A SYSTEM FOR MEDICAL ASSISTANCE BY TEXT CONVERSATION

By

Mahin Rashid Chowdhury
Roll: 1907021

&

Jobair Al Nahian
Roll: 1907071

Supervisor:

Dr. K. M. Azharul Hasan

Professor

Dept. of Computer Science and Engineering

Khulna University of Engineering & Technology

____

Signature

Department of Computer Science and Engineering

Khulna University of Engineering & Technology

Khulna 9203, Bangladesh

November 2023

Acknowledgements

First and foremost, I must be grateful to and wish to acknowledge my insightful indebtedness to Dr. K. M. Azharul Hasan, Professor of the Department of Computer Science and Engineering and the supervisor of the project. His unfathomable knowledge in this field influenced me to carry out this project up to this point. His endless endurance, scholarly guidance, continual encouragement, constant and lively supervision, constructive criticism, and priceless suggestions made it possible to come up to this phase. Without his inspiring, enthusiasm, and encouragement, this work could not have been completed. Last, but by no means least, I thank Allah for the talents and abilities I was given that made it possible to undertake this project.

Next, We want to express gratitude to all of my class teachers who have shared their knowledge with us, which has aided us in completing our project. Also, a big thanks to my classmates for sharing their knowledge and always giving support, which helped us finish this project.

Abstract

In the medical field, it's often tough to get quick and reliable information about symptoms, health conditions, and what to do next. Many people can't easily reach healthcare professionals, and there's a growing number of health questions. This situation calls for a smart solution to make sure everyone can access the information they need.

Fixing the problem we talked about would be good for both doctors and regular people. If everyone can easily find accurate health information, it helps people make better choices about their health. This could also mean fewer trips to the emergency room for things that aren't super urgent. And overall, it helps everyone understand more about staying healthy.

The idea is to create a smart Medical assistant known as a chatbot, with expertise in medical information. This System uses advanced language skills and a dataset (100 x 4) of medical knowledge. It's designed to help people by giving them accurate and timely information about symptoms, conditions, and what steps to take. The System's design aligns to address the broader problem of limited accessibility to reliable medical information. Make it more simple.

For health-related queries, the chatbot is like having a health assistant. Its knowledge base contains current and reliable medical information, so users can rely on its advice. Chatting with the System enables people to make informed decisions and have a better understanding of their health.

Contents

List of Tables

List of Figures

1 Introduction

1.1 Background

In the world of health, sometimes it's hard to find the right information quickly. People have questions about symptoms, illnesses, and what to do to feel better, but they might not always have easy access to a doctor. That's where our smart friend, the System, comes in. It's here to help bridge that gap and make sure everyone can get reliable information about their health. Think of it like having a knowledgeable friend who can answer your health questions anytime, day or night. The idea is to solve the challenge of not always being able to find accurate and timely health information when you need it. This System is like a friendly guide in the world of health, ready to assist and provide trustworthy advice whenever someone has a health-related question. People may feel a little uncertain or concerned about their health in numerous circumstances, particularly if they find it difficult to get in touch with a medical practitioner. The System is intended to be a comforting and trustworthy information source. It's similar to getting a nice virtual health companion that comprehends your inquiries.

Ensuring that everyone, wherever they may be, has hassle-free access to clear and accurate health information is the challenge we're taking on. The System simplifies health information for all users by speaking your language like a human encyclopedia. People will be able to take control of their health and well-being and make knowledgeable decisions about it. The ultimate objective is to equip people with the knowledge they need to maintain their health and feel.

1.2 Objectives

- Create a Medical Assistant System: Develop a system that can answer health questions and help people understand possible medical issues.
- Make an Easy and Friendly Interface: Design a website that's simple to use, so anyone can find the health information they need without confusion.
- Understand Questions Better: Teach the System to comprehend what people say in a natural way, improving its ability to figure out possible health problems.

- Give Simple Info on Diseases: Ensure that the System shares clear and easy-to-understand details about what might be causing a health issue and what can be done about it.
- Maintain a database: The system maintains a database to store user information and past conversations. This ensures that users can easily refer back to previous responses, promoting reliability and continuity in interactions.

These objectives aim to build a Medical Assistant that's not only good at tech stuff but also makes it easy and helpful for everyone who needs health info, using regular language.

1.3 Scope

The System has a broad scope that includes a variety of elements to offer a customized and broad user experience. This is a summary of the project's scope:

- User Authentication and Profile Management: Users can log in to the website, creating individual profiles that allow for personalized interactions and secure storage of health-related data.
- Chat Interface: A user-friendly chat interface where users can communicate with the System by providing symptoms and asking health-related questions.
- Symptom Analysis: The System analyzes user-inputted symptoms to identify potential health conditions or diseases.
- Disease Information: Provides detailed information about identified diseases, including causes, symptoms, and preventive measures to enhance user awareness and understanding.
- Treatment Recommendations: Offers personalized treatment recommendations based on identified diseases, guiding users on potential courses of action or medical interventions.
- Conversation History: Saves and categorizes each user's conversation history, allowing users to review previous interactions, symptoms, and responses for ongoing health management.

1.4 Unfamiliarity of the problem

The unfamiliarity with the problem in the context of a medical chatbot lies in the challenge of limited accessibility to timely and accurate health information. Many individuals face difficulty obtaining quick and reliable answers about symptoms, conditions, and healthcare options. The unfamiliarity arises from the gap in easily reaching healthcare professionals and the increasing need for accessible and trustworthy health guidance. The medical chatbot aims to address this unfamiliarity by providing a user-friendly solution that offers reliable information, empowers users, and contributes to improved health literacy.

1.5 Project planning

Our plan for the medical Chatbot is like a roadmap showing how we'll turn our idea into reality. We want to solve the problem of people not easily getting health info. The plan includes features like checking symptoms, explaining diseases, and suggesting treatments. We've set a clear schedule and figured out what we need, like people and money, to make it happen. We're also thinking about possible problems and how to make sure everything works well. Our plan is like a guide, making sure we stay on track and create a helpful and easy-to-use System for everyone.

2 Related Works

- "**Maya - It's ok to ask for help**" [1]: An online android based application. It's a health care app which connects users with hundreds of doctors, psychiatrists and beauticians ready to serve 24 hours a day. Here user can write questions about his problems about health and the doctors' advice giving reply and facility to video call and can get digital prescription from doctors and there is extra facility to shop and read medical blogs.
- "**Text messaging-based medical diagnosis using natural language processing and fuzzy logic.**" [2]: The service focuses on assessing the symptoms of tropical diseases in Nigeria. Telegram Bot Application Programming Interface (API) was used to create the interconnection between the chatbot and the system, while Twilio API was used for interconnectivity between the system and a short messaging service (SMS) subscriber.

2.1 Gap in Existing Solutions

- "**Maya - It's ok to ask for help**" [1]: There are some limitations. In the app, the user has to wait for doctors, psychiatrists and beauticians to reply which is frustrating and may delay treatment. Manpower is a limitation here. In some cases, users have to pay for conversations with doctors.which may make these technologies less accessible to some patients.
- "**Text messaging-based medical diagnosis using natural language processing and fuzzy logic.**" [2]: Here a limitation is the conversation is based on question-answering. Its algorithm makes decisions by asking questions and based on the reply the bot gives the solution for diseases. So it is time-consuming and the user may get monotonous.

3 System Design

The proposed approach involves the use of a chatbot that can interact with users through text conversation, providing them with an avenue to express their concerns in a way that is comfortable and accessible to them. To respond to the user's queries, the system uses an expert model that uses advanced algorithms to provide intelligently and accurate responses.

3.1 Analysis of the system

The system is analyzed through a couple of diagrams to depict the procedures and interactions of various elements. It follows with DFD and a Use Case diagram to exhibit the flow of working steps.

3.1.1 DFD Diagram

The user input, which usually consists of a list of symptoms connected to a specific medical condition, is analyzed by the System using advanced algorithms. To produce an accurate result, these algorithms are designed to process and interpret the user's input. Working with large amounts of data or sensitive information requires an efficient understanding of the system flow. As such the movement of data is shown in the Data Flow Diagram (DFD), figure 3.1

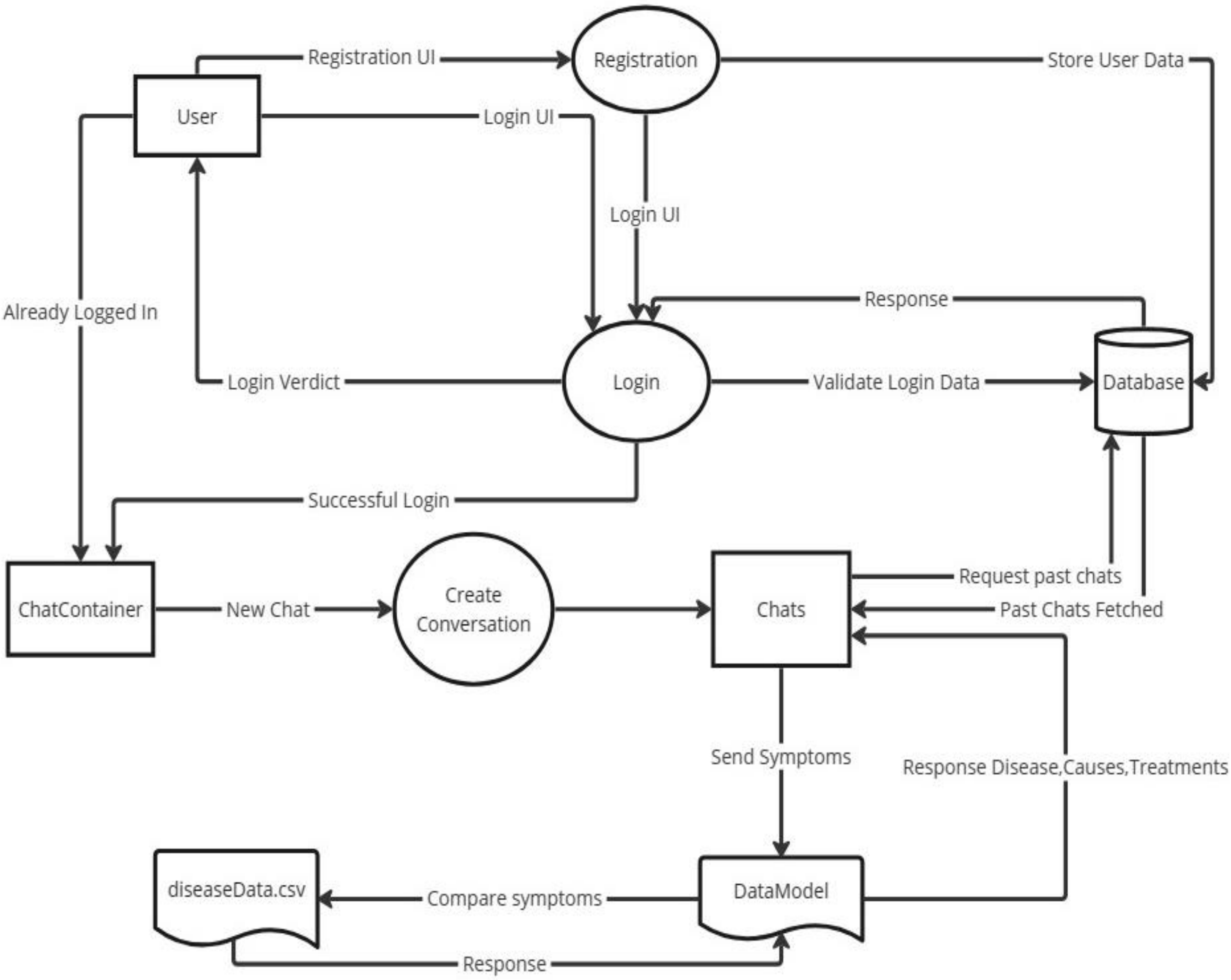

Figure 3.1: Data Flow Diagram of the medical Assistant system.

3.1.2 Use Case Diagram

There are many different types of roles in the system. The figure 3.1.2 illustrates these roles' points of view

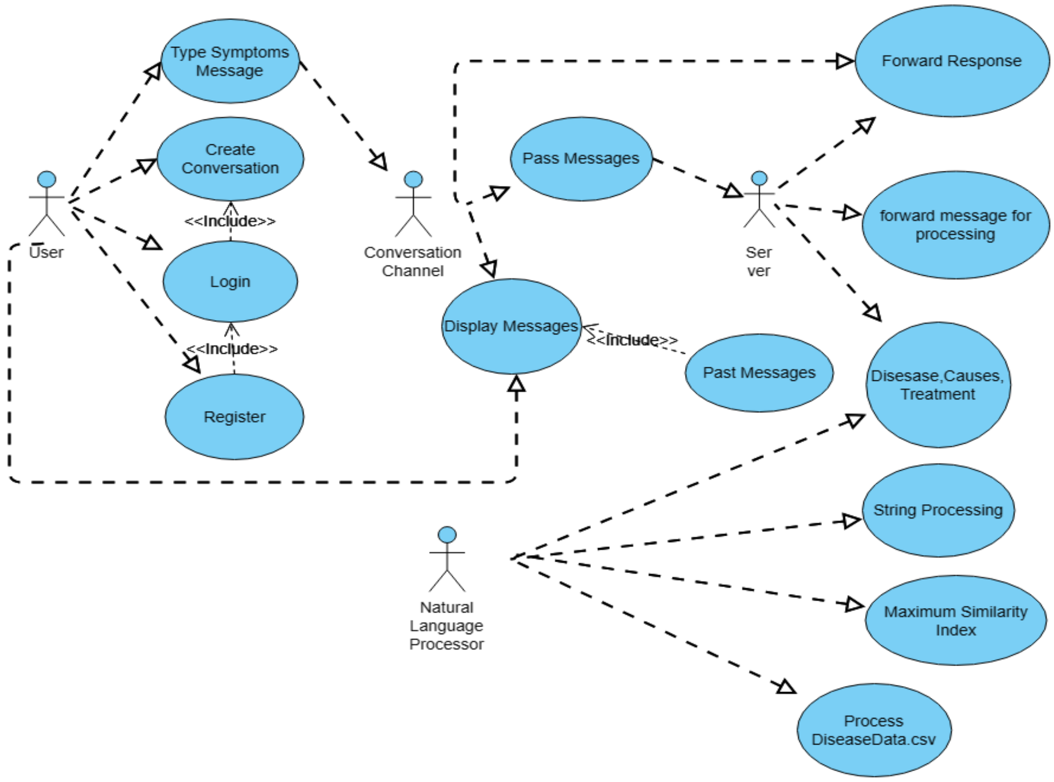

Figure 3.2: Use case diagram of Medical Assistant System

3.2 System architecture

The project's architecture must be fully understood in order to build it. A class diagram displays the project's structure by encapsulating and associating important classes, while an ER diagram displays the database that houses the information and the relationships that are related to it.

3.2.1 Class Diagram

The information and functionality of association and encapsulation of each individual class or building block is shown in the figure 3.3. where the necessary information is reflected by the blocks.

3.2.2 Schema Diagram

The schema diagram is shown at figure 3.4 where structured query language is used.

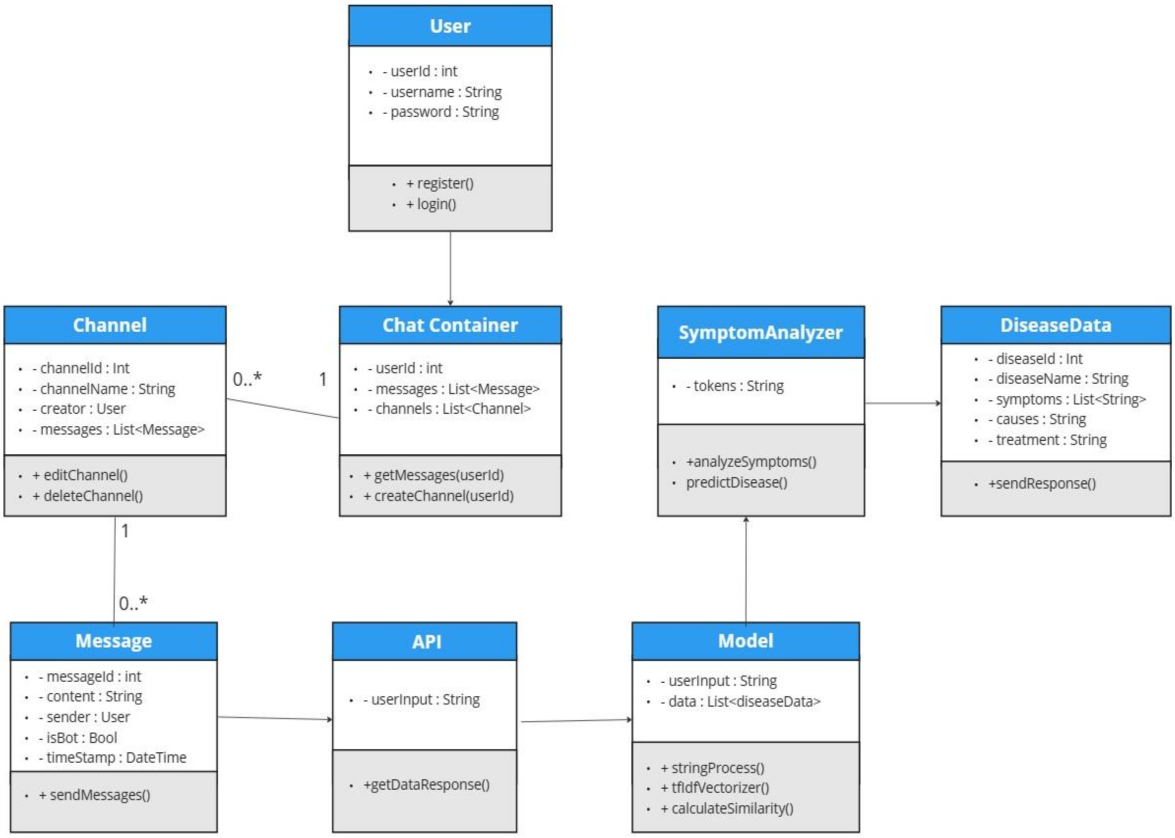

Figure 3.3: Class Diagram of Medical Assistant System.

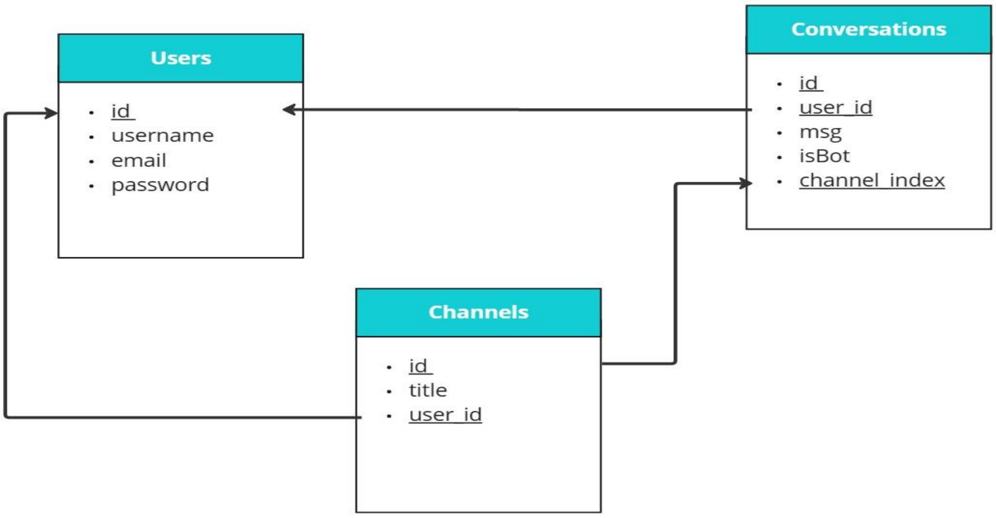

Figure 3.4: Schema Diagram for Medical Assistance System.

3.3 Tools used

To finish the system, many different types of tools were employed. Every tool contributed significantly to the project's success.

3.3.1 Flask

Flask is a lightweight Python web framework that provides useful tools and features for creating web applications in the Python Language. It gives developers flexibility and is an accessible framework for new developers because you can build a web application quickly using only a single Python file. [4]

3.3.2 Google Colab

Colaboratory, or "Colab" for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing access free of charge to computing resources including GPUs. [5]

3.3.3 React

When using React to create interactive websites, styles such as Tailwind CSS or Bootstrap are frequently used to make the website responsive. By writing styles that adapt to screens and starting with mobile design, by using React website will look good and work smoothly, whether on a computer, tablet, or phone.

3.3.4 VS Code

One useful code editor is Visual Studio Code. Coding productivity is increased by the many extensions available for web development languages and Python.

3.3.5 MySQL

It is an open-source relational database management system (RDBMS). MySQL is widely used for managing and organizing data within databases. It uses a structured query language (SQL) to interact with databases, making it a popular choice for web applications that require data storage and retrieval. MySQL is known for its speed, reliability, and ease of use. It is often used in conjunction with programming languages like PHP, Python, and others to create dynamic and data-driven websites.

3.3.6 Apache

Apache is one of the most widely used web servers globally. Its primary function is to serve web pages and handle HTTP requests from clients (such as web browsers).

3.3.7 Python Libraries

Pandas is a Python library that simplifies data manipulation through its Data Frame structures, enabling efficient cleaning and analysis of structured datasets. SpaCy is a Python library for natural language processing, renowned for its speed and accuracy in tasks such as recognizing entities and tagging parts of speech. Scikit-learn is a comprehensive machine learning library for Python, offering a wide range of algorithms for tasks like classification, cosine similarity, and TF-IDF Vactorization.

4 Project Implementation

The procedures and individual steps that were taken to finish the system project will be covered in this chapter.

4.1 System implementation

The system being a web application a website was designed using Flask as the backend and React basic HTML, CSS as the frontend. The NLP technique was used for the model training and pre-trained model Libraries was used for the task. The dataset for the task was created from different websites. The details of these procedures are as following,

4.1.1 Web Implementation

The frontend implementation is done through the help of a Javascript library knows as React. For styling Tailwind CSS framework was used. Also some vanilla CSS was also used for custom styling. The backend of the website is done using python framework Flask. Navigation, Routing and API requests were handled by Flask. For the database system, a structured query language MySQL was used. The website also includes light mode and dark mode for better visibility on terms of user preference,

4.1.2 Model Implementation

The Model implementation includes Tokenization, TF-IDF Vactorization, and Cosine Similarity etc.

4.1.2.1 Tokenization

As we know, computer systems cannot directly understand human language. That's why Natural Language Processing (NLP) can help computer systems to understand the language that humans use and classify them, as well as analyze them if the dialog needs response.One of the important functions of NLP is Tokenization, removing stopwords, and lemmatization. This technique involves identifying specific characters or symbols such as spaces, punctuations, and special characters that separate different parts of the text, and then using them as markers to divide the text into separate words. This simplifies the text analysis process as each word can be examined individually. Furthermore, punctuation marks are removed to make the text even simpler to process. After the text is broken down into Individual words, it can be further analyzed to extract patterns or understand the meaning of the text. We have Spacy and NLTK library to perform text tokenization. Preprocessing is used to remove words that are often used like 'is', 'like' etc. Lemmatization is then used to obtain the base word of each token.

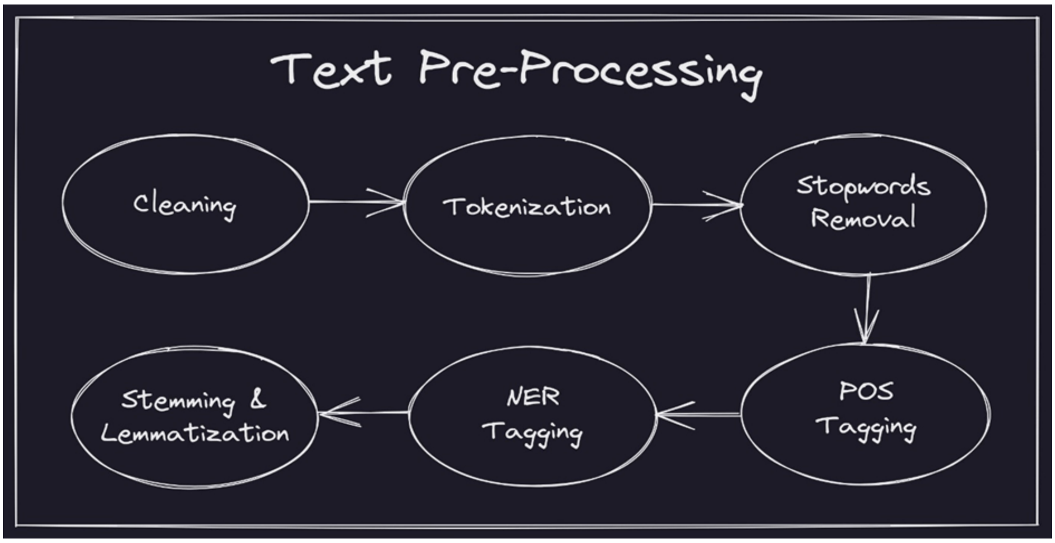

Figure 4.1: Text Pre-processing steps

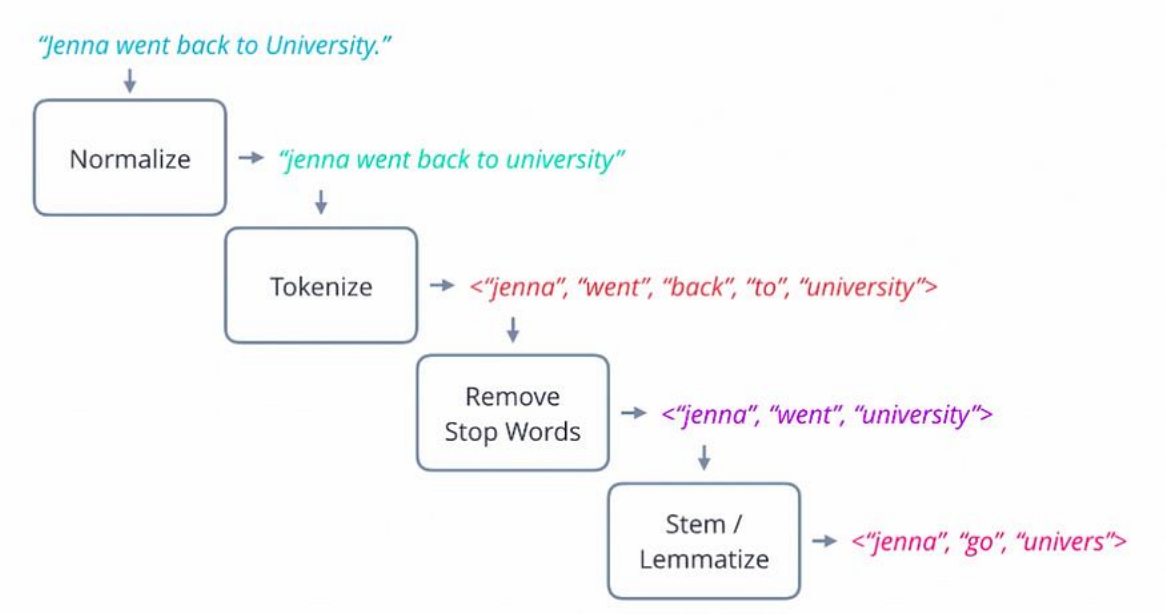

Figure 4.2: Text Pre-processing steps with example.

4.1.2.2 TF-IDF

TF-IDF stands for Term Frequency - Inverse document Frequency, it's a simple method to try to give scores to documents that look the same as the user's query. It is built over two main concepts:[6]

- How many times this word (in the query) appeared in the each document => Term Frequency.
- How unique is this word => Inverse document frequency.

$$TF(t,d) = \left(\frac{Total\ Number\ of\ time\ term\ \mathbf{t}\ is\ present\ in\ doc\ A}{Total\ number\ of\ tokens\ in\ doc\ A}\right)$$

$$IDF(t) = log\left(\frac{Total\ Documents}{Number\ of\ documents\ term\ \mathbf{t}\ is\ present\ in}\right)$$

$$TF - IDF = TF(t,d) * IDF(t)$$

Figure 4.3: TF – IDF calculation formulae.

4.1.2.2 Term Frequency:

It is a simple count of how many of a specific word appeared in a given document, we count all words in all given documents => to give scores to documents.[6]

Example:

Here, in this example, we have multiple documents, here they are Shakespeare's plays, we test for specific word occurrences through these documents, as here when only working on Term Frequency, we are only interested in word counts. Now we would need to measure score just according to term frequency (word count), in coming sections, we would leverage the way we calculate the score but Raw term frequency is not what we want as: A document with 10 occurrences of the term is more relevant than a document with 1 occurrence of the term. But not 10 times more relevant. As Relevance does not increase proportionally with term frequency.

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	73	0	0	0	0
Brutus	4	157	0	1	0	0
Caesar	232	227	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	2	0	3	5	5	1
worser	2	0	1	1	1	0

Figure 4.4: Term Frequency Example.

. So we would use Log frequency not just simple word count:

$$w_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{tf}_{t,d}, & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases}$$

But TF is just word count, so whenever TF > 0, we would calculate score as 1+Log, else if word count is zero (TF = 0), score would be zero. So now to calculate score.

$$score = \sum_{t \in q \cap d} (1 + \log \mathrm{tf}_{t,d})$$

Term Frequency is not enough but we must account for how rare a certain word is if compared to other words through all documents. So to complete TF-IDF we must take into consideration document frequency, or to be more precise it is inverse to take into consideration the rarity of words.

4.1.2.3 `Inverse Document Frequency

Here, we tend to measure rarity of words, this is done to calculate how many documents a word appeared in. We count a word appeared once in a document once seen in this document (even if it appeared multiple times in this document), as our main goal here is counting documents not words.[6]

Example: Assume a million document and you look for a list of words and see how many documents these words appeared in.

Table 4.1: Looking for a list of words out of million document

term	df_t
calpurnia	1
animal	100
sunday	1,000
fly	10,000
under	100,000
the	1,000,000

df here stands for number of documents. So here a word like “calpurnia“ appeared in only one document (*very rare*) so it would have high score, so a small df (document count) gets a big score, so we get the inverse of document frequency :

$$\mathrm{idf}_t = \log_{10} (N/\mathrm{df}_t)$$

So by calculating this for all the above terms (words), results would be:

Table 4.2: Example of Inverse Document Frequency.

term	df_t	idf_t
calpurnia	1	$\log_{10} 1000000/1 = 6$
animal	100	$\log_{10} 1000000/100 = 4$
sunday	1,000	3
fly	10,000	2
under	100,000	1
the	1,000,000	0

As you have seen here, IDF (inverse document frequency) is independent of the query as it only depends on documents, so we don't need to calculate idf for each new query, as there is one idf value for each term. This can be used for optimization, calculating idf prior to query. So if we want to measure score just according to IDF (inverse document frequency):

$$\text{score} = \sum_{t \in d \cap q} \log_{10}(N/\text{df}_t)$$

4.1.2.4 Combining both TF-IDF:

We first calculate weight for a specific term in a specific document.

$$w_{t,d} = (1 + \log \text{tf}_{t,d}) \times \log_{10}(N/\text{df}_t)$$

Applying this to the previous example of Shakespeare's Play:

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0.35
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95

Figure 4.5: Example of combining TF-IDF

Now each document is represented by a real-valued vector of tf-idf weights.

Now to calculate score of a specific query versus a specific document.

$$\text{Score}(q,d) = \sum_{t \in q \cap d} \text{tf.idf}_{t,d}$$

Here, t = term (word) , q = query , d = document

So you calculate the score for all terms (t) in the query (q) (as tf =0 if this term wasn't found in the given document). And you sum scores for all terms in the query to get a score of this document.

4.1.2.5 Cosine Similarity [6]

There are many ways to use the concept of vector space, but we specifically use Cosine similarity as when using vector space, we would face a problem, which is different length documents would result in wrong scores (as discussed before), to solve this problem, we must consider using the concept of Length normalization, so this is the reason why we use cosine similarity

A vector can be length normalized by dividing each of its components by its length – for this, we use the L2 norm:

$$\|\vec{x}\|_2 = \sqrt{\sum_i x_i^2}$$

Now let's really compute the similarity using cosine similarity:

$$\mathit{Cosine\ Similarity} = \frac{A \cdot B}{\|A\|\|B\|}$$

$$\mathit{Cosine\ Similarity} = \frac{\|A\|\|B\| * Cos(\theta)}{\|A\|\|B\|}$$

Figure 4.6: Formulae for Cosine Similarity.

4.1.3 Model Architecture

The Workflow of the Model given below in Figure 4.1.

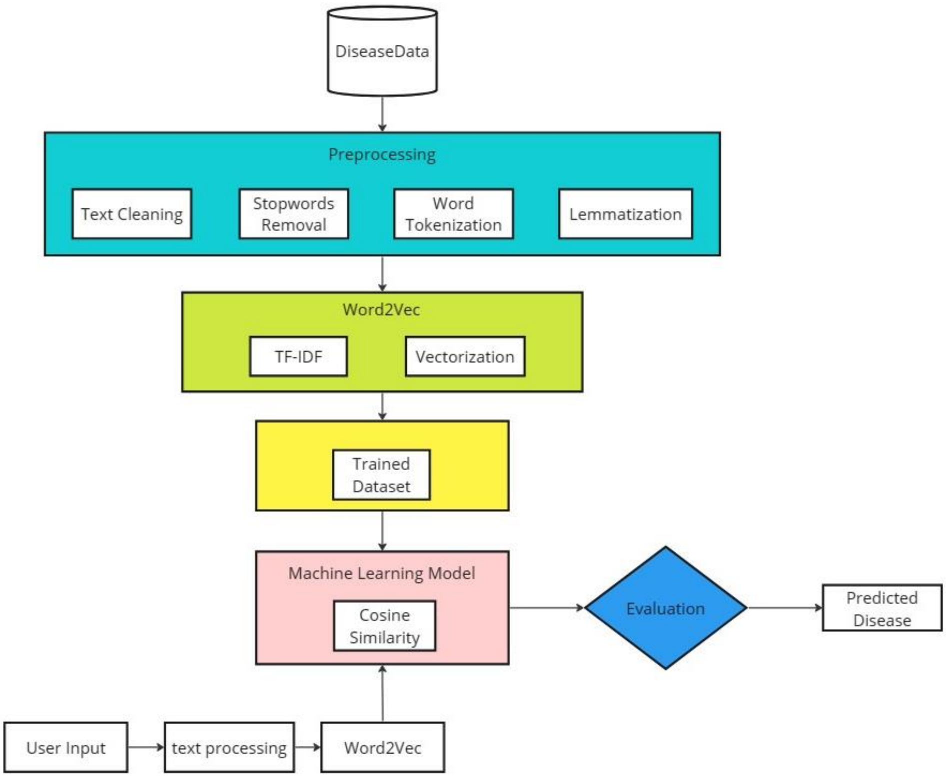

Figure 4.7: Workflow of System detecting diseases category task.

4.1.4 Dataset

	A	B	C	D
1	Symptoms	Disease	Causes	Treatment
10	continuous low mood or sadness feeling hopeless and helpless having low self-esteem feeling tearful feeling guilt-ridden feeling irritable and intolerant of others having no motivation or interest in things finding it difficult to make decisions not getting any enjoyment out of life feeling anxious or worried having suicidal thoughts or thoughts of harming	Depression	Illness Alcohol and drugs Loneliness Menopause Pregnancy and giving birth Family history Personality Stressful events	If a GP diagnoses you with mild depression, they may suggest waiting a shor Guided self-help Your GP may suggest trying guided self-help to see if it can help with your de There's evidence that exercise can help depression, and it's one of the main If you have mild depression that's not improving, you may find a talking thera Antidepressants are medicines that treat the symptoms of depression. There take a course of antidepressants plus talking therapy If you have severe depression, you may be referred to a mental health team Talking therapies Cognitive behavioural therapy (CBT) Interpersonal therapy (IPT)
11	feeling very thirsty peeing more frequently than usual, particularly a feeling very tired weight loss and loss of muscle bulk itching around the penis or vagina, or frequent e blurred vision	Diabetes	are living with overweight or obesity do not have a healthy diet have a family history of type 2 diabetes are of Asian, Black African or African Caribbean origi take certain medicines such as steroids for a long tin have high blood pressure have had gestational diabetes during pregnancy Diabetes can damage the nerves in your feet and ca If you have diabetes, your eyes are at risk from diabe	Medications (e.g., metformin medicine), diet control, exercise Regular monitoring of blood glucose levels is essential for individuals with Ty Many individuals with Type 1 diabetes use carbohydrate counting to help ma A balanced diet is important for people with Type 1 diabetes. Meeting with a Regular physical activity is beneficial for managing blood sugar levels. Stress, illness, and changes in routine can affect blood sugar levels. Insulin Therapy
12	a high temperature difficulty breathing – babies may make grunting being sleepier than normal or difficult to wake blue, grey, pale or blotchy skin, lips or tongue – a rash that looks like small bruises or bleeding u a headache a sore throat swollen or painful joints	Haemophilus influenzae type b (Hib	infection of the lining of the brain and spinal cord (me a severe immune reaction (sepsis) infection of part of your windpipe (epiglottitis) infection of the lungs (pneumonia) You can become very ill with these conditions and wi	Vaccination is the best way to prevent serious Hib infections. Vaccines that protect against Hib are offered to babies as part of the NHS va The main treatment for a Hib infection is antibiotics. You'll usually need to be You may be given oxygen and other medicines such as steroids, depending If you have a Hib infection, your close contacts, such as anyone you live with
13	blood pressure is high Severe headaches headaches in back of the head		are overweight eat too much salt and do not eat enough fruit and ve do not do enough exercise drink too much alcohol or coffee (or other caffeine-ba smoke have a lot of stress	cut your salt intake to less than 6g (0.2oz) a day, which is about a teaspoonfu eat a low-fat, balanced diet – including plenty of fresh fruit and vegetables; g be active – read some tips about getting more exercise cut down on alcohol – get tips on cutting down, including downloading a drin lose weight – find out what your ideal weight is using the BMI healthy weight drink less caffeine – found in coffee, tea and cola

Figure 4.8: Dataset for Medical Assistant System.

Dataset is very important for training a model. We have created a dataset of 100 common diseases .We have collected about diseases, symptoms, causes and treatment. The diseases collected from *NHS* website. [7] .A Demo is shown below in Figure 4.1

4.2 User Manual

The GUI is added of different interface.

4.2.1 Home Page

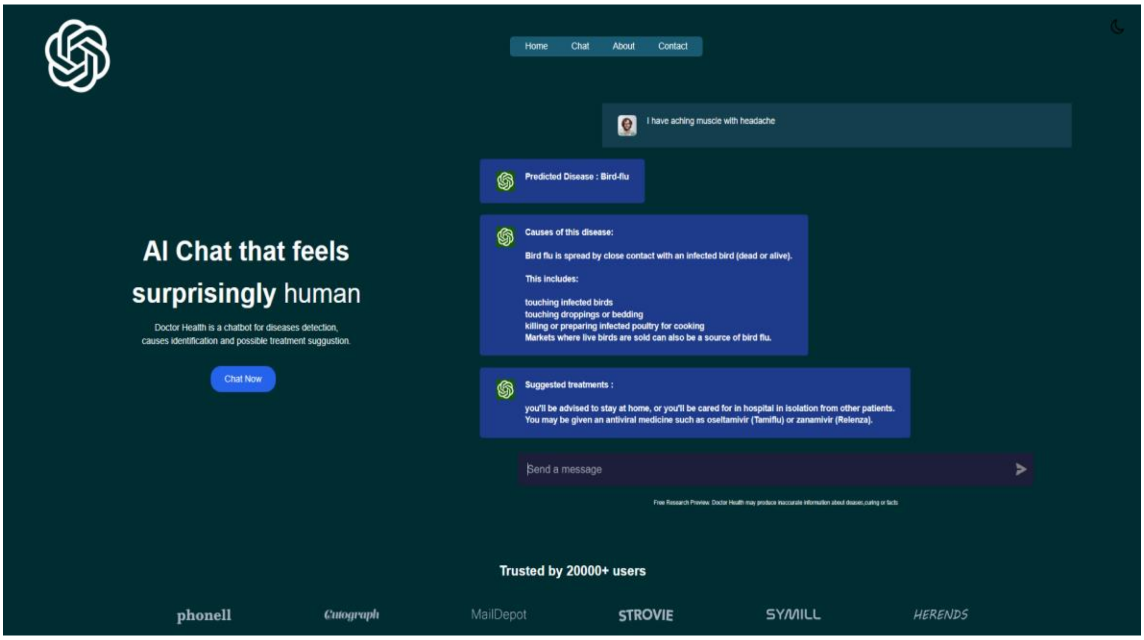

Figure 4.9: Website Home Page

First the user will see the home page. From the home page user can redirect to Home, Chat, About, Contact page through the navigation bar. To use the system user have to click the “Chat Now” button which will redirect to Login Page for user authentication (will directly redirect to chat page if user authentication is done already.

4.2.2 Login Page

User need to enter valid Email and Password to complete user authentication. If User isn’t already registered he need to go to the register page and complete registration. The E-mail address need to be a valid email address and password must be strong that is length is minimum 6 characters.

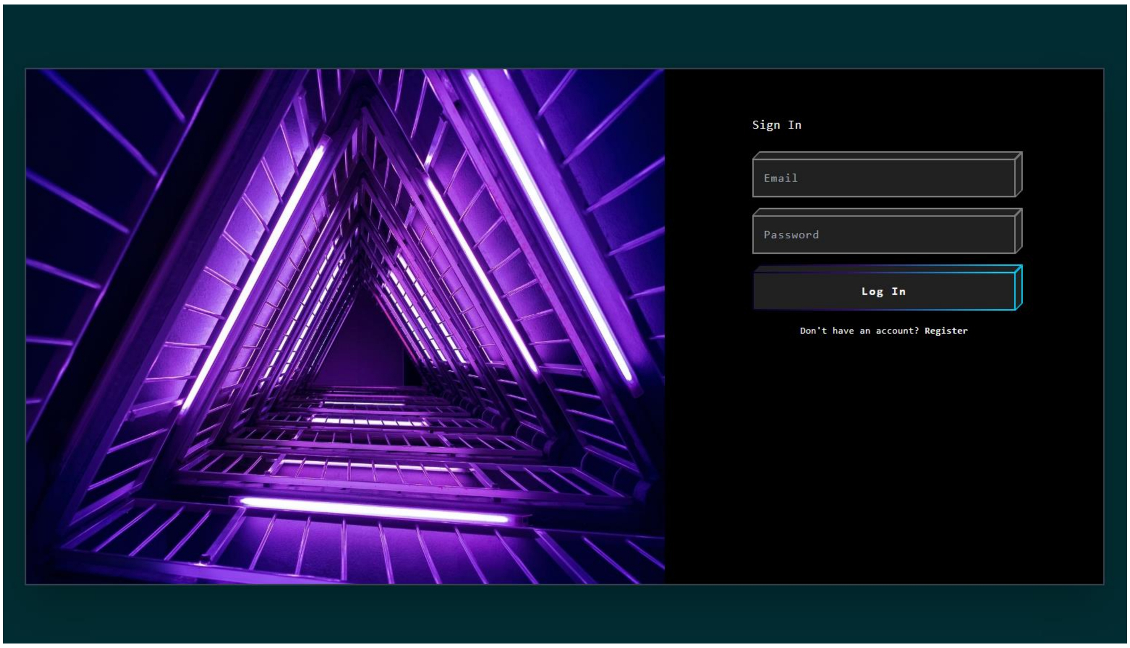

Figure 4.10: Website Login Page

4.2.3 Register Page

To complete registration, an user need to provide username, valid email and password in the corresponding boxes. The E-mail address need to be a valid email address and password must be strong that is length is minimum 6.

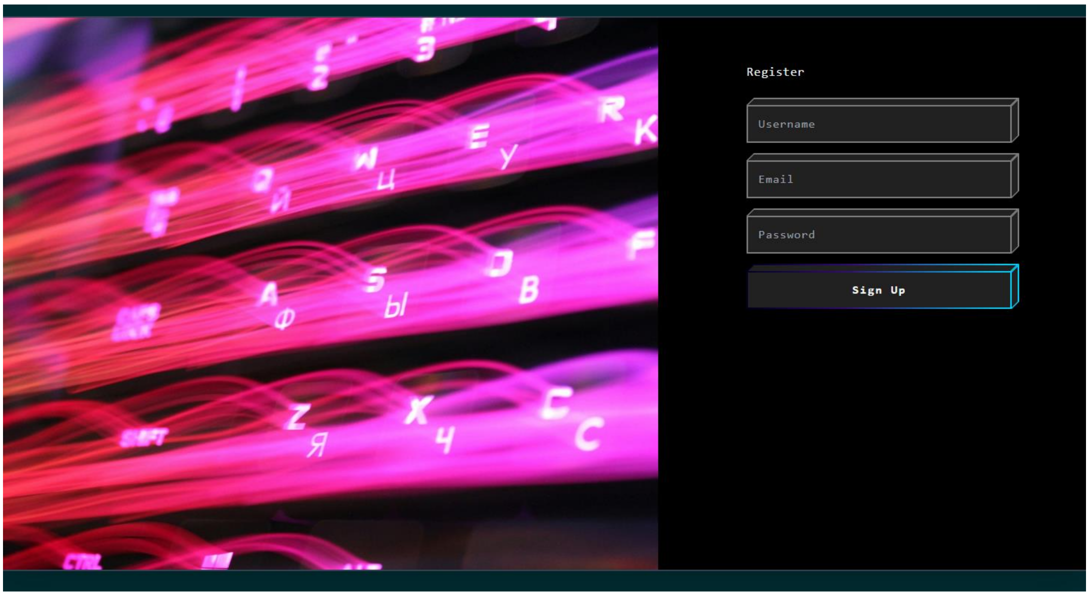

Figure 4.11: Website Register Page

4.2.4 Chat Page

Figure 4.12: Chat interface page

At first an user need to create a new channel, to do it he need to click the "New Chat" button. It will create a new chat channel named "New Conversation".

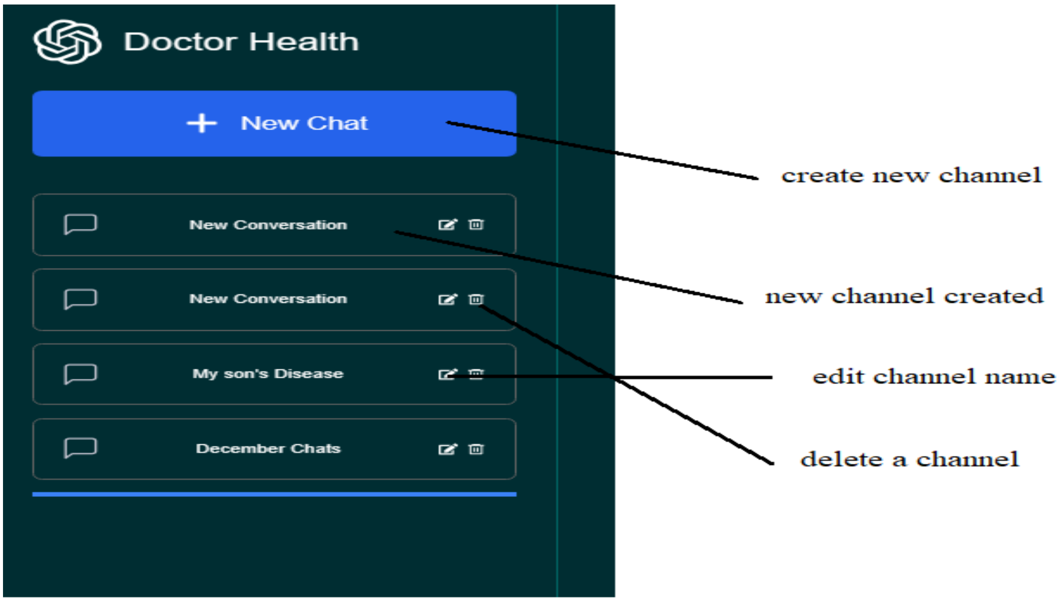

Figure 4.15: Creating New Channel

Each channel can be renamed and deleted. This is to provide user a better experience for managing chats, instead of stacking everything together. After entering a channel the user need to type about his symptoms in detail on the text box to get his disease prediction. It is recommended to provide every symptoms of a patient in detail to get better result. User must select a channel to start texting.

4.2.5 Disease Prediction

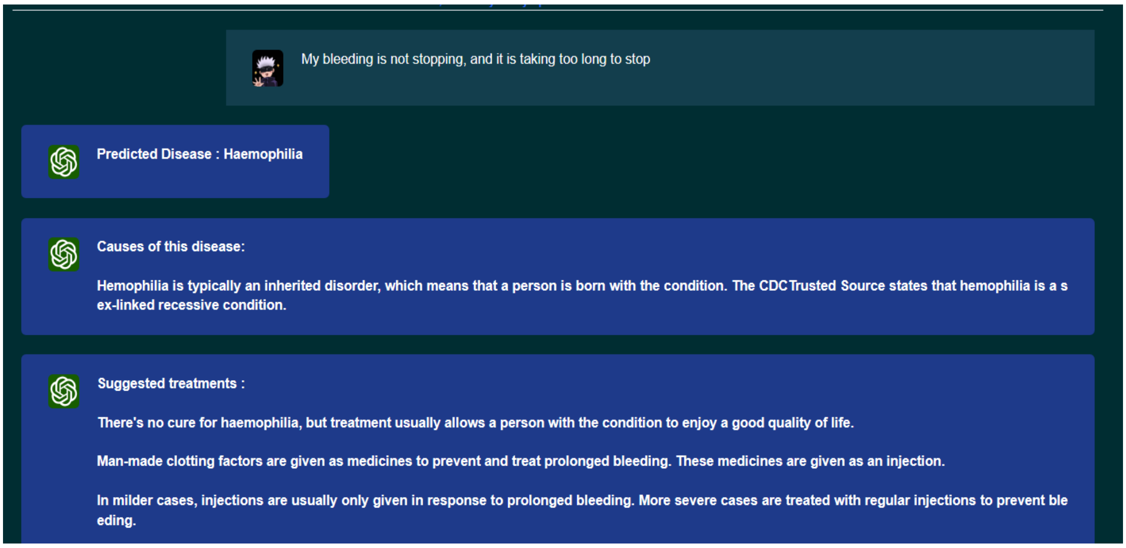

Figure 4.16: Disease Prediction of the system

After using the system an user can logout and exit from the website. The system also has dark mode-light mode feature for better visibility on terms of user preference.

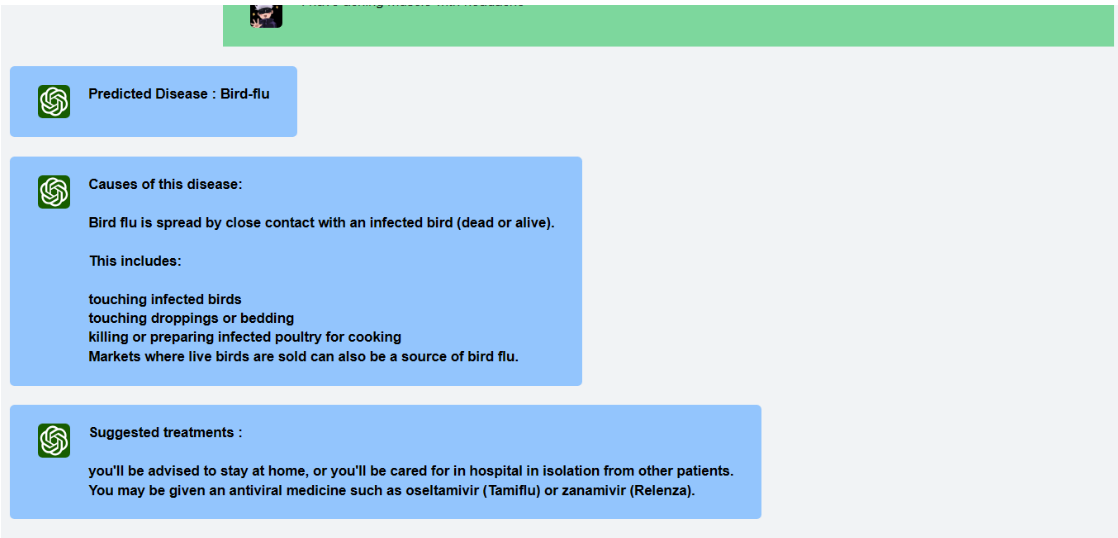

Figure 4.17: Disease Prediction of the system on Light mode

4.3 Morality or Ethical Issues

Many different sources were used in the development of the project. The website from which the dataset was gathered [7]. Several libraries, including the Spacy [8], Pandas, and Scikit-learn (sklearn) libraries for the TfidfVectorizer and cosine similarity functions [6], were used to create the model. The project's suggested technique was picked up from some papers.[2][3][10][11][12] and articles.[6] Images and icons used were gathered from open-source websites. For the web application, smooth productivity was ensured by adhering to the Flask documentation [4]. The class diagrams and database system implementation were taken from earlier academic curricula. Personal information about its users is also included in the system, but it is fully guaranteed that the information or data will be kept private from others. The information will remain private and secure within the system, with no access granted to third parties. The project respects moral principles, giving users' privacy top priority.

4.4 Socio-economic impact and sustainability

- A medical chatbot can enhance accessibility to healthcare information. People in remote or underserved areas may benefit by obtaining medical advice and information without the need for physical presence.
- Chatbots can assist in early detection of symptoms and provide preventive measures. This can lead to early intervention and improved health outcomes. It can be used for Educational purposes also for medical students and practitioners.
- Chatbot poses a risk of providing incorrect advice or misdiagnosing conditions, potentially leading to delayed or inappropriate medical interventions.
- Timely advice and information from a medical Assistant System may reduce unnecessary visits and thereby lowering overall healthcare costs for individuals and the healthcare system.
- It is essential to conduct routine updates and enhancements to the System's knowledge base and functionality to guarantee its applicability and efficacy in furnishing precise and current medical information.

4.5 Financial analyses and budget

Financial analysis and budget for this project involves the cost of development, equipment, implementation, maintenance and other relevant expenses. A general measurement of the budget has been estimated in following table 4.1. The budget as a whole is distinguished by low external spending, the bulk of which are self-funded, and the use of readily available software and technologies. The total time of the whole project is divided into different sections. Here, we show a gantt chart indicating number of days it required particularly for every task It is given in the figure 4.18.

Table 4.3: Financial analysis and budget of the project

Types of cost	**Budget(tk)**
Personnel Salaries	50,000 – 70,000
Online Services (python packages)	10,000 – 20,000
Software (Tools used)	10,000 – 20,000
Data collection and processing (creating dataset)	5,000 – 10,000
Development tools	5,000 – 10,000
Project management tools	5,000 – 14,000
Miscellaneous	5,000 – 6,000
Total:	90,000 – 1,50,000

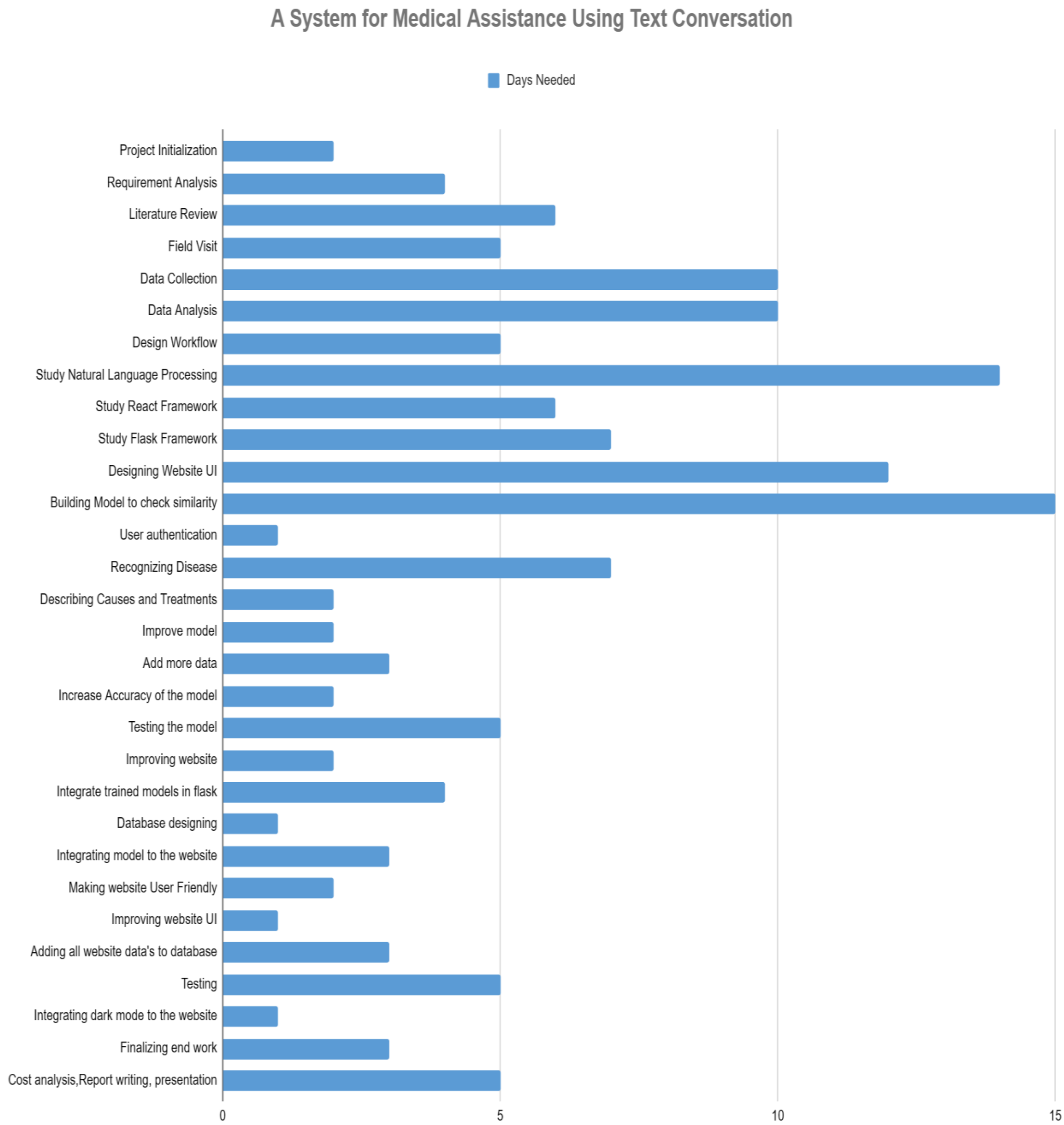

Figure 4.18: Gantt Chart of Medical Assistant System Project

5 Conclusion

A chatbot is an artificial intelligence (AI) program that can talk with users via text or voice commands by using natural language processing (NLP). Chatbots can be made to mimic human-like interactions with users, answering their questions quickly and precisely.

5.1 Conclusion and challenges faced

The platform provides users with an easy-to-use means of having enlightening discussions about various illnesses. Its advanced Natural Language Processing (NLP) capability, which deftly extracts base words and comprehends the user's actual symptoms, excels in producing accurate results by providing detailed symptoms. The process of developing this effective system wasn't without its difficulties. One significant obstacle was the lack of detailed descriptions provided by users, which negatively impacted the model's ability to accurately detect diseases. To overcome this obstacle, multiple models had to be trained on different datasets. Moreover, precision was hampered by the limitations of a big dataset. Another obstacle in the realm of web technology surfaced during the integration of models, where prolonged loading times and UI/UX design issues needed addressing. Despite these challenges, the team confronted them directly, ensuring the project's overall success and reinforcing the system's efficacy in delivering valuable health information.

5.2 Future Study

This model can be further made to broaden its scope. To increase the efficiency of medical chatbots, more word combinations can be added to their databases, allowing them to manage a wider range of ailments. By adding more data and improving algorithms, smedical chatbot will become increasingly accurate in diagnosing and treating patients. Chatbots can be integrated with various messaging platforms, such as Facebook Messenger, WhatsApp, and Slack, making them accessible to users on their preferred messaging platform. The use of chatbots has become increasingly popular in recent years, and it is expected to continue to grow as businesses and organizations look for ways to improve customer engagement and streamline their operations.

5.3 Limitation

- It exclusively focuses on symptoms, lacking the ability to engage in general conversations.
- Accuracy heavily depends on precise symptom matching, potentially resulting in inaccuracies for less typical inputs.
- The system is restricted to medical contexts and may struggle with non-medical conversations.

- Predictions are reliant on the quality and specificity of user-provided symptoms.
- Inconclusive matches prompt the system to request additional details, impacting user experience.
- Lack of detailed explanations for predictions may reduce transparency and user trust.

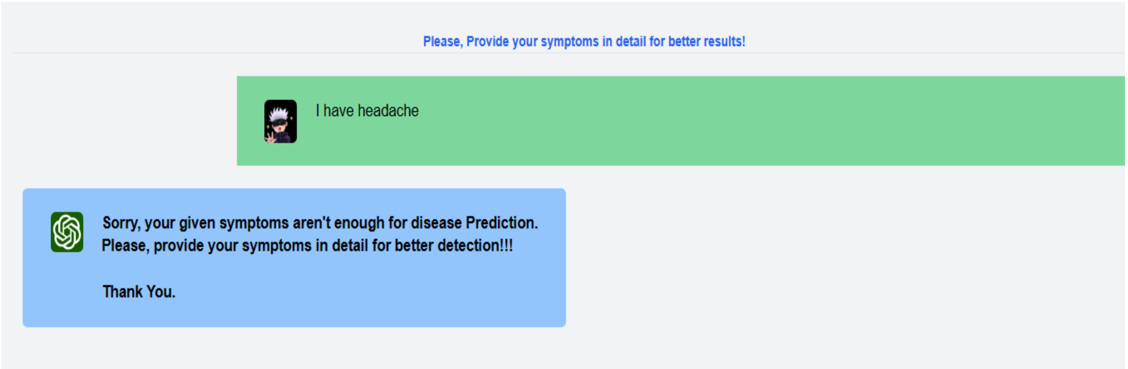

Figure 5.1: System requesting for more precise input

References

[1] https://play.google.com/store/apps/details?id=com.maya.mayaapaapp&hl=en&gl=US

[2] Kevin, Bruce, Brikes VIkin, and Manju Nair. "BUILDING A CHATBOT FOR HEALTHCARE USING NLP." (2023).

[3] Omoregbe, Nicholas AI, et al. "Text messaging-based medical diagnosis using natural language processing and fuzzy logic." *Journal of Healthcare Engineering* 2020 (2020): 1-14.

[4] https://www.digitalocean.com/community/tutorials/how-to-create-your-first-web-application-using-flask-and-python-3

[5] https://research.google.com/colaboratory/faq.html#:~:text=Colaboratory%2C%20or%20%E2%80%9CColab%E2%80%9D%20for,learning%2C%20data%20analysis%20and%20education.

[6] https://www.codeproject.com/Articles/1204813/Lina-ChatBot-Generating-Response-Using-Document-Re

[7] https://www.nhs.uk/conditions/

[8] https://spacy.io/models

[10] P. I. Prayitno, R. P. Pujo Leksono, F. Chai, R. Aldy and W. Budiharto, "Health Chatbot Using Natural Language Processing for Disease Prediction and Treatment," 2021 1st International Conference on Computer Science and Artificial Intelligence (ICCSAI), Jakarta, Indonesia, 2021, pp. 62-67, doi: 10.1109/ICCSAI53272.2021.9609784.

[11] V. Gupta, V. Joshi, A. Jain and I. Garg, "Chatbot for Mental health support using NLP," 2023 4th International Conference for Emerging Technology (INCET), Belgaum, India, 2023, pp. 1-6, doi: 10.1109/INCET57972.2023.10170573.

[12] J. E. Christopherjames et al., "Natural Language Processing based Human Assistive Health Conversational Agent for Multi-Users," 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2021, pp. 1414-1420, doi: 10.1109/ICESC51422.2021.9532913.