&A system and this approach can deal with the unseen questions based on the statistics learned in the system.

However, the disadvantage of this approach is that it treats each term independently and fails to identify linguistic features for combination of words or phrases.
(Dwivedi, S. K., & Singh, V., 2013)

Pattern Matching Approach

This approach uses the expressive power of text patterns to replace the sophisticated processing involved in other competing approaches. (Dwivedi, S. K., & Singh, V., 2013). To simplify, Pattern Matching Approach uses pre-defined patterns in the back-end system to find the matching patterns from the user's questions.

Advantage of this approach is that it is simple to make a pattern matching based Q&A system compared to the other approaches as it requires relatively short time to make short-medium sized answering systems and as it does not require complex systems to build or maintain.

There are two types of the pattern matching approaches, which are surface pattern based and template based approaches. Most of the patterns matching QA systems use the surface text patterns while some of them also rely on templates for response generation.
(Dwivedi, S. K., & Singh, V., 2013).

Surface Pattern based approach extracts answers from the surface structure of the retrieved documents by relying on an extensive list of patterns. Answers to a question are identified on the basis of similarity between their reflecting patterns having certain semantics.

A Template based approach makes use of preformatted patterns for questions. The focus of this approach is more on illustration rather than interpretation of questions and answers.
(Dwivedi, S. K., & Singh, V., 2013).

There are many classification methods we can use.In the Article 'Traditional Machine Learning Models and Bidirectional Encoder Representations From Transformer(BERT)- Based Automatic Classification of Tweets About Eating Disorder: Algorithm Development and Validation Study' (Benítez-Andrades, 2022), writes about methods such as random forest, recurrent neural networks, bidirectional long short-term memory networks and pretrained bidirectional encoder representations.
Bidirectional long short-term memory and bidirectional encoder representations from transformers seem to be the most promising model for natural language processing (Benítez-Andrades, 2022).

Classification Method	Description
Random Forest	● Random forest models are constructed from a set of decision trees, which are usually trained with a method called bagging, to take advantage of the independence between simple algorithms,since error can be greatly reduced by averaging outputs of the simple models. (Benítez-Andrades, 2022). ● Several decision trees are built and fused in order to obtain a more stable and accurate prediction. (Benítez-Andrades, 2022)
Recurrent Neural Network(RNN)	● Type of neural network where a temporal sequence that contains a directed graph made up of connections between different nodes is defined. (Benítez-Andrades, 2022) ● These networks have the capacity to show a dynamic temporal. (Benítez-Andrades, 2022) ● Derived from feedforward neural networks,which have the ability to use memory to process input sequences of varying length. (Benítez-Andrades, 2022)
Bidirectional Long Short-Term Memory	● Bidirectional long short-term memory networks are constructed from 2 long short-term memory modules that, at each time step, take past and future states into account to produce the input. (Benítez-Andrades, 2022)
Bidirectional Encoder Representation from Transformer-Based Models(BERT)	● Bidirectional encoder representation is not a model itself. It can be considered a 'language understanding' model. (Benítez-Andrades, 2022). ● A neural network is trained to learn a language, similar to

	transfer learning in computer vision neural networks, and follows linguistic representation in a bidirectional way, looking at the words both after and before each words. (Benítez-Andrades, 2022)

There are similarities in the sense of ranking the category of tweets mention in the Article 'Traditional Machine Learning Models and Bidirectional Encoder Representations From Transformer (BERT)-Based Automatic Classification of Tweets About Eating Disorder: Algorithm Development and Validation Study' (Benítez-Andrades, 2022) in the sense that we would need properly to label what we would consider a high level answer and what is a low level answer in the same way they label the categories of a tweet.

3.3 Q&A System Structure

Based on the article of Pundge, A. M., Khillare, S. A., & Mahender, C. N. (2016), The Q&A system is made up of four modules, which are Question Processing module, Document Processing Module, Paragraph Extraction Module, and Answer Extraction module.

Question Processing Module

The question processing module pre-process the user's questions into system-friendly queries (Natural Language queries) so that the question can be analysed and the answer type can be determined by the system.

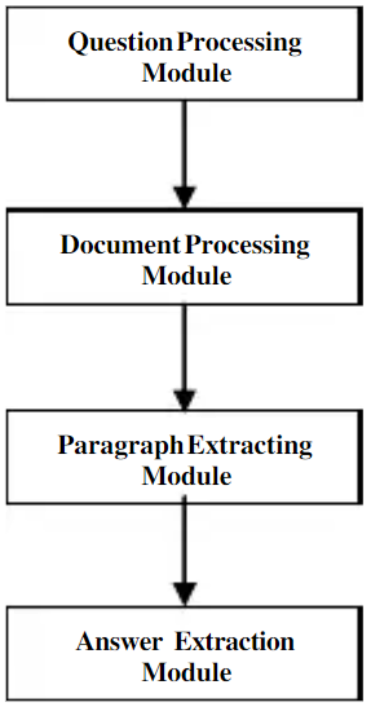

Question Processing Module
Document Processing Module
Paragraph Extracting Module
Answer Extraction Module

Figure 2.3.1 : Basic Structure of Q&A System
(Pundge, A. M., Khillare, S. A., & Mahender, C. N., 2016)

Document Processing Module

The document processing module is an Information retrieval module that focuses on gathering relevant documents. (Lalithnarayan, C., 2020). It consists of a query generation algorithm and text search engine. The query generation algorithm takes an input the user's question and creates a query containing terms likely to appear in documents containing an answer. This query is passed to the text search engine, which uses it to retrieve a set of documents. (Pundge, A. M., Khillare, S. A., & Mahender, C. N., 2016).

Paragraph Extraction Module

In this module, the documents obtained from the previous step are reduced to produce a concise answer. (Lalithnarayan, C., 2020). 'Passage retrieval' algorithms to reduce the amount of text, which is called 'finding passages' in the documents in scientific terms, are used to break the documents into paragraphs or sentences (passages) and select appropriate passages by using scores for each passage and by selecting the passages with the highest scores.

Answer Extraction Module

The module is the final module of the system, which takes the passages selected from the previous Paragraph Extraction Module as input and finally returns the most precise and reformatted answers to the users. Self-learning Q&A system often updates the system by evaluating the answers by itself.

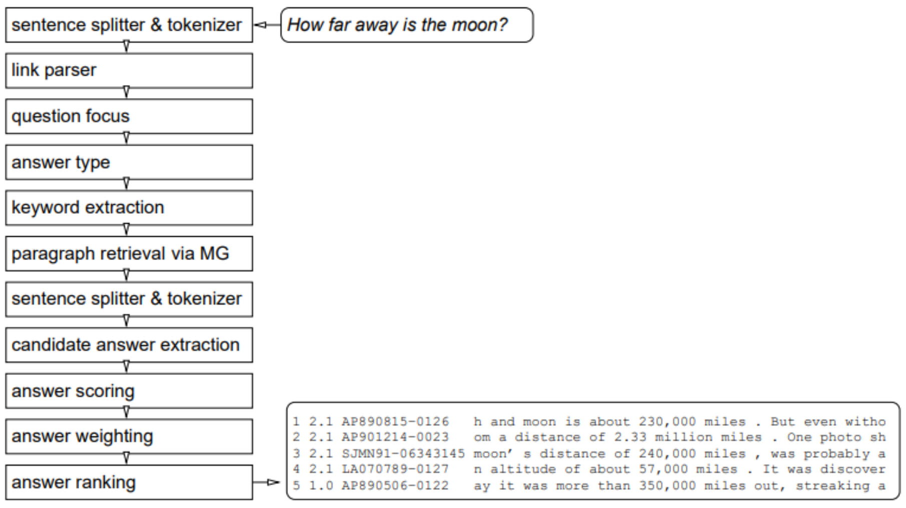
3.4 Real Q&A System's Workings

To have a better understanding of the Q&A system, to decide which approach we should take and to have a detailed plan on how to structure our own Q&A system, we have read a research article of a real Q&A system.

Cooper, R. J., & Ruger, S. M. (2000).
A simple question answering system

Firstly, all the articles available in the database of the programme are stored as a raw content of the article. For example, $ and £ are replaced by words "dollars" and "pounds" to get and store the raw content only.

The actual question processing is executed as a long pipeline of perl modules which use XML, which is mark-up entities or to communicate other information between the modules.


sentence splitter & tokenizer
How far away is the moon?
link parser
question focus
answer type
keyword extraction
paragraph retrieval via MG
sentence splitter & tokenizer
candidate answer extraction
answer scoring
answer weighting
answer ranking

```
1 2.1 AP890815-0126   h and moon is about 230,000 miles . But even witho
2 2.1 AP901214-0023   om a distance of 2.33 million miles . One photo sh
3 2.1 SJMN91-06343145 moon’ s distance of 240,000 miles , was probably a
4 2.1 LA070789-0127   n altitude of about 57,000 miles . It was discover
5 1.0 AP890506-0122   ay it was more than 350,000 miles out, streaking a
```

Figure 2.4.1 : Data Flow of the simple question answering system (Cooper, R. J., & Ruger, S. M.,2000)

So, the actual question will be separated into parts based on the pre-defined patterns or characters such as a question mark or an exclamation mark and will be formatted into XML codes so that the question can be easily digested by the system.
If I take the example of a question from the article, “How far away is the moon?” will be separated into “how”, “far”, “away”, “is”, “the”, “moon”, “?” and the separated parts will be formatted into XML codes with multiple additional data contained for the question and this process is called ‘Link Parser’ in this system.

```
<sentence><t n="1">How</t> <t n="2">far</t> <t n="3">away</t> <t n="4">is</t>
<t n="5">the</t> <t n="6">moon</t><t n="7">?</t><parse><pos n="2" pos="a"/><pos
n="4" pos="v"/><pos n="6" pos="n"/><link name="Xp" l="0" r="7"/><link name="Wq"
l="0" r="2"/><link name="PF" l="2" r="4"/><link name="MVp" l="2" r="3"/><link
name="SIs" l="4" r="6"/><link name="Ds" l="5" r="6"/><link name="RW" l="7"
r="8"/></parse></sentence>
```

Figure 2.4.2 : Example XML codes of the simple question answering system
(Cooper, R. J., & Ruger, S. M.,2000)

Third process is called “Question Focus”, in which the system identifies the critical keywords of a asked question. We can easily know that each ‘when’, ‘who’, ‘where’, ‘whom’ and ‘why’ is a critical keyword of a question. For example, if a user asks “Where is Monash University?”, then, “where” and “University” will be the critical keywords of the question. Then, we can define those critical keywords in the system as back-end logics and add more data to the question. When the system finds the critical keywords, an extra XML element will be added to the existing XML codes, which notifies the system what words are critical. However, the article says that we need to be careful in deciding the critical keywords as there could be questions like “In what city is the US Declaration of Independence located?”, which the question does not start with easy critical keywords.

Next process is called “Answer Type”, in which the system identifies what kind of answers the system should provide to the user. The process is highly dependent on the “Question Focus” process, which is the previous process as the algorithm of this process must be based on the question focus that is identified in the previous process. The article mentions that answer type must be decided with detailed algorithms as the question focus will not simply indicate the purpose of the question. When the answer type is decided, an additional answer type XML element will be added to the original XML codes like the third process.

Then, the system will look for the related articles (as this programme is based on news articles database) in the database, using the answer type and question focus found in the above stages. The look-up of articles will be based on the weightings of the keywords and the most-likely subset articles will be chosen as the result.

The candidate answers from the previous process will be splitted and tokenized for a further processing and the question's answer concept is looked up in WordNet and all of its hyponyms are found. A regular expression is then built by taking the disjunction of those hyponyms and any region of text that matches that regular expression is marked up as a candidate answer. This process is called "Candidate Answer Extraction".
The problem with this process is that all the hyponyms could not be related to the question and questions which require descriptions cannot be easily answered using this process as they must be based on very complex Natural Language Processing (NLP).

Lastly, when the candidate answers are chosen in the system, the answer will be scored based on heuristics methods, which are:

(i) score_comma_3_word
If a comma follows the candidate answer then this score is the number of the three words following the comma that appear in the question
(ii) score_punctuation
Scores one if a punctuation mark immediately follows the candidate and zero otherwise
(iii) score_same_sentence
Computes the number of question words that are in the same sentence as the candidate answer
(iv) score_description_before
If the answer concept being looked for is a description then this score is the number of words immediately preceding the candidate answer that appear in the question
(v) score_description_in
Similar to score question before but counting question words that appear in the candidate answer.

After candidate answers going through these heuristics, all the candidate answers will be paired with (id, score) and this will be added as additional XML elements again. Then, the all answer scores will be combined into a final score based on the weight of each heuristics and using the final score of each candidate answers, the final answer that has the highest final score will be provided to the user.

3.5 Front-end UI Design

Sheniderman's Eight Golden Rules (Wong, E. (2018), Goonawardene, P. (2021))

- <u>Strive for consistency</u>

- Enable frequent users to use shortcuts
- Offer informative feedback
- Design dialogue to yield closure
- Offer simple error handling
- Permit easy reversal of actions
- Support internal locus of control
- Reduce short-term memory load

1. **Strive for consistency**

Strive for consistency means that the developers must use the same design patterns and sequence of actions in similar situations to maintain the consistency. According to Wong. E (2018)’s article, it argues that standardizing the way information is conveyed ensures users to apply knowledge without the need to learn new presentations for the same actions again. Also, Goonawardene, P. (2021) says that failing to maintain the consistency of the UI will increase the users’ cognitive load because the users will be forced to learn something new every time. Both articles say that ‘Strive for Consistency’ plays an important role in allowing users to complete their tasks and achieve their goals easily.

2. **Enable frequent users to use shortcuts**

This principle means that the UI must offer shortcuts to access features. By following this rule, experienced users will save time by using the shortcuts and inexperienced users will be able to use the programme normally without using the shortcuts. So, enabling users to use shortcuts can satisfy both types for users, who are experienced and inexperienced users.
Goonawardene, P. (2021) says that the rule can be satisfied by offering customization of the features and settings to the users as well. It allows users to make their own decisions on how to use the product. So, if a user feels uncomfortable with some actions, the user can customize the features and settings so that they can change the actions that they feel more comfortable with.

3. **Offer Informative Feedback**

‘Offer Informative Feedback’ means that whenever users perform any actions, they must be always informed with appropriate informative feedback so that the users can know where they are at and what is going on at all times. Goonawardene, P. (2021) says that all the feedback that is provided to users must be meaningful, clear, and relevant to the context.

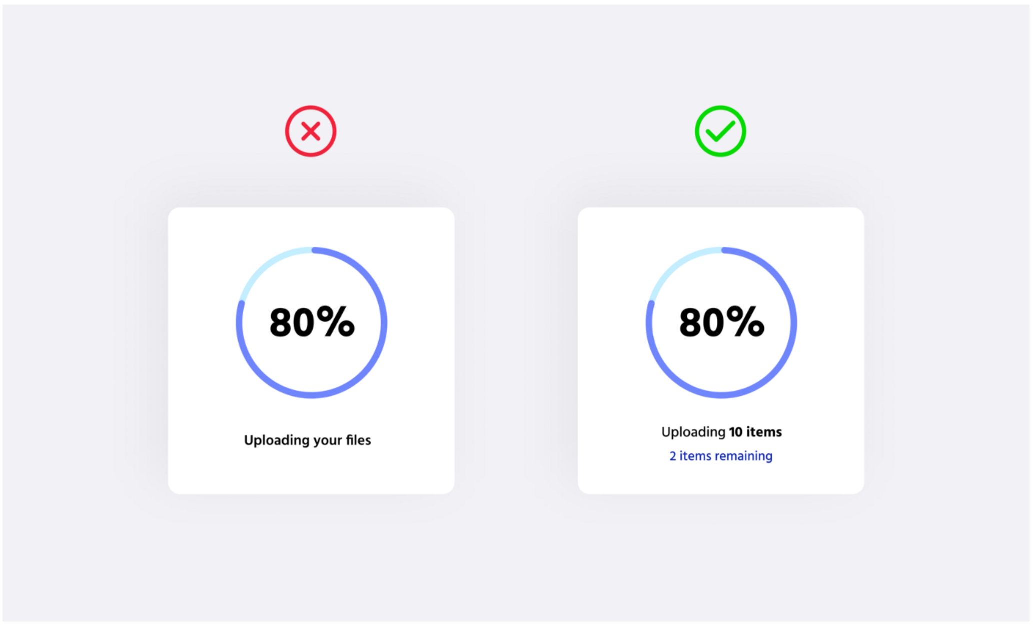


Figure 2.5.1 Example of Informative Feedback (Goonawardene, P. (2021))

4. Design dialogue to yield closure

'Don't keep your users' guessing'. The users must be notified what is going on currently. The rule states that the actions must be consistent, and well organized with a clear beginning, middle and end. When the users complete their action or the processes are completed, the user must be notified that the processes are completed.

5. Offer simple error handling

Products must keep users on the right track and must be error-free as much as possible. However, when unavoidable errors occur when users are interacting with the product, the users must be informed that an error occurred during the interaction and the product must provide simple step-by-step instructions to resolve the problem. Moreover, the rule means that the product must notify the users as simply as possible when the users make errors. For example, if a user misses a text field when filling up a form, the product can just flag the blank text field, without big error messages pop-up that blocks the users from continuing the actions.

6. Permit easy reversal of actions

The users must be able to reverse their actions and there must be a way to reverse any actions. By applying this rule, the users will feel safe since the users know that any errors can be undone and due to the reason, the users will be encouraged to explore unfamiliar options.

7. Support internal locus of control

This rule means that the users must feel like they are in control of the system. The users must be the main initiators of any actions. To follow this rule, any irrelevant information must be removed and users must be provided with a clean and smooth interface so that the users can easily interact with the product.

8. **Reduce Short-term memory load**

Human's memory is limited. Humans are only capable of maintaining around 7 items in the short-term memory. Therefore, the UI / UX design must be as simple as possible to reduce the amount of information in one screen.
'Recognition is better than recall'. Instead of using lengthy words, it is better to use common symbols so that the screen can minimize its content and minimize the users' cognitive load.
For example, we all know that the scissor icon symbolizes the cut function and the garbage bin icon symbolizes the delete function, (Goonawardene, P. (2021)).

Colours in UI Design (Color Matters. 6 Tips on Choosing UI Colors. (2017))

- Learn 60-30-10 rule
- Contrast is a friend
- Consider the psychology of colours
- Don't forget cultural differences
- Strive to colour harmony
- Steal ideas from nature

1. **Learn 60-30-10 rule**

To bring the balance into the composition, the colours should be combined in the proportion of 60%-30%-10%. 60% is the dominant colour that will take the biggest part, 30% is the secondary colour, and 10% is used to make accents.
The proportion is known to be comfortable with the human eyes.

2. **Contrast is a friend**

Colour contrast is a key part of any visual composition. UI with the same colour family will not be noticeable by users and will be hard to draw users' attention. Developers must control the level of contrast so that important features can have high contrast and less important features have low contrast.
However, just applying high contrast is not always the answer. It is because if the contrast is too high, it will be difficult for users to read or scan texts. So, developers should use high contrast to important features only, which should draw attention.

3. Consider the psychology of colours

Human's mind reacts to colours although we usually do not notice it. When we see any colours, our brain gives signals to the endocrine system, which is responsible for the mood and emotions. So, it is important to have knowledge about how each colour influences our mind.

According to Slava Vaniukov. (2020)'s article, 'Red' can symbolize both positive and negative feelings, which are anger, love, confidence and passion.
'Orange' is the colour that represents excitement.
'Yellow' symbolizes joy, happiness and sunlight.
'Blue' means security, trust and safety. The blue colour has positive connotations for most people and it is one of the colours that a lot of companies use for their logos.
'Green' is the colour that gives calmness.
'Black' is related to negative feelings, which is often associated with death and tragedy. However, the black colour often used to represent modern or traditional nowadays.
'White' means wholesomeness, clarity, purity and innocence.

Likewise, all the colours have their own meanings and symbols.

4. Don't forget cultural differences

In the above, we found that the colours have their own meanings and symbols. However, the meanings can have absolutely different meanings in different countries. For example, white represents purity in European countries, however, white means death and sorrow in Asian countries.
So, developers must keep in mind that the meaning of colours can be different in different cultures and need to choose appropriate colours to prevent misunderstandings which could be fatal for a product.

5. Strive for colour harmony

Developers must try to balance the colours in the UI design. The colour harmony is the arrangement of the colours in design and it is important to make the UI to be attractive.

There are different types of the basic colour schemes.
'Monochromatic' is based on one colour with various tones and shades.
'Analogous' applies the colours right next to each other on the colour wheel.
'Complementary' is the mix of colours which aims to produce high contrast.
'Triadic' is based on three colours. One colour becomes the dominant colour and the other colours become accents.

6. Steal ideas from nature

Colour combinations that we can see in nature are close to perfect because the colour harmony is actually based on the natural colour combinations.

3.6 Database design process (Database design basics(n.d))

A properly designed database is essential for completing your objectives when creating your database. So it is important to know how to design a good database

The database design process has the following steps:

- Determine the purpose of your database
- Find and organise the information required
- Divide the information into tables
- Turn information items into columns
- Specify primary keys
- Set up the table relationships
- Refine your design
- Apply the normalisation rules

1. **Determine the purpose of your database**

It is a good idea to write down the purpose of the database on paper – its purpose, how you expect to use it, and who will use it. This will help you have a good idea on what your main objectives will be and you can look back at it when you have problems in your decision making regarding the database.

2. **Find and organise the information required**

Firstly start with organising your current existing information. Examine what your data shows such customer names,addresses,etc.. You can see that each one is a potential column in a table. List out such items of data. Then think about what kind of outputs you want to produce from the database. Design how you would like the outputs to look as this will help identify items you will need in the database.
Key point is to break each piece of information into its smallest useful parts such as in the case of names being broken down into First name and last Name.

3. **Divide the information into tables**

To divide the information into tables, you must choose the major entities or subjects. When designing your database try to record each row or record just once. Otherwise try putting such data into a separate table. Once you have chosen an entity for that table, its columns should only store attributes about that entity only.

4. **Turn information items into columns**

To determine the columns in a table, decide what information you would need to track about the subject or entity being recorded in the table. When selecting your columns make sure you don't include calculated data, as this would be redundant as you could query the data using existing columns to calculate the values you're looking for instead. And always store the information in its smallest logical parts as this would make it easier for you when trying to retrieve information from said table.

5. **Specify primary keys**

Each table should include a column or set of columns that uniquely identifies each row stored in the table usually called a primary key. A primary key cannot change in value and cannot be empty, any column that can become so is not a candidate for a primary key.

6. **Set up the table relationships**

After creating tables you need to link the information together in meaningful ways, this would be by creating relationships between the tables such as:

- One to many relationships
- Many to one relationships
- Many to many relationships

7. **Refine your design**

After creating the tables,fields and relationships you need, then you should populate the tables with sample data. Use this sample data to query and create new records. As you look through your initial database you might find other ways in which you can refine your database further. Such as if there is duplicate data or if there's fields that are empty most of the time or too many records in one table. Such ways are how you could identify and refine the database design.

8. **Apply the normalisation rules**

Finally you can apply data normalisation as the next step in your design process. You can use these said rules to see if your database is structured correctly or not. There are three stages in normalisation, being:

- First normal form
- Second normal form
- Third normal form

Each form comes after the other and cannot just be implemented from the middle and must come one after the other respectively starting from converting the database into the first normal form.

3.7 Conclusion

After having the literature review on the Q&A system, we could get a detailed idea on how to structure our Q&A system and what kind of algorithms and approaches we can use for our project. We found out that all the Q&A systems follow the same step, which is pre-processing the user’s query, retrieving the relevant documents stored in the database and finally getting the appropriate answers and returning the answer to the user.
We also found out that most of the Q&A system makes use of ‘Candidate answers’ and chooses the most correct answer among multiple possible answers, not returning an answer directly from a single document.

Moreover, after reading the research article of a real Q&A system, Cooper, R. J., & Ruger, S. M. (2000, November). *A simple question answering system*, we could know that the theories and structures discussed in the other articles are actually implemented in a real Q&A system.

The data we found about the actual Q&A system and the theory of the Q&A response system were very helpful in making the actual product. Our program is very similar to the 'pattern matching' method. Our database has a question & answer pair that is prepared and our program uses answers that correspond to the question pattern.

After completing the literature reviews on the Q&A system, we started to search for knowledge and principles that will be useful for the front-end website that we are going to make. The front-end knowledge and theories that we found in our research have been very helpful in creating the actual front-end website for our project.

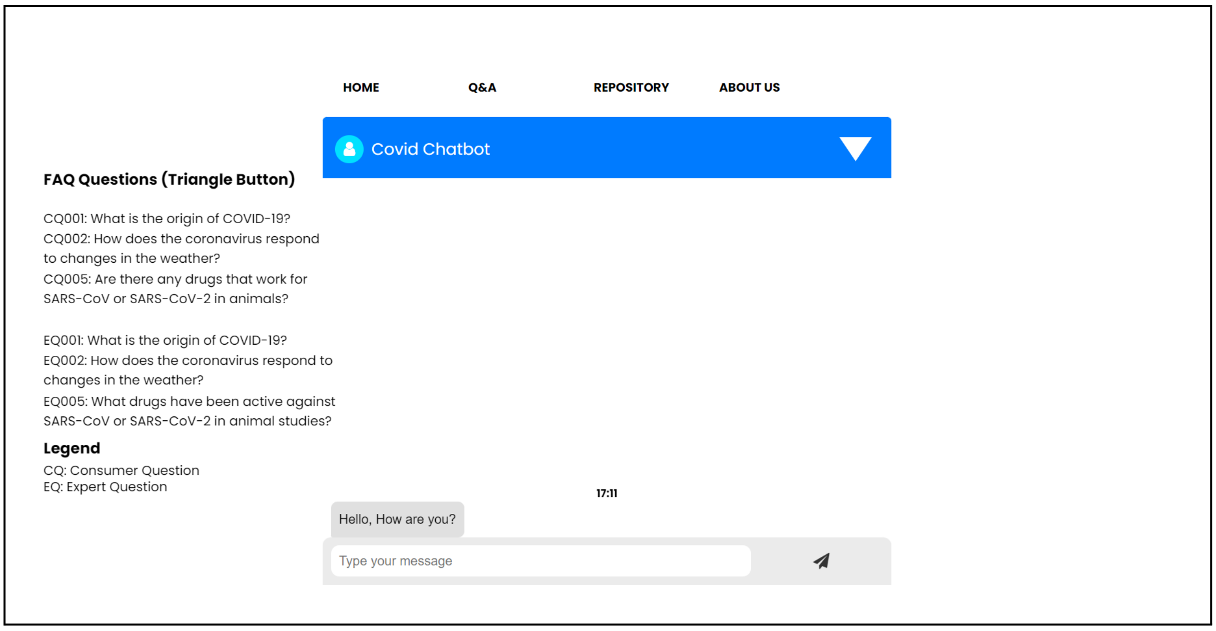
After completing the literature review on the database we chose the clear objective we wanted our database to achieve.We created an ERD diagram that we would use to create and implement the database as this gave us a good view of what was needed in our database. This clear objective setting and ERD diagram helped us immensely when creating the database as while we were implementing the project we had some problems that we had to deal with which affected our database, they helped us make the decisions we needed to make regarding the changes to solve the problems we faced..

4.0 Outcomes

4.1 Implementation

In this project, our team split the implementation into two parts: the front-end and the back-end.

For the front-end, two main features had been implemented - a covid chatbot, and a self-browsing repository. The covid chatbot's purpose is to evaluate and answer queries input by the user in the search bar. On the other hand, the self-browsing repository is to allow users to manually search and visualize their queries directly from the database.



FAQ Questions (Triangle Button)

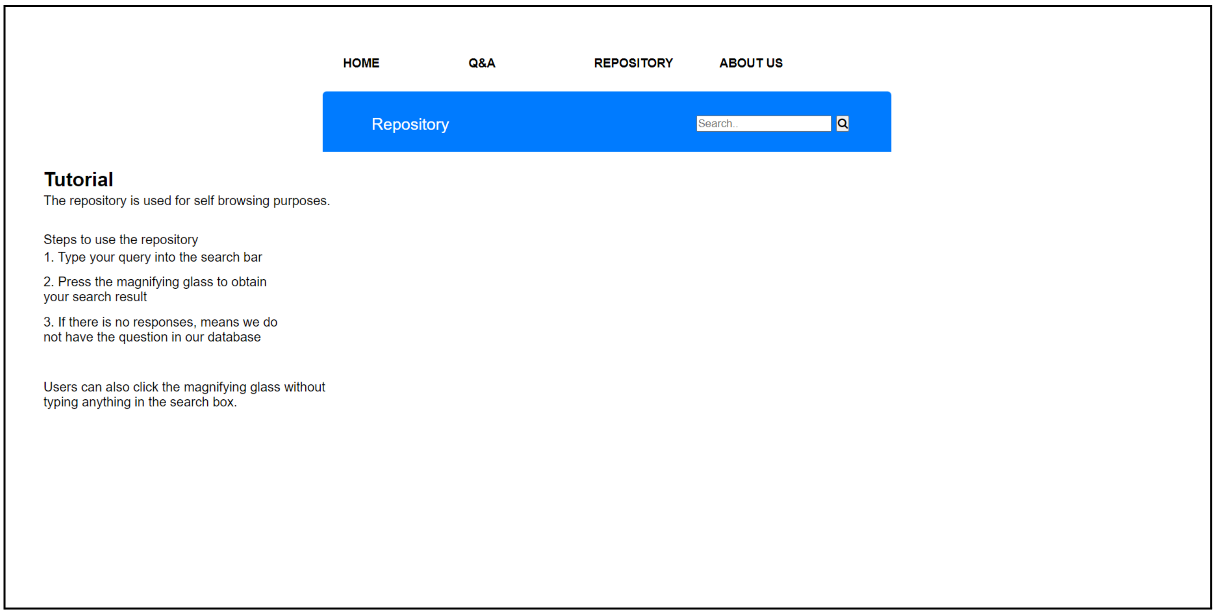
CQ001: What is the origin of COVID-19?
CQ002: How does the coronavirus respond to changes in the weather?
CQ005: Are there any drugs that work for SARS-CoV or SARS-CoV-2 in animals?

EQ001: What is the origin of COVID-19?
EQ002: How does the coronavirus respond to changes in the weather?
EQ005: What drugs have been active against SARS-CoV or SARS-CoV-2 in animal studies?

Legend
CQ: Consumer Question
EQ: Expert Question

Hello, How are you?

Figure 4.1.1: The covid chatbot



Tutorial

The repository is used for self browsing purposes.

Steps to use the repository
1. Type your query into the search bar

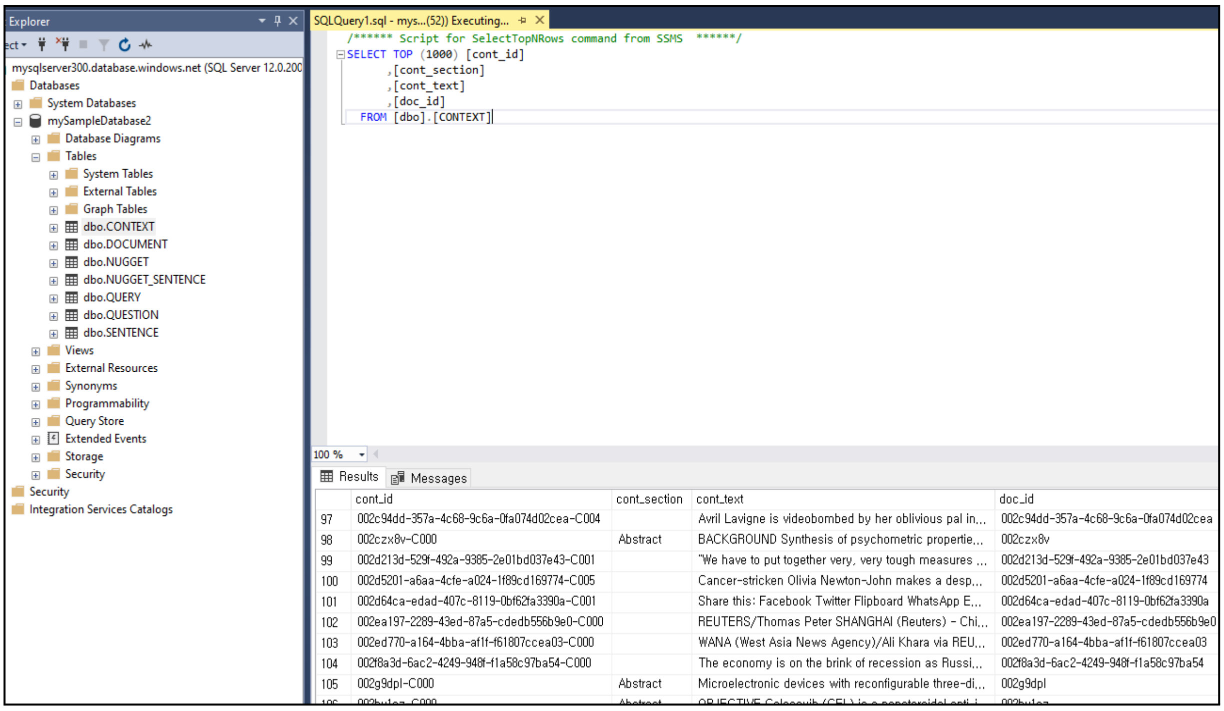
2. Press the magnifying glass to obtain your search result

3. If there is no responses, means we do not have the question in our database

Users can also click the magnifying glass without typing anything in the search box.

Figure 4.1.2: The self-browsing repository

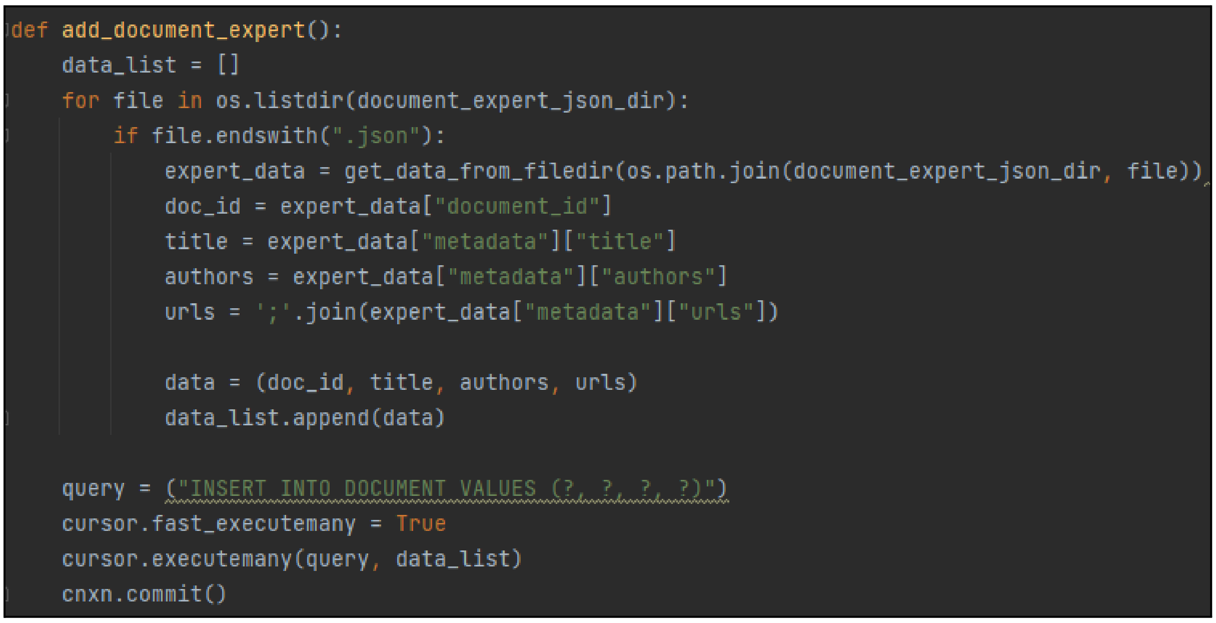
For the back-end, we created an Azure SQL database server, Python codes to populate data inside the DB server, and Web scraping codes to get question & answer pairs and insert the data into the database automatically.



```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [cont_id]
      ,[cont_section]
      ,[cont_text]
      ,[doc_id]
  FROM [dbo].[CONTEXT]
```

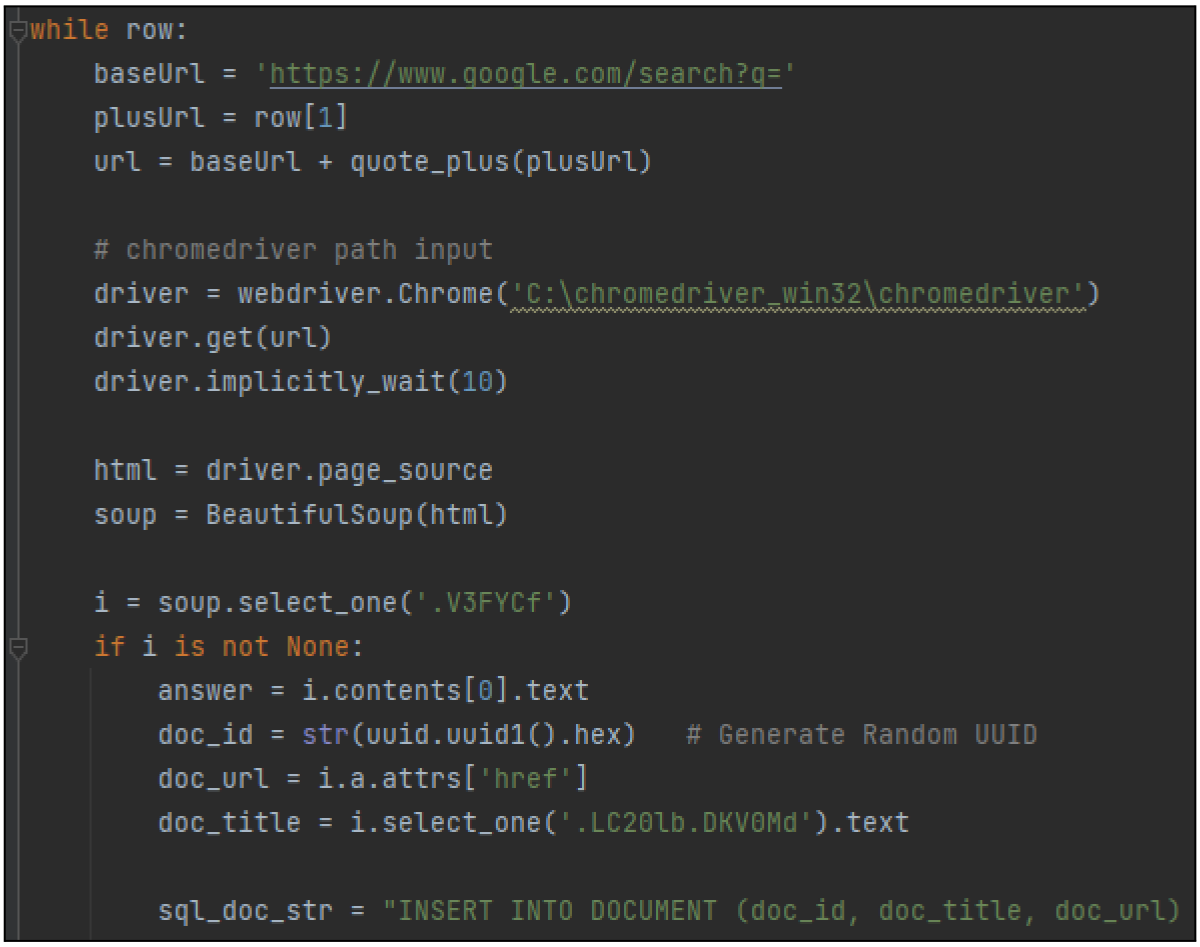
	cont_id	cont_section	cont_text	doc_id
97	002c94dd-357a-4c68-9c6a-0fa074d02cea-C004		Avril Lavigne is videobombed by her oblivious pal in,..	002c94dd-357a-4c68-9c6a-0fa074d02cea
98	002czx8v-C000	Abstract	BACKGROUND Synthesis of psychometric propertie,..	002czx8v
99	002d213d-529f-492a-9385-2e01bd037e43-C001		"We have to put together very, very tough measures ...	002d213d-529f-492a-9385-2e01bd037e43
100	002d5201-a6aa-4cfe-a024-1f89cd169774-C005		Cancer-stricken Olivia Newton-John makes a desp,..	002d5201-a6aa-4cfe-a024-1f89cd169774
101	002d64ca-edad-407c-8119-0bf62fa3390a-C001		Share this: Facebook Twitter Flipboard WhatsApp E,..	002d64ca-edad-407c-8119-0bf62fa3390a
102	002ea197-2289-43ed-87a5-cdedb556b9e0-C000		REUTERS/Thomas Peter SHANGHAI (Reuters) - Chi,..	002ea197-2289-43ed-87a5-cdedb556b9e0
103	002ed770-a164-4bba-af1f-f61807ccea03-C000		WANA (West Asia News Agency)/Ali Khara via REU,..	002ed770-a164-4bba-af1f-f61807ccea03
104	002f8a3d-6ac2-4249-948f-f1a58c97ba54-C000		The economy is on the brink of recession as Russi,..	002f8a3d-6ac2-4249-948f-f1a58c97ba54
105	002g9dpl-C000	Abstract	Microelectronic devices with reconfigurable three-di,..	002g9dpl

Figure 4.1.3: Database server



```python
def add_document_expert():
    data_list = []
    for file in os.listdir(document_expert_json_dir):
        if file.endswith(".json"):
            expert_data = get_data_from_filedir(os.path.join(document_expert_json_dir, file))
            doc_id = expert_data["document_id"]
            title = expert_data["metadata"]["title"]
            authors = expert_data["metadata"]["authors"]
            urls = ';'.join(expert_data["metadata"]["urls"])

            data = (doc_id, title, authors, urls)
            data_list.append(data)

    query = ("INSERT INTO DOCUMENT VALUES (?, ?, ?, ?)")
    cursor.fast_executemany = True
    cursor.executemany(query, data_list)
    cnxn.commit()
```

Figure 4.1.4: Part of data population Python script



```python
while row:
    baseUrl = 'https://www.google.com/search?q='
    plusUrl = row[1]
    url = baseUrl + quote_plus(plusUrl)

    # chromedriver path input
    driver = webdriver.Chrome('C:\chromedriver_win32\chromedriver')
    driver.get(url)
    driver.implicitly_wait(10)

    html = driver.page_source
    soup = BeautifulSoup(html)

    i = soup.select_one('.V3FYCf')
    if i is not None:
        answer = i.contents[0].text
        doc_id = str(uuid.uuid1().hex)   # Generate Random UUID
        doc_url = i.a.attrs['href']
        doc_title = i.select_one('.LC20lb.DKV0Md').text

        sql_doc_str = "INSERT INTO DOCUMENT (doc_id, doc_title, doc_url)
```

Figure 4.1.5: Part of web scraping Python script

4.2 Product Delivered

The products delivered from this project are as follows:

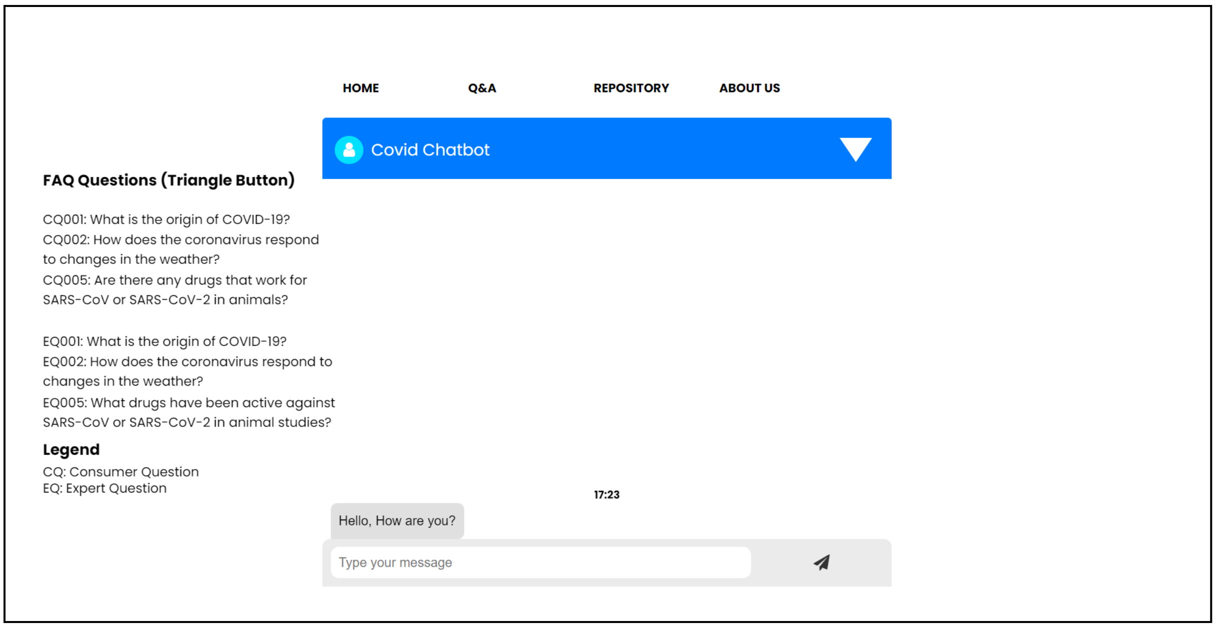
1. Home Page
2. Covid Chatbot
3. Self-Browsing Repository
4. About Us Page

4.2.1 Home Page

Figure 3 below shows the Home Page of our software product. As seen in the figure below, when the user first runs the program on the live server, this is the page that the user will land on. Firstly, at the top of the page, there is a navigation bar that brings the user to each page that we have implemented upon clicking. This enables discoverability in our software, to allow the user to immediately and directly navigate to a certain page. Besides that, the home page provides a short introduction to coronavirus, and then briefly states our project's main objective below it. There is also a quote of the day at the bottom of the page in order to prevent the page from being too dull and boring.

4.2.2 Covid Chatbot

Figure 4.2.1 below shows the covid chatbot of our software product. The covid chatbot is the main feature and functionality of our software. Firstly, as seen on the top of the page, similar to all pages, the navigation menu is there to allow users to conveniently navigate from page to page. Besides that, on the left, there is a legend which displays the type of questions in the Frequently Asked Questions (FAQ) list.



FAQ Questions (Triangle Button)

CQ001: What is the origin of COVID-19?
CQ002: How does the coronavirus respond to changes in the weather?
CQ005: Are there any drugs that work for SARS-CoV or SARS-CoV-2 in animals?

EQ001: What is the origin of COVID-19?
EQ002: How does the coronavirus respond to changes in the weather?
EQ005: What drugs have been active against SARS-CoV or SARS-CoV-2 in animal studies?

Legend

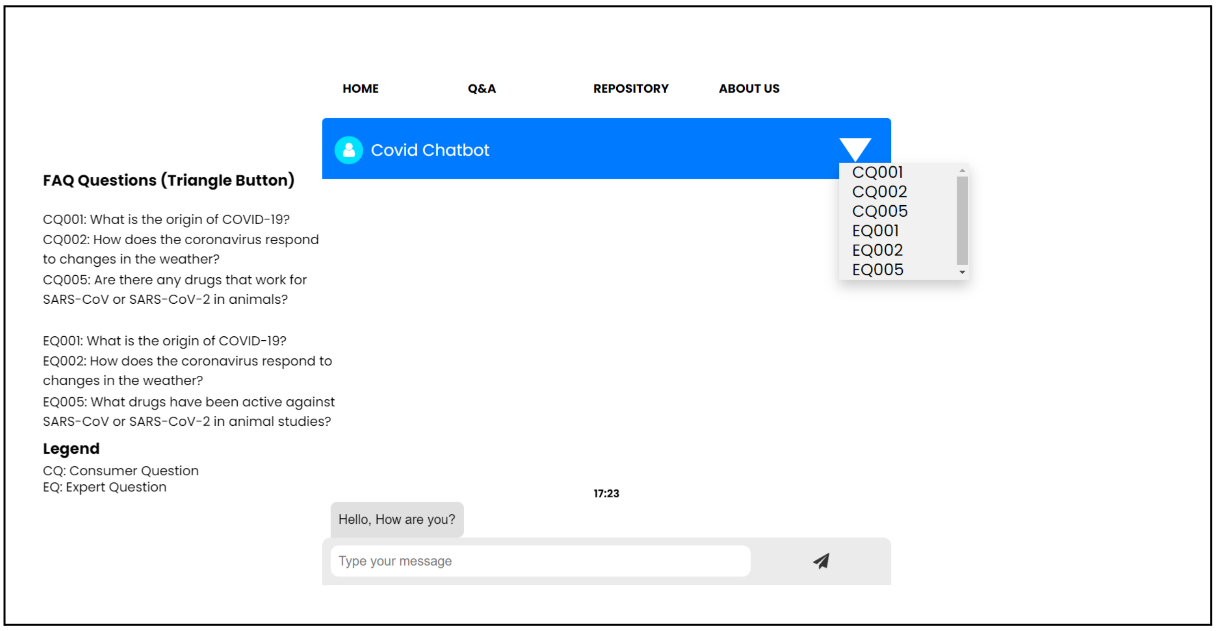
CQ: Consumer Question
EQ: Expert Question

17:23

Hello, How are you?

Figure 4.2.1: The covid chatbot

On the other hand, as seen across the page, there is a triangle icon at the top right corner. The triangle icon is directed downwards for affordance, to hint to the user regarding its functionality when clicked on, which is to display a drop-down list of FAQ (Figure 4.2.2). Upon selecting a question, a pop-up message will appear on the top of the screen, displaying the answer to the question selected (Figure 4.2.3).



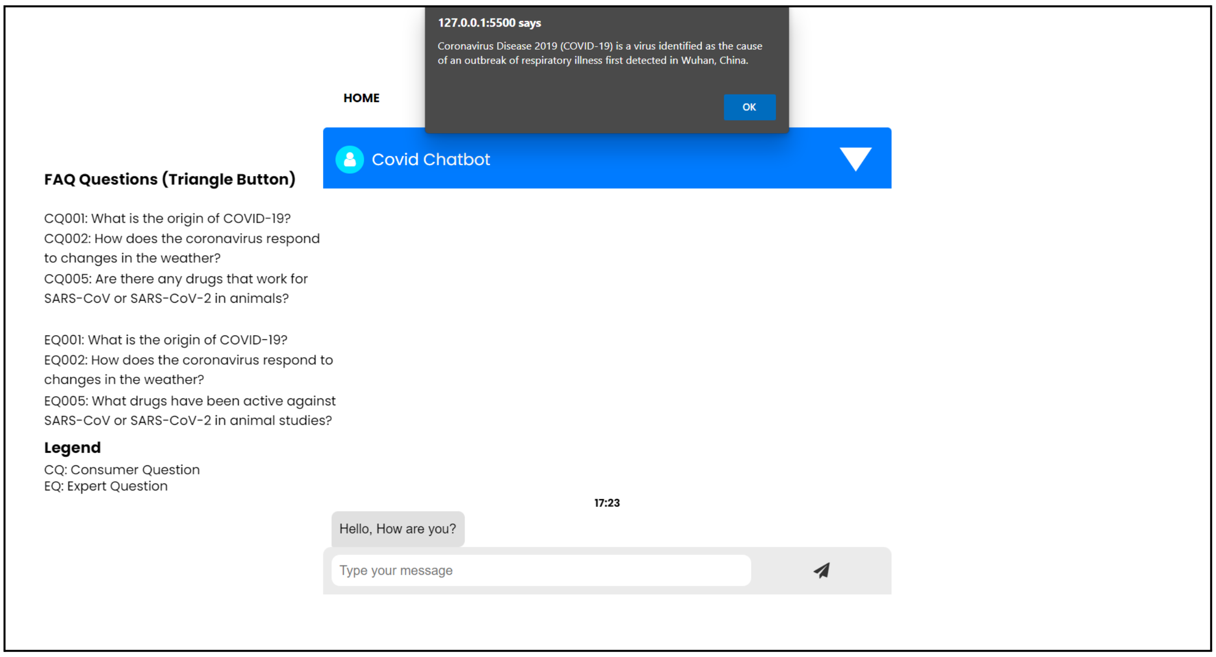
FAQ Questions (Triangle Button)

CQ001: What is the origin of COVID-19?
CQ002: How does the coronavirus respond to changes in the weather?
CQ005: Are there any drugs that work for SARS-CoV or SARS-CoV-2 in animals?

EQ001: What is the origin of COVID-19?
EQ002: How does the coronavirus respond to changes in the weather?
EQ005: What drugs have been active against SARS-CoV or SARS-CoV-2 in animal studies?

Legend
CQ: Consumer Question
EQ: Expert Question

Figure 4.2.2: The drop-down list of FAQ



127.0.0.1:5500 says

Coronavirus Disease 2019 (COVID-19) is a virus identified as the cause of an outbreak of respiratory illness first detected in Wuhan, China.

FAQ Questions (Triangle Button)

CQ001: What is the origin of COVID-19?
CQ002: How does the coronavirus respond to changes in the weather?
CQ005: Are there any drugs that work for SARS-CoV or SARS-CoV-2 in animals?

EQ001: What is the origin of COVID-19?
EQ002: How does the coronavirus respond to changes in the weather?
EQ005: What drugs have been active against SARS-CoV or SARS-CoV-2 in animal studies?

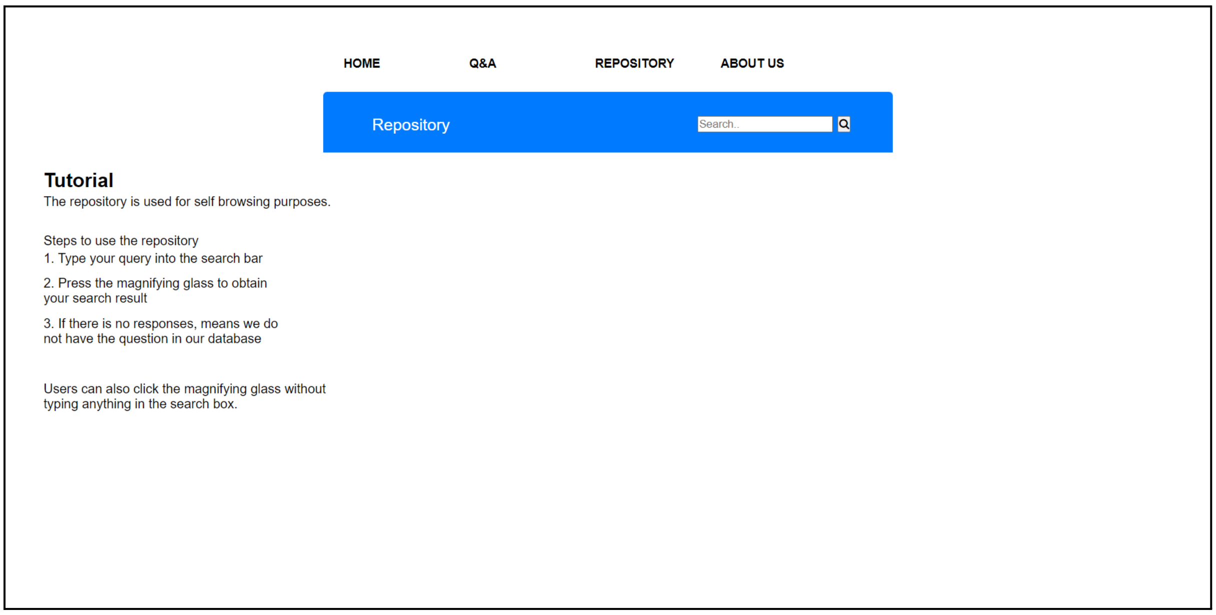
Legend
CQ: Consumer Question
EQ: Expert Question

Figure 4.2.3: The pop-up window displaying the answer to the FAQ selected

Next, the main feature of our software is the covid chatbot. As seen in Figure 4.2.1, the covid chatbot takes up the entire middle area of the screen, as it is the main focus of the program. The search bar and send message icon work together, allowing users to input alphabets and numbers to query to the system anything related to coronavirus. In return, the system fetches the relevant information related to the keywords in the question and subsequently answers the user's question.

4.2.3 Self-Browsing Repository

The second main feature of our software is the self-browsing repository (Figure 4.2.4). As shown in the figure below, there is a navigation menu at the top of the page for easy navigation. Then, on the left, there is a tutorial section dedicated to guide the user through using the repository. It gives a brief introduction to the repository, as well as short, concise, and easy to understand instructions to use it. Users can browse through the repository using their desired keyword using the search bar and magnifying glass icon on the top right.



Tutorial

The repository is used for self browsing purposes.

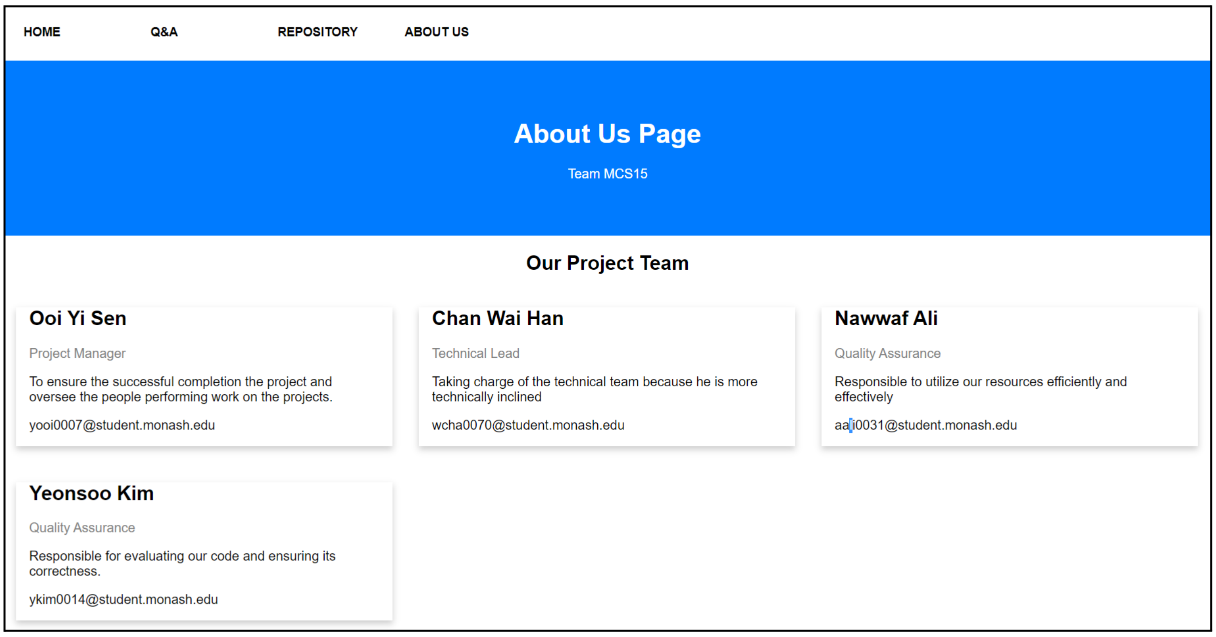
Steps to use the repository
1. Type your query into the search bar
2. Press the magnifying glass to obtain your search result
3. If there is no responses, means we do not have the question in our database

Users can also click the magnifying glass without typing anything in the search box.

Figure 4.2.4: The self-browsing repository page

4.2.4 About Us Page

The final page that we implemented was the About Us Page. This page is just to formally introduce our team to its users. It consists of the names, roles, short description of the role, and email of each project team member (Figure 4.2.5).



About Us Page

Team MCS15

Our Project Team

Ooi Yi Sen
Project Manager
To ensure the successful completion the project and oversee the people performing work on the projects.
yooi0007@student.monash.edu

Chan Wai Han
Technical Lead
Taking charge of the technical team because he is more technically inclined
wcha0070@student.monash.edu

Nawwaf Ali
Quality Assurance
Responsible to utilize our resources efficiently and effectively
aali0031@student.monash.edu

Yeonsoo Kim
Quality Assurance
Responsible for evaluating our code and ensuring its correctness.
ykim0014@student.monash.edu

Figure 4.2.5: The about us page

4.3 Requirements Met

During the planning phase of the project, our team had defined a set of requirements that needed to be satisfied by the end of the final product. The main functional requirements that were set are as follows:

- Access to the latest information about Covid. Users must be able to search for information about Covid, read the user guide on how to use our system, and see how to apply some of the information in real life
- Access to a repository where our user can perform self-browsing
- Access to reliable databases. So that our system can provide accurate information to our users
- Access to technical tools, languages, and software for our project.

Requirements 1 and 2 were implemented in the Covid Chatbot and Repository respectively.

Besides that, the business and social objectives that were set are as follows:

- Business objective: Provide useful information relating to covid
- Social objective: To improve everyone’s knowledge, acceptance and perception of covid

4.4 Decisions Made

Throughout the course of the project, many obstacles were met and many different decisions had to be made to overcome them.

4.4.1 Front-End

There was a technical issue with the coding language that we used. Initially, our team had decided on using HTML, CSS, and Python to program the front-end user interface. However, halfway into coding the functions of the chatbot, the front-end team were met with problems when sending messages to the chatbot. Essentially, when the user sends a new message to the chatbot, the previous messages that were sent could not be retained and would disappear. Therefore, this would increase the amount of effort required on the side of the user, where they would have to remember the keyword that they had previously queried. Our system could not adopt this implementation as our chatbot would have multiple questions related to certain keywords, and would subsequently ask the user to choose a question to be answered. Thus, the user interface has to retain the previously sent messages. In order to fix this problem, the front-end team had to switch to JavaScript. Essentially, JavaScript did not have to use HTTP Request to get this task done, therefore allowing the previously sent messages to remain on the screen when new messages are sent by the user.

4.4.2 Back-End

Using Azure SQL was not our team’s original plan in FIT3161/3163.
Before we started to build an actual product, after a few meetings, we tried to use 'Microsoft SQL server', which provides various text processing features because text processing is essential and the most important part in our project.

But we actually ran into a problem that we couldn't solve while creating the SQL server database. SQL server is a local hosted database server, so we had to set up all the databases and network settings for connection, which was the biggest problem.

We created a SQL server database using Yeonsoo Kim’s local PC, however, all of the team members could not connect to the database. To solve this problem, we had several meetings with team members and changed various network settings, but we could not solve the problem.

Finally, the way we chose to do it was to re-create the database with other member’s computers. However, this method also had the same problem, and eventually failed to solve it.

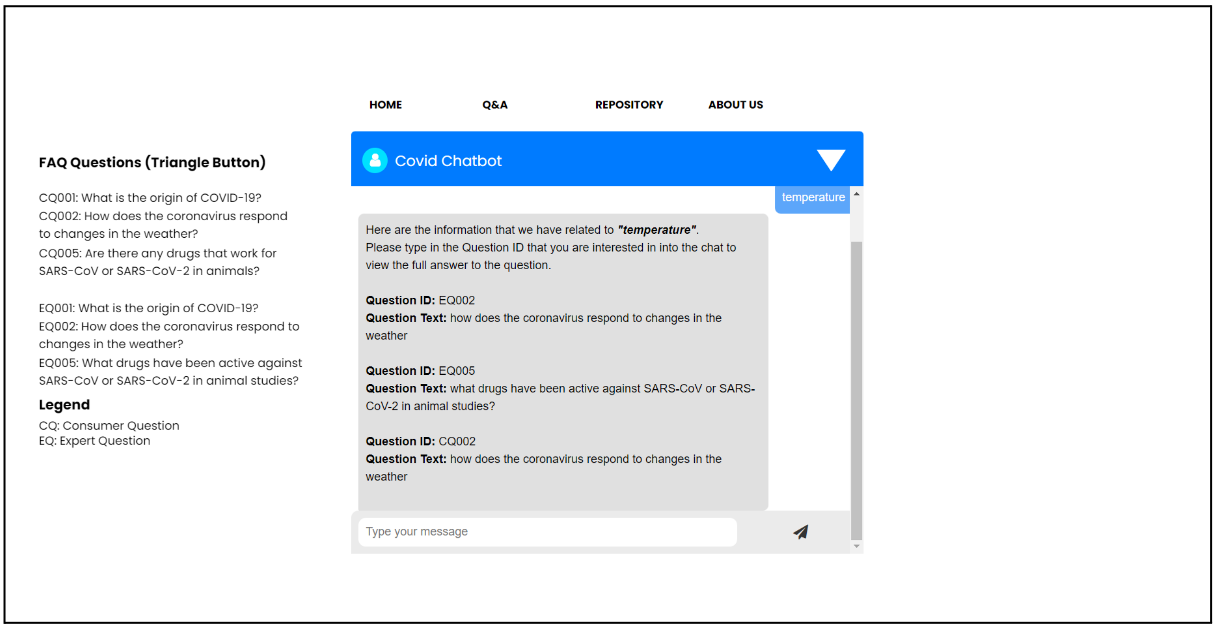
After a meeting with the supervisor about this issue, we finally decided to use 'Azure SQL', the same database software as SQL server, but a cloud-based database. By using cloud databases, we were able to completely solve problems with network settings, and we were able to use them.

4.5 Discussion of all results

This section discusses the output of the two main features of our software - the covid chatbot and the self-browsing repository.

4.5.1 Covid Chatbot

The covid chatbot is very straightforward, and acts like any other chat that a person would use on a daily basis on their social media. The user would just have to input a keyword into the search bar, and then click on the send message icon to send the message to the chatbot as a query. In Figure 4.5.1 below, the keyword "temperature" is sent to the chatbot. The chatbot subsequently checks the keyword and then fetches the data provided by the database to display the list of questions related to the keyword "temperature". The user would then select the question id to be answered, which would then be displayed by the chatbot (Figure 4.5.2).



FAQ Questions (Triangle Button)

CQ001: What is the origin of COVID-19?
CQ002: How does the coronavirus respond to changes in the weather?
CQ005: Are there any drugs that work for SARS-CoV or SARS-CoV-2 in animals?

EQ001: What is the origin of COVID-19?
EQ002: How does the coronavirus respond to changes in the weather?
EQ005: What drugs have been active against SARS-CoV or SARS-CoV-2 in animal studies?

Legend
CQ: Consumer Question
EQ: Expert Question

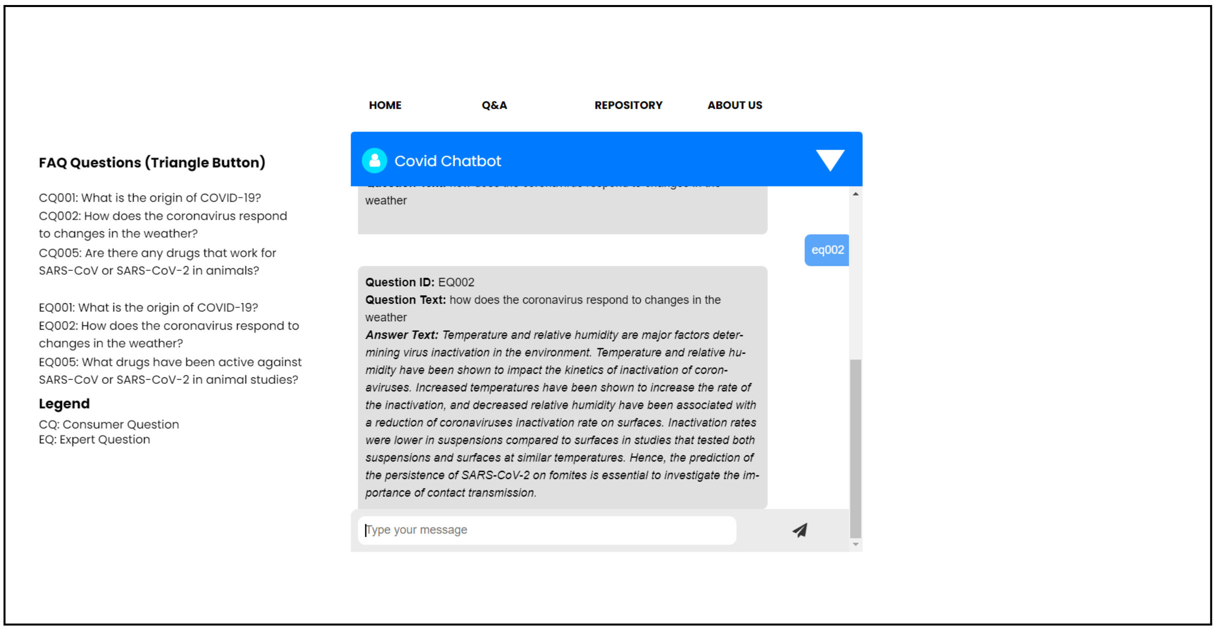
Here are the information that we have related to ***"temperature"***. Please type in the Question ID that you are interested in into the chat to view the full answer to the question.

Question ID: EQ002
Question Text: how does the coronavirus respond to changes in the weather

Question ID: EQ005
Question Text: what drugs have been active against SARS-CoV or SARS-CoV-2 in animal studies?

Question ID: CQ002
Question Text: how does the coronavirus respond to changes in the weather

Figure 4.5.1: Querying the keyword "temperature" to the covid chatbot


HOME Q&A REPOSITORY ABOUT US

FAQ Questions (Triangle Button)

CQ001: What is the origin of COVID-19?
CQ002: How does the coronavirus respond to changes in the weather?
CQ005: Are there any drugs that work for SARS-CoV or SARS-CoV-2 in animals?

EQ001: What is the origin of COVID-19?
EQ002: How does the coronavirus respond to changes in the weather?
EQ005: What drugs have been active against SARS-CoV or SARS-CoV-2 in animal studies?

Legend
CQ: Consumer Question
EQ: Expert Question

Covid Chatbot

weather

eq002

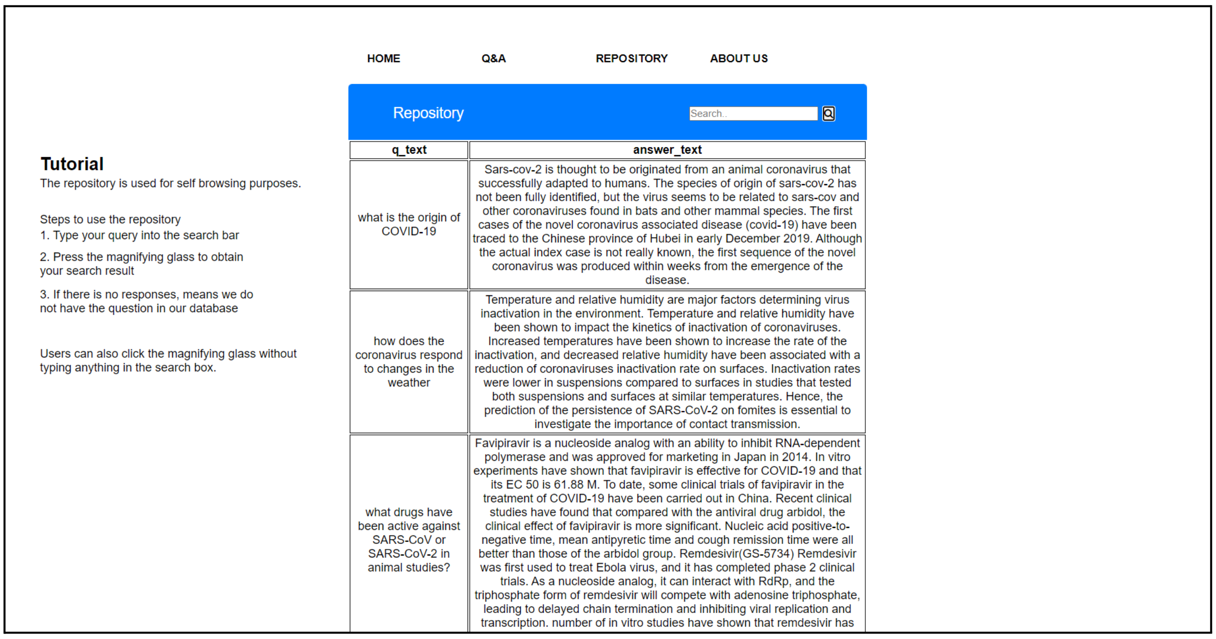
Question ID: EQ002
Question Text: how does the coronavirus respond to changes in the weather
Answer Text: *Temperature and relative humidity are major factors determining virus inactivation in the environment. Temperature and relative humidity have been shown to impact the kinetics of inactivation of coronaviruses. Increased temperatures have been shown to increase the rate of the inactivation, and decreased relative humidity have been associated with a reduction of coronaviruses inactivation rate on surfaces. Inactivation rates were lower in suspensions compared to surfaces in studies that tested both suspensions and surfaces at similar temperatures. Hence, the prediction of the persistence of SARS-CoV-2 on fomites is essential to investigate the importance of contact transmission.*

Type your message

Figure 4.5.2: Selecting the question id to be answered by the chatbot

4.5.2 Self-Browsing Repository

The repository was made for users to browse through our entire database themselves. Instead of going through the interface of a chatbot, the user can straight away view all of the questions and answers that we have available. The data will be displayed in the form of a table. FIgure 4.5.3 shows how a user can view our entire database, by clicking on the search icon without any keyword in the search box. On the other hand, Figure 4.5.4 shows how the repository would create a table to display the information related to the user query. In this case, it would be for the keyword "temperature".


HOME Q&A REPOSITORY ABOUT US

Tutorial
The repository is used for self browsing purposes.

Steps to use the repository
1. Type your query into the search bar
2. Press the magnifying glass to obtain your search result
3. If there is no responses, means we do not have the question in our database

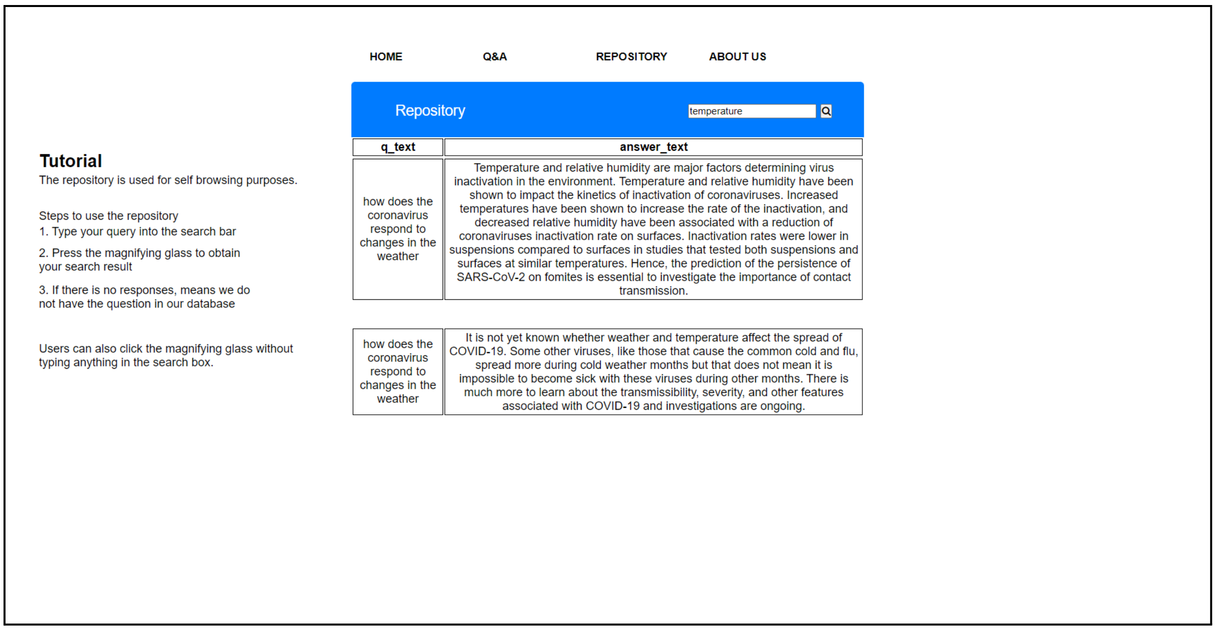
Users can also click the magnifying glass without typing anything in the search box.

Repository

Search..

q_text	answer_text
what is the origin of COVID-19	Sars-cov-2 is thought to be originated from an animal coronavirus that successfully adapted to humans. The species of origin of sars-cov-2 has not been fully identified, but the virus seems to be related to sars-cov and other coronaviruses found in bats and other mammal species. The first cases of the novel coronavirus associated disease (covid-19) have been traced to the Chinese province of Hubei in early December 2019. Although the actual index case is not really known, the first sequence of the novel coronavirus was produced within weeks from the emergence of the disease.
how does the coronavirus respond to changes in the weather	Temperature and relative humidity are major factors determining virus inactivation in the environment. Temperature and relative humidity have been shown to impact the kinetics of inactivation of coronaviruses. Increased temperatures have been shown to increase the rate of the inactivation, and decreased relative humidity have been associated with a reduction of coronaviruses inactivation rate on surfaces. Inactivation rates were lower in suspensions compared to surfaces in studies that tested both suspensions and surfaces at similar temperatures. Hence, the prediction of the persistence of SARS-CoV-2 on fomites is essential to investigate the importance of contact transmission.
what drugs have been active against SARS-CoV or SARS-CoV-2 in animal studies?	Favipiravir is a nucleoside analog with an ability to inhibit RNA-dependent polymerase and was approved for marketing in Japan in 2014. In vitro experiments have shown that favipiravir is effective for COVID-19 and that its EC 50 is 61.88 M. To date, some clinical trials of favipiravir in the treatment of COVID-19 have been carried out in China. Recent clinical studies have found that compared with the antiviral drug arbidol, the clinical effect of favipiravir is more significant. Nucleic acid positive-to-negative time, mean antipyretic time and cough remission time were all better than those of the arbidol group. Remdesivir(GS-5734) Remdesivir was first used to treat Ebola virus, and it has completed phase 2 clinical trials. As a nucleoside analog, it can interact with RdRp, and the triphosphate form of remdesivir will compete with adenosine triphosphate, leading to delayed chain termination and inhibiting viral replication and transcription. number of in vitro studies have shown that remdesivir has

Figure 4.5.3: Repository displaying the entire database



Tutorial

The repository is used for self browsing purposes.

Steps to use the repository
1. Type your query into the search bar
2. Press the magnifying glass to obtain your search result
3. If there is no responses, means we do not have the question in our database

Users can also click the magnifying glass without typing anything in the search box.

q_text	answer_text
how does the coronavirus respond to changes in the weather	Temperature and relative humidity are major factors determining virus inactivation in the environment. Temperature and relative humidity have been shown to impact the kinetics of inactivation of coronaviruses. Increased temperatures have been shown to increase the rate of the inactivation, and decreased relative humidity have been associated with a reduction of coronaviruses inactivation rate on surfaces. Inactivation rates were lower in suspensions compared to surfaces in studies that tested both suspensions and surfaces at similar temperatures. Hence, the prediction of the persistence of SARS-CoV-2 on fomites is essential to investigate the importance of contact transmission.
how does the coronavirus respond to changes in the weather	It is not yet known whether weather and temperature affect the spread of COVID-19. Some other viruses, like those that cause the common cold and flu, spread more during cold weather months but that does not mean it is impossible to become sick with these viruses during other months. There is much more to learn about the transmissibility, severity, and other features associated with COVID-19 and investigations are ongoing.

Figure 4.5.4: Repository displaying the information related to the keyword "temperature

4.6 Limitations

There are a few limitations in terms of the functionalities and features of our software. Firstly, the main limitation is that the keywords that the user queries should ideally be keywords only. This is because for our chatbot if the user queries with a single alphabet, such as the letter “o”, the chatbot would return all questions and answers associated with the character “o”. With “o” being a vowel, it is very likely that most questions and answers would be returned. Ultimately, doing such a thing will not benefit the user.

Besides that, our software cannot answer detailed or non-popular questions. This is because our software does not automatically generate answers when it receives a user query. Instead, it fetches information from a pre-processed database related to coronavirus.

4.7 Possible Improvements and Future Works

Based on the limitations mentioned above, there are a few possible improvements that could be done to hopefully overcome those problems. Firstly, an improvement to the software would be to look for or create an algorithm that will be able to search for keywords within a sentence. This would allow the user to query an entire sentence, rather than just limiting them to querying a keyword only. Besides that, we could improve the software by enabling the ability for it to generate answers for detailed and non-popular questions through direct fetching of answers from Google itself. Therefore, rather than fetching data from a pre-processed database, we can implement it in a way where it straight away gets the best possible answer from Google.

In terms of future work, the software can be continuously fed with new information. This would allow the chatbot to be constantly up-to-date, and will be able to always provide the latest information regarding coronavirus to the users. Besides that, an escalation path can be added to the chatbot, where an eventual interaction with a human advisor will take place should a user be unsatisfied with the answer provided to their query.

5.0 Methodology

In this section, the final design of the Chatbot and Repositoru will be displayed. Besides that, the frontend software will be discussed about. Additionally, the deviation from th initial design, as well as the software and tools used will be discussed.

5.1 Final Design of the Question and Answer Chatbot

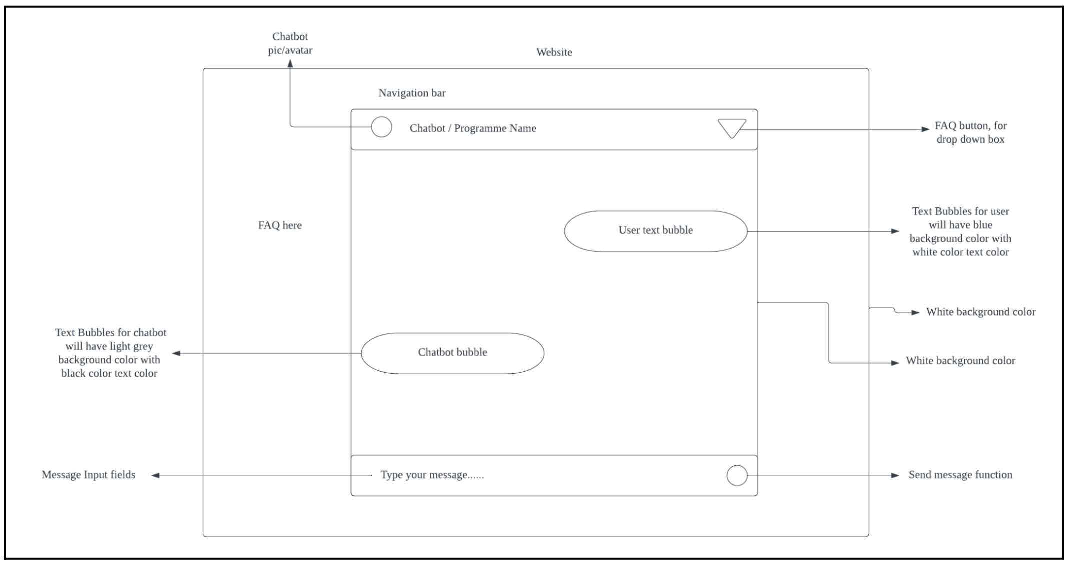


Figure 5.1: Final Design of the Question and Answer Chatbot

5.2 Final Design of the Repository

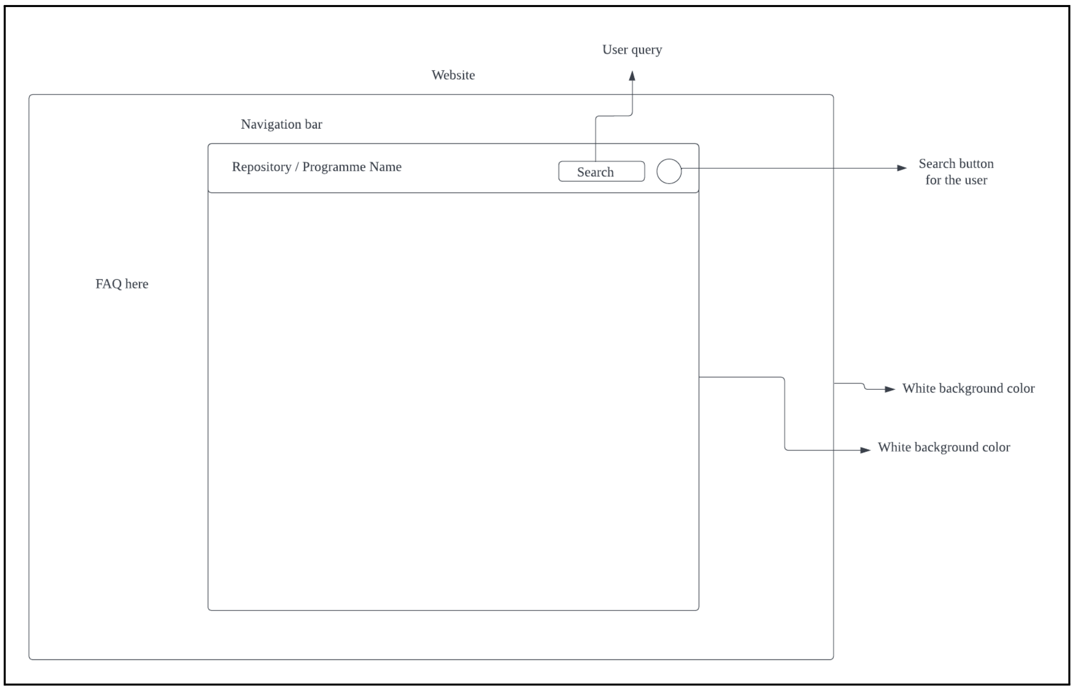


Figure 5.2: Final Design of the Repository

5.3 Front-End Software

Visual Studio Code was utilized by our team to construct the website's front end. We choose Visual Studio Code as our development environment because it combines the ease of use of a source code editor with strong developer tools such as IntelliSense code completion and debugging. Another reason we choose Visual Studio Code is that it has extensions that enable us to operate the local server directly from our computers and also to connect with the azure SQL database and retrieve the dataset.

5.4 Deviation from the Initial Design for Front-End

For our initial design, we just had two web pages for our software:

The Chatbot:

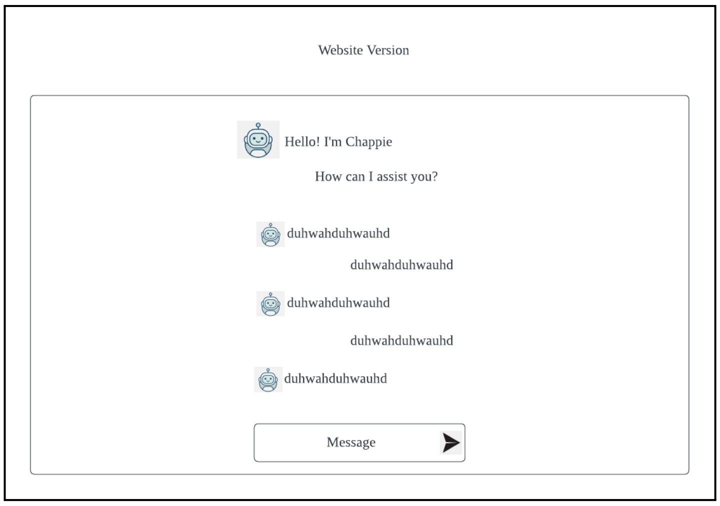


Figure 5.3: Initial Design for Chatbot

The Repository:

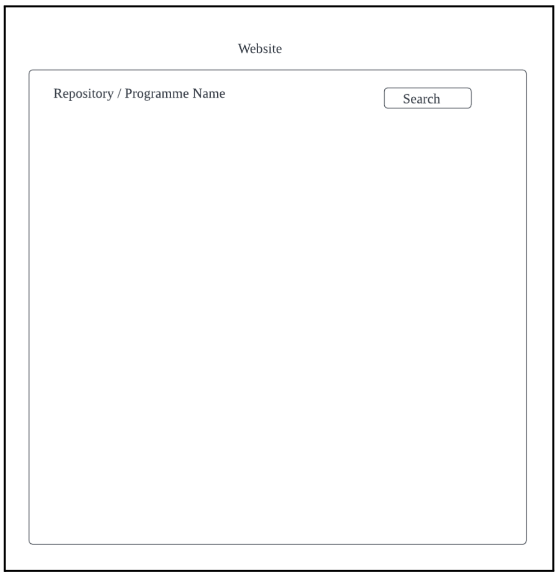


Figure 5.4: Initial Design for Repository

We added two more pages to our product after receiving comments from our project supervisor. The extra pages provide an introduction to our project as well as a description of our project team. Our project team agreed with the project supervisor's recommendation since the extra two pages will make our software seem more professional on the website.

Aside from that, our team deviated from the initial design of the chatbot and repository website. We did not include a tutorial on how to use our pages in our first proposal. However, our project manager advised us to add it because it may be inconvenient for some users. We also added a navigation bar on top of the title of the page, so that users can easily navigate through each of our web pages.

We also added extra functionalities to our chatbot and repository pages. We didn't include the dropdown box feature to enable users to choose our commonly asked questions in the initial design for the chatbot website.

Another deviation was the programming language we utilized. We intended to utilize Python as our primary coding language to create the logic for the chatbot and the repository. However, we ran into some issues where, while we could collect the query from our users and send the request using Python, we couldn't create a response chatbot. After studying several coding languages, we discovered that utilizing JavaScript addresses the challenge of creating a responsive chatbot as well as receiving and submitting user requests.

5.5 Software and Tools used for the Front-End

1. HTML/CSS is used for website development
2. Javascript will be used to create the responsive functionality
3. GitHub project used to store our source code
4. Visual Studio Code for our front-end development

5.6 Final Design of the Database

Final Design of the Database

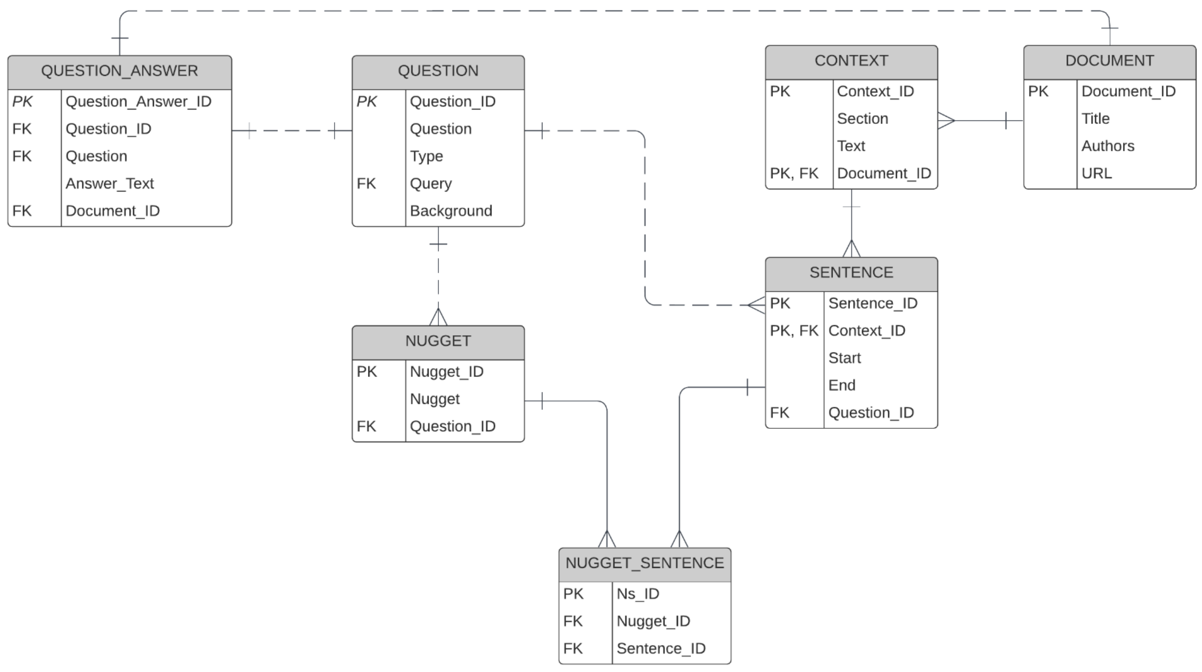


Figure 5.5: Final Database Entity Relationship diagram

This is the final ER diagram for our database.
‘Document’ table stores a document that contains all the texts of a document.
‘Context’ table stores contexts (paragraphs) of documents.
‘Sentence’ table stores the sentences of the contexts.

‘Question’ table stores the possible users’ questions.
‘Nugget’ table stores the keywords of the questions.
‘Question_Answer’ stores the question & answer pair that is used in the front-end software.

Database Software

The Database software that our team uses is ‘Microsoft Azure SQL’, which is a cloud-based database server, provided by Microsoft.
The reason why we decided to use Microsoft Azure SQL is that Microsoft Azure SQL provides a variety of text processing functions and text processing is the most important and core part of our project. In addition, the other reasons we chose this software are that we do not have to turn on one’s computer all day to maintain connections and we do not have to configure network settings to enable others to connect to the database because Microsoft Azure SQL is a cloud-based DB server.

5.7 Deviation from the Initial Design for Back-End

Deviation From the Initial Design

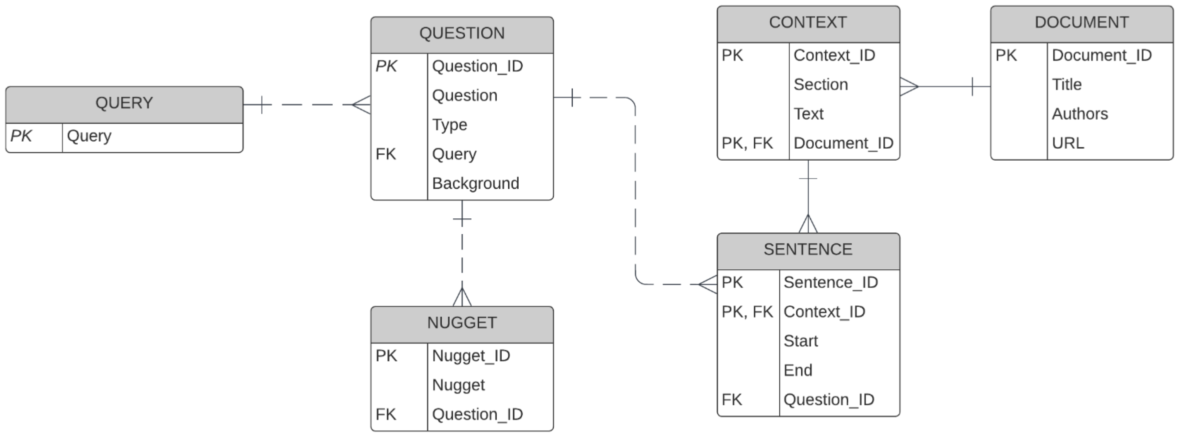


Figure 5.6 Initial ER diagram planning

The structure of the database has been changed as well. Figure x.x is the initial design of our database. We thought this structure was good enough, and we put the datasets in the database.
However, as the project was developed, there were many problems with this structure.
First, there was an unnecessary number of queries that had to be performed to get the answer to the question, and the query itself was very complicated.
There was no problem with the structure itself, but we added a separate table to save our answers because we often had problems when creating or understanding queries during development.
Second, the Query table became completely unnecessary as the development progressed. We initially intended to use the Query table, but as we progressed the development, we decided that the Query element itself in the Question table was sufficient.
By applying these improvements and modifying them, the final database structure was finally completed like Figure x.x.

Software and Tools Used for the Back-end

1. Microsoft Azure SQL, for the database server
2. Python is used to populate the data inside the DB server

6.0 Software Deliverables

In this section, we will be describing the software deliverables for our project.

6.1 Summary of software deliverables

Our project consists of two main software deliverables:

1. The source code for our Visual Studio Code, where we write and perform testing for our code.
2. A webpage to demonstrate the functionality of our system.

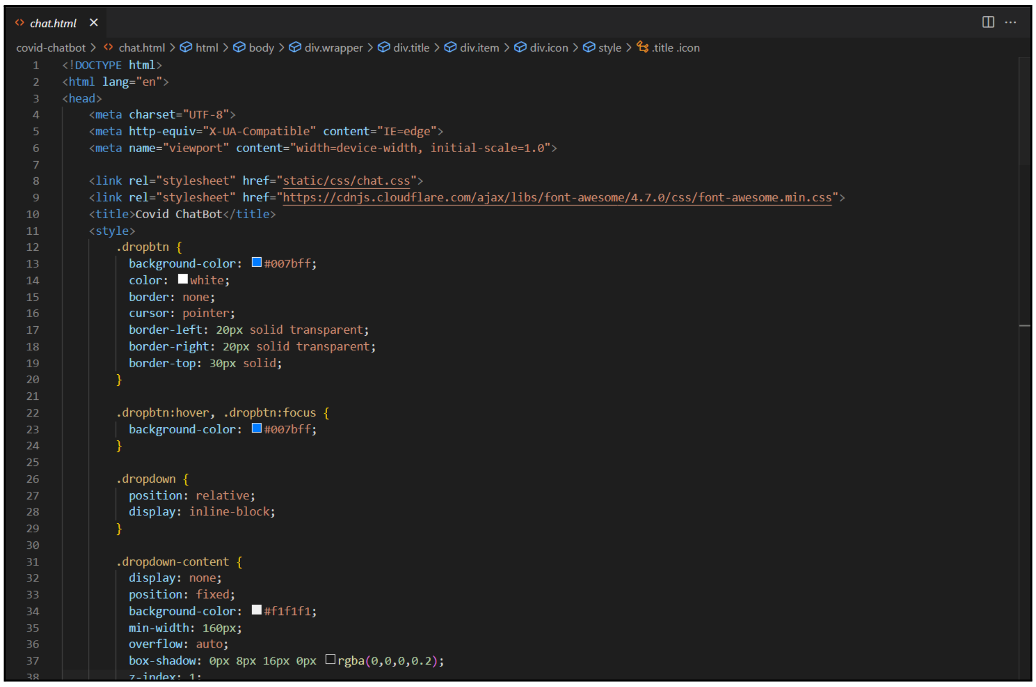
The libraries that are used to implement our software:

Front-End
1. Google Fonts API

Back-End
1. 'json' to read JSON files
2. 'os' to read system files
3. 'pyodbc' to connect to the database
4. 'CSV' to convert .json files to .csv file
5. ‘BeautifulSoup' format HTML codes into data
6. 'selenium' to run virtual chrome taps
7. 'UUID' to generate a random ID for new documents found

Sample screenshots of our software:

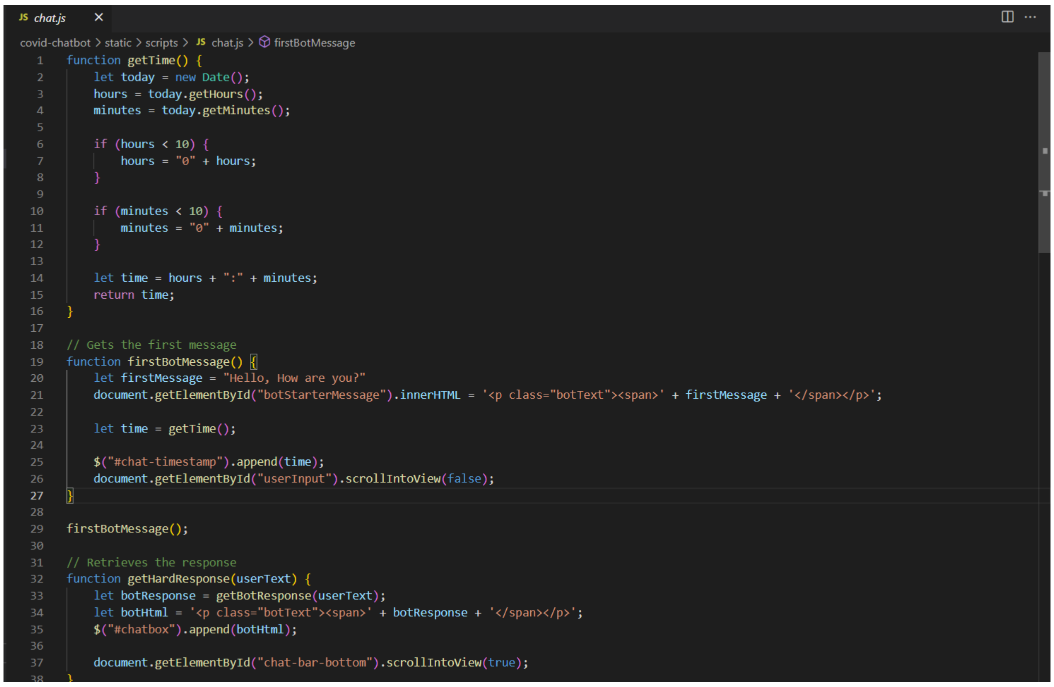
1. Chat.html


chat.html
covid-chatbot > chat.html > html > body > div.wrapper > div.title > div.item > div.icon > style > .title .icon

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="static/css/chat.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
    <title>Covid ChatBot</title>
    <style>
        .dropbtn {
          background-color: #007bff;
          color: white;
          border: none;
          cursor: pointer;
          border-left: 20px solid transparent;
          border-right: 20px solid transparent;
          border-top: 30px solid;
        }

        .dropbtn:hover, .dropbtn:focus {
          background-color: #007bff;
        }

        .dropdown {
          position: relative;
          display: inline-block;
        }

        .dropdown-content {
          display: none;
          position: fixed;
          background-color: #f1f1f1;
          min-width: 160px;
          overflow: auto;
          box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
          z-index: 1;
```

This HTML file contains the layout and design for our question-and-answer (QnA) chatbot.

2. Chat.js

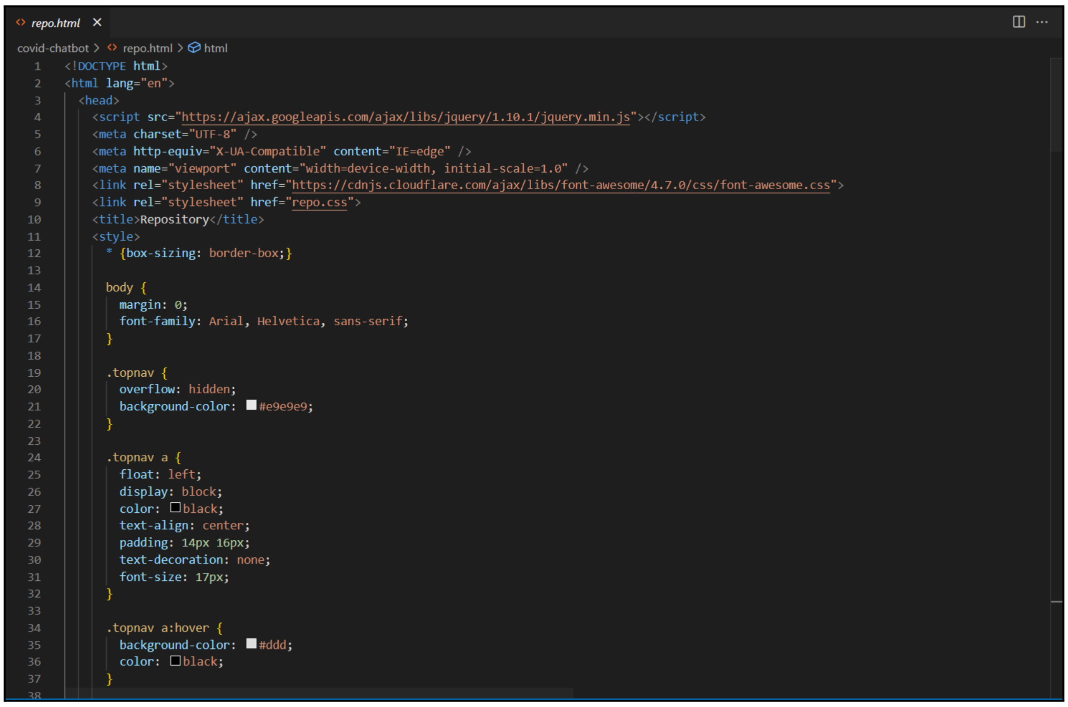

chat.js
covid-chatbot > static > scripts > chat.js > firstBotMessage

```javascript
function getTime() {
    let today = new Date();
    hours = today.getHours();
    minutes = today.getMinutes();

    if (hours < 10) {
        hours = "0" + hours;
    }

    if (minutes < 10) {
        minutes = "0" + minutes;
    }

    let time = hours + ":" + minutes;
    return time;
}

// Gets the first message
function firstBotMessage() {
    let firstMessage = "Hello, How are you?"
    document.getElementById("botStarterMessage").innerHTML = '<p class="botText"><span>' + firstMessage + '</span></p>';

    let time = getTime();

    $("#chat-timestamp").append(time);
    document.getElementById("userInput").scrollIntoView(false);
}

firstBotMessage();

// Retrieves the response
function getHardResponse(userText) {
    let botResponse = getBotResponse(userText);
    let botHtml = '<p class="botText"><span>' + botResponse + '</span></p>';
    $("#chatbox").append(botHtml);

    document.getElementById("chat-bar-bottom").scrollIntoView(true);
}
```

This JS file contains the major functions for our question-and-answer (QnA) chatbot. All functions here would provide us with query functionality

.

3. Repo.html



```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.1/jquery.min.js"></script>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.css">
    <link rel="stylesheet" href="repo.css">
    <title>Repository</title>
    <style>
      * {box-sizing: border-box;}

      body {
        margin: 0;
        font-family: Arial, Helvetica, sans-serif;
      }

      .topnav {
        overflow: hidden;
        background-color: #e9e9e9;
      }

      .topnav a {
        float: left;
        display: block;
        color: black;
        text-align: center;
        padding: 14px 16px;
        text-decoration: none;
        font-size: 17px;
      }

      .topnav a:hover {
        background-color: #ddd;
        color: black;
      }
```

This HTML file contains the layout and design for our repository.

All of these deliverables can be found in our GitHub repository.

6.2 Software Qualities

There are two major concerns with software s. The first concern is about the functional quality of the software, and the second is about the non-functional quality of the software. The functional quality of the software is how effectively it corresponds to given functional requirements. In other words, it is the extent to which appropriate software was produced. The non-functional quality of the software is whether it supports the supply of functionality needs such as maintenance, robustness, and so on. During the early phases of development, our project team devoted special attention to the software quality of our project. We will now discuss each individual attribute below.

6.2.1 Robustness

To ensure all of our inputs are valid, our software only accepts inputs that are in the format of words and digits. This prevents users from sending any other inputs with unreadable formats, which may potentially cause our program to fail.

If the user proceeds in inputting the wrong format, our software will then provide a notification telling the user that we do not have this question in our database and to input a new question.

6.2.2 Security

Website security refers to any action or software put in place to guarantee that website data is not exposed to cybercriminals or to prevent the website from being abused in any way. These activities aid in the protection of sensitive data, hardware, and software on a website against the numerous sorts of threats that exist today.

Since we are using a local host rather than online hosting, website security is not a primary concern for our project. We picked local hosting because it allows us to do private testing on our computer without transmitting files over the internet, which is more secure and will not expose your website to the public (Juviler, 2021).

HTTPs protocol is used by the website from where we obtained our datasets. The use of HTTPs avoids interceptions and disruptions while the material is in transit. We also verify that the website has an SSL Certificate to ensure the internet connection is safe. SSL is another essential site protocol that encrypts data to prevent others from accessing it while it is in transit. It prevents individuals without sufficient authorization from accessing the data (Hendrick, n.d.). Hence we can conclude that our dataset is safe and secure.

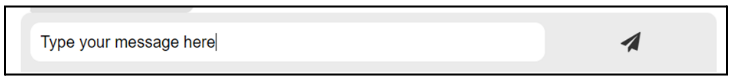
Security of the back-end Database strictly depends on the security of the Microsoft Azure SQL Database itself. It is because Microsoft Azure SQL is a cloud-based database that is actually managed by Microsoft, which means that the database is not opened and configured by individuals. The only security measures we can do are to set up firewall rules, classify user/admin roles and prevent the admin account password leakage.

But our team thinks this is an advantage, not a disadvantage. Security measures are often not within the reach of developers and are largely dependent on security professionals. Because our team is also just students, not security experts, we believe that using cloud-based DB, run by large corporations is bound to be more secure than DB, running personally.

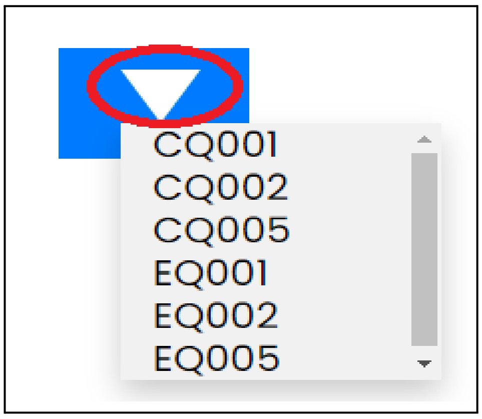
6.2.3 Usability

The three main UI components that the user will interact with are:

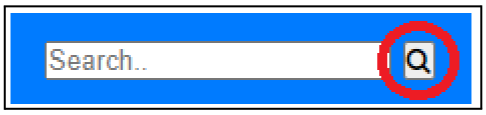
1. The input bar to query for questions in our question and answer page.



2. The dropdown bar for our question and answer page.



3. The input bar to query for our dataset on our repository page.



All of the above-mentioned features are intended to be simple to use. If the user still has questions about how to use our software, a short tutorial on how to utilize our functions will be available on the left side of the website.

6.2.4 Scalability

The scalability of our software was considered when we were in the early planning stages of our project. In order to ensure the scalability of our software, our team performed preprocessing in the backend. The preprocessing ensures that only the required columns are put into the Azure database. Therefore, when the database information is retrieved in JavaScript to be stored in the JSON file, it is extremely straightforward. Essentially, if more rows are added to the data, it will always be preprocessed in the backend before being retrieved. In other words, the JavaScript code will only retrieve the dataset, regardless of its size, and will not need to perform any processing.

6.2.5 Documentation and Maintainability

During the implementation phase, our project team thoroughly documented all of our source code. Aside from that, we created an end-user guide and a technical guide to assist anyone who may be using our software. Each function is thoroughly documented to ensure that other developers would understand our source code.

6.3 Sample Source Code

Refer to Appendix 10.1 for a snipper of our source code. This code snippet shows how our software handles the response from our user and handles the sending text with button clicks.

7.0 Critical Discussion

The listed requirements which we had to satisfy was to come up with a working chatbot and a self browsing repository. Therefore, our team believes that the project has been executed decently, as we have managed to satisfy those two main requirements. However, there is no doubt that the execution was not perfect. It could have been much more efficient and smooth, especially during the plannnig phase. Essentially, during the planning phase, our team did not go into much detail regarding many aspects, such as the libraries and functions that could be used for the code. Besides that, we had done inadequate research for the database platforms. Due to the insufficient research from the front-end and back-end team, we suffered inevitable setbacks during the implementation and development phase. Consequently, not only did we have to spend time migrating to a different code and platform, but we also had to spend time researching the correct code and platform that would get the job done for us. This time spent could have been mitigated, had we planned in more detail in the planning phase of this project. Subsequently, we could have spent that time on other unit assessments, or on improving this project in terms of performance or functionalities and features.

Regardless, our team would consider our project to be a decent success, as the following outcomes were successfully achieved:

- Successfully migrated the failed front-end Python code to JavaScript
- Successfully migrated from Microsoft SQL Server to Azure SQL
- Succesfully input the correct dataset
- Developed the front-end with an interactable user interface
- Developed the back-end to be able to send and receive information

Furthermore, as outlined in our Project Proposal in FIT3161, and in Section 4.3 - Requirments Met of this document, we had successfully satisfied the main functional requirements, business and social objectives of our project.

8.0 Conclusion

In this project, we have successfully created a covid chatbot that is able to provide answers based on user queries. Besides that, we have also successfully developed a repository for users to self-browse for information. The two main functionalities of our project were made possible through the implementation of a database in the back-end to store information, and then a front-end system to filter and retrieve the relevant information based on the user input data.

Additionally, we were able to create a Home page to provide a brief introduction to our software, as well as an About Us page to briefly introduce the development team to the users. Throughout implementation, our team ensured that our design adhered to functional and non-functional qualities, such as robustness, security, usability, scalability, as well as documentation and maintainability.

Our team concludes that the project has been a success as we have been able to satisfy the requirements as defined previously:

- Access to the latest information about Covid. Users must be able to search for information about Covid, read the user guide on how to use our system, and see how to apply some of the information in real life
- Access to a repository where our user can perform self-browsing
- Access to reliable databases. So that our system can provide accurate information to our users
- Access to technical tools, languages, and software for our project.
- Business objective: Provide useful information relating to covid
- Social objective: To improve everyone's knowledge, acceptance and perception of covid

Lastly, this project can be focused on various aspects if it were to be developed further in the future, mainly:

- Querying with more than just keywords
- Ability to generate answers for detailed and non-popular questions
- Ability to fetch answers from Google
- Periodic, incremental updates of Q&A database
- Escalation path for interaction with human advisor

9.0 References

Benítez-Andrades, José Alberto, Alija-Pérez, José-Manuel, Vidal, Maria-Esther, Pastor-Vargas, Rafael, & García-Ordás, María Teresa. (2022). *Traditional Machine Learning Models and Bidirectional Encoder Representations From Transformer (BERT)-Based Automatic Classification of Tweets About Eating Disorders: Algorithm Development and Validation Study. JMIR Medical Informatics, 10(2), e34492–e34492.* https://doi.org/10.2196/34492

Color Matters. 6 Tips on Choosing UI Colors. (2017). *Tubik Blog: Articles about Design.* https://blog.tubikstudio.com/color-matters-6-tips-on-choosing-ui-colors/

Cooper, R. J., & Ruger, S. M. (2000, November). *A simple question answering system. In TREC.* http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.21.1041&rep=rep1&type=pdf

Dwivedi, S. K., & Singh, V. (2013). *Research and reviews in question answering system. Procedia Technology, 10, 417-424.* https://reader.elsevier.com/reader/sd/pii/S2212017313005409?token=E5102E56714A39CA5D993001AD15B7E8F73DACD1481AD7EE5FE938CA9CC4B26083219D554B32280ABEDB6186066AC9C3&originRegion=eu-west-1&originCreation=20220525181349

Goonawardene, P. (2021). *How can Shneiderman's 8 Golden Rules Contribute to Usability? Medium.* https://pathumpmgux.medium.com/how-can-shneidermans-8-golden-rules-contribute-to-usability-c908f7f4de03

Hendricks, D. (n.d.). 10 Essential Steps To Improve Your Website Security. https://www.computer.org/publications/tech-news/trends/10-essential-steps-to-improve-your-website-security

Juviler, J. (2021, October 14). What Is Localhost? (And How Can You Use It?). https://blog.hubspot.com/website/what-is-localhost

Lalithnarayan, C. (2020, December 30). *An Introduction to Question Answering Systems. Engineering Education (EngEd) Program | Section.* https://www.section.io/engineering-education/question-answering/

Microsoft(n.d.) Database design basics https://support.microsoft.com/en-us/office/database-design-basics-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5

MOSCHITTI, Alessandro, & QUARTERONI, Silvia. (2011). *Linguistic kernels for answer re-ranking in question answering systems. Information Processing & Management, 47(6), 825–842.* https://reader.elsevier.com/reader/sd/pii/S0306457310000518?token=B88315AC26AEE377540C92D62DD61BF3DE6908099C0F0F17E03929EC5B64B635FB8AC5BF74893D2178D6ABC0110295E6&originRegion=eu-west-1&originCreation=20220526194409

Pundge, A. M., Khillare, S. A., & Mahender, C. N. (2016). *Question answering system, approaches and techniques: a review. International Journal of Computer Applications, 141(3), 0975-8887.* https://www.academia.edu/25420092/Question_Answering_System_Approaches_and_Techniques_A_Review?auto=citations&from=cover_page

Slava Vaniukov. (2020). *Colors in UI Design: A Guide for Creating the Perfect UI - Usability Geek. Usability Geek.* https://usabilitygeek.com/colors-in-ui-design-a-guide-for-creating-the-perfect-ui/

WHO Director-General's opening remarks at the media briefing on COVID-19 - 11 March 2020. (2020, March 11). World Health Organization. https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020

Wong, E. (2018). *Shneiderman's Eight Golden Rules Will Help You Design Better Interfaces. The Interaction Design Foundation; UX courses.* https://www.interaction-design.org/literature/article/shneiderman-s-eight-golden-rules-will-help-you-design-better-interfaces

10.0 Appendix

Appendix 10.1

```
function getTime() {
    let today = new Date();
    hours = today.getHours();
    minutes = today.getMinutes();

    if (hours < 10) {
        hours = "0" + hours;
    }

    if (minutes < 10) {
        minutes = "0" + minutes;
    }

    let time = hours + ":" + minutes;
    return time;
}

// Gets the first message
function firstBotMessage() {
    let firstMessage = "Hello, How are you?"
    document.getElementById("botStarterMessage").innerHTML = '<p class="botText"><span>' + firstMessage + '</span></p>';

    let time = getTime();

    $("#chat-timestamp").append(time);
    document.getElementById("userInput").scrollIntoView(false);
}

firstBotMessage();

// Retrieves the response
function getHardResponse(userText) {
    let botResponse = getBotResponse(userText);
    let botHtml = '<p class="botText"><span>' + botResponse + '</span></p>';
    $("#chatbox").append(botHtml);

    document.getElementById("chat-bar-bottom").scrollIntoView(true);
}

//Gets the text text from the input box and processes it (original)
function getResponse() {
    let userText = $("#textInput").val();

    if (userText != "") {
        let userHtml = '<p class="userText"><span>' + userText + '</span></p>';

        $("#textInput").val("");
        $("#chatbox").append(userHtml);
        document.getElementById("chat-bar-bottom").scrollIntoView(true);

        setTimeout(() => {
            getHardResponse(userText);
        }, 1000)
    }

}

// Handles sending text via button clicks
function buttonSendText(sampleText) {
    let userHtml = '<p class="userText"><span>' + sampleText + '</span></p>';

    $("#textInput").val("");
    $("#chatbox").append(userHtml);
    document.getElementById("chat-bar-bottom").scrollIntoView(true);

    setTimeout(() => {
        getHardResponse(sampleText);
    }, 1000)
}

function sendButton() {
    getResponse();
}

// Press enter to send a message
$("#textInput").keypress(function (e) {
    if (e.which == 13) {
        getResponse();
    }
});
```