

Chatbot for Patient Screening

Group Members:

1. Bakshi Shantanu (B19CSE019)
2. Barun Shakya (B19CSE020)

Mentor:

Dr. Sumit Kalra

I. Abstract

As we have noticed, patients have to wait outside the doctors cabin before actual diagnosis. Also doctors ask some screening questions to the patients before actual diagnosis. In this project we have developed components to help the doctors in initial pre consultation screening of the patient. We have developed two components that will help doctors patient screening. The first one is chatbot. The chatbot is a type of adaptive chatbot that will take input as problems/symptoms patients are experiencing. These inputs can be pertaining to a symptom that the patient is experiencing or a sense of discomfort a patient is experiencing in a certain body part . The chatbot will then ask questions related to these symptoms to the patients. The questions asked along with responses can be used by a doctor to know the problems that the patient is experiencing. This will utilize the patient waiting time and the doctor can screen the patient by looking at the responses. This will help the doctor in diagnosing the patient in less time .This would also reduce the amount of time a doctor spends asking repetitive questions about a certain condition .

We also built an interface through which the patient can mark the locations at which they are having pain. This functionality would complement our chatbot by allowing the patient to demarcate the area where he/she is feeling pain or irritation .

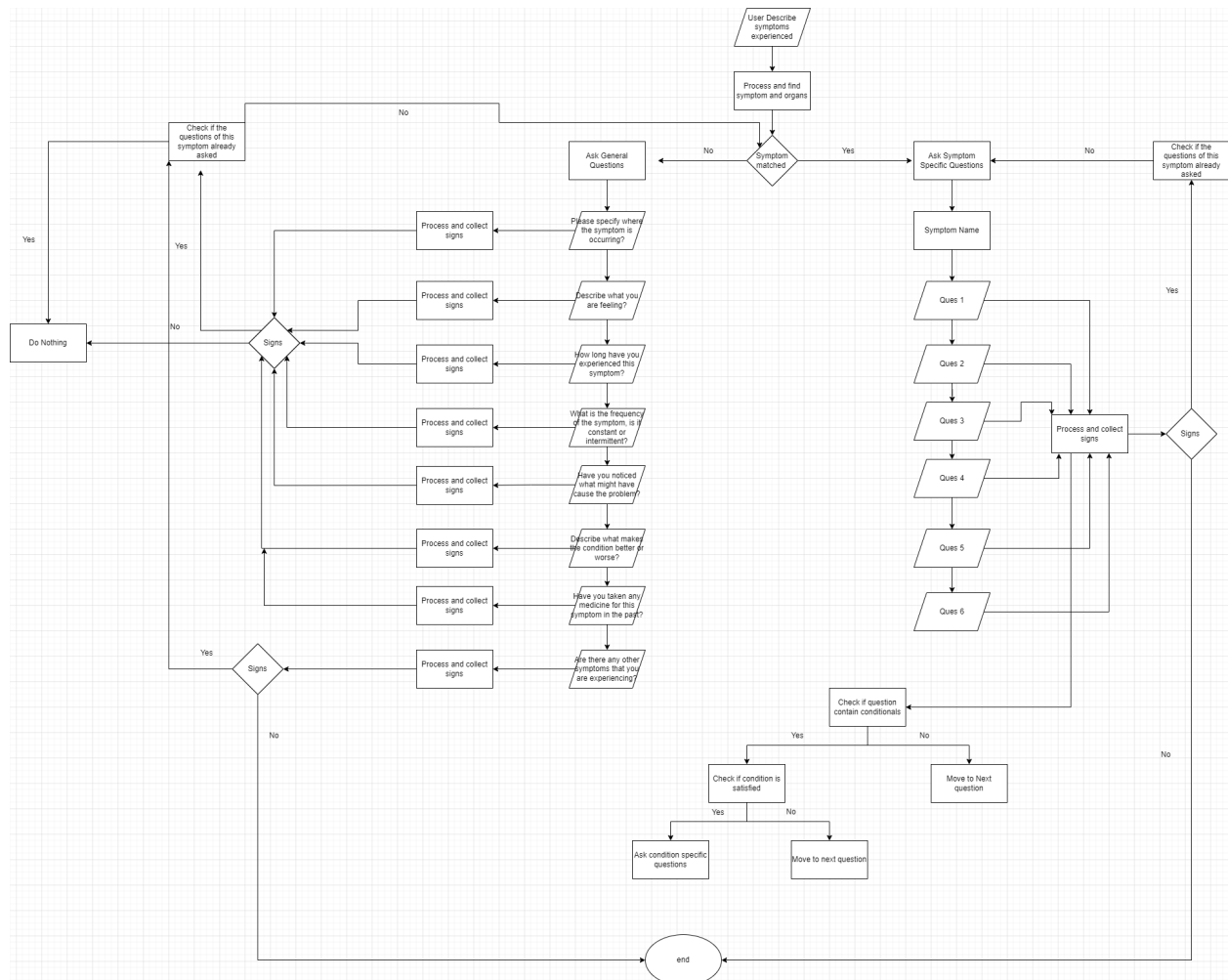
II. Methodology:

1. Chatbot

The High level flow is as follows:

- The user input is first preprocessed using various preprocessing techniques. The preprocessing step involves removal of white spaces, lemmatization of words, removal of stopwords, removing punctuations, etc.
- The phrases of symptoms are extracted from text using the spacy library. TextRank pipeline is used to extract words and phrases.
- The extracted symptoms are then matched into the dataset, and if the symptom is present in the dataset then the questions of this symptom are asked.
- If extracted symptoms do not match any of the symptoms in the dataset, then the dataset contains some general set of questions. The chatbot then asks these general sets of questions.
- The responses of these questions are again processed to find if we can find more information / symptoms and then ask questions of this symptom as well.
- Before asking the question of new symptoms we check if the questions of this symptom are already asked. If the questions of this symptom are not already asked then we ask the questions else do not ask the questions of this symptom.
- Some questions contain conditions using which we move to some condition specific questions. So we check if the question contains conditionals. In case the question contains conditional then we check in the response if the condition is satisfied for this question. If the condition is satisfied then we ask the condition specific questions and move to the next question.
- After all the questions of the symptoms the patient is experiencing are asked, the flow ends.

The high level flow of chatbot:



FLOW EXPLANATION

Data Collection and Data Structure: The data consists of various symptoms and organs along with associated screening questions. The questions of symptoms can give rise to certain conditions, for these conditions we have specific questions. We earlier collected data from web sources but then we modified our dataset using **Symptoms Analysis: A visual guide** book. We studied the symptoms and understood the symptoms which are accompanied by these symptoms and designed our questions according to it. We also took help from a doctor in designing questions and dataset

verification. We have collected data for around 40 symptoms. The total number of decision points are around 250.

Example of Data Structure for Fever Symptom:

```
{
  "symptom": "Fever",
  "questions": [{
    "ques" : "How long have you experienced fever?"
  },
  {
    "ques" : "Is it associated with aches and pain?",
    "condition": {"yes": ["Do you also have dry cough, sore throat, and stuffy or runny nose?",
      "Do you also have vomiting and diarrhea?"]}
  },
  {
    "ques" : "Do you have abdominal pain?",
    "condition": {"yes": ["Is it associated with cough?",
      "Do you generally feel unwell?"]}
  },
  {
    "ques" : "Is it associated with neck stiffness?",
    "condition": {"yes": ["Do you have eye irritation?", "Do you have rashes?",
      "Do you experience seizures?"]}
  },
  {
    "ques" : "Is it associated with headache and tiredness",
    "condition": {"yes": ["Do you have abdominal pain and constipation?",
      "Do you have diarrhea?",
      "Do you rashes on chest and abdomen?"]}
  },
  {
    "ques" : "Do you experience confusion and dis-orientation?",
    "condition": {"yes": ["Do you generally breath fast and shallow?",
      "Do you experience seizures?",
      "Have you ever experienced loss of consciousness?"]}
  },
  {
    "ques" : "Have you experienced muscle and joint pain?",
    "condition": {"yes": ["Is it associated with headaches?",
      "Is it associated with shivering and sweating?",
      "Have you ever experienced loss of appetite?",
      "Is it associated with diarrhea and vomiting?"]}
  }
]
}
```

General Questions if No symptom is matched: The general set of questions are designed using seven attributes of the symptom namely location, quality, quantity,

timing, setting (what might have caused the symptom), factors that aggravate or relieve the symptom and associated symptoms.

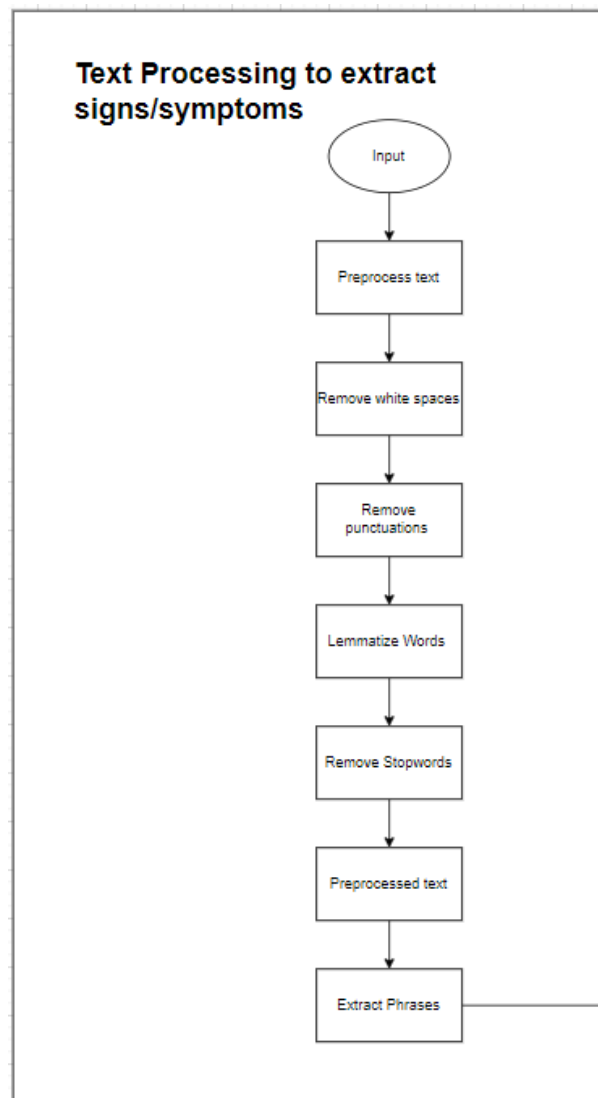
- Please specify where the symptom is occurring?
- Describe what you are feeling?
- How long have you experienced this symptom?
- What is the frequency of the symptom, is it constant or intermittent?
- Have you noticed what might have caused the problem?
- Describe what makes the condition better or worse?
- Have you taken any medicine for this symptom in the past?
- Are there any other symptoms that you are experiencing?

Text Preprocessing: The raw input taken from the user is preprocessed using NLP. The preprocessing step cleans the user input and formats the text so that the text can be used for further processing.

The preprocessing steps are:

- **Removing double and white spaces:** The text might contain double and empty white spaces. These white spaces might create problems in the symptom matching process. So we remove these extra white spaces.
- **Removing punctuations:** The punctuation removal process will help in treating each and every component of text equally.
- **Words Lemmatization:** Lemmatization helps in reducing the words to its meaningful base word. The lemmatization process links words from the input text with similar meanings to one word. WordNetLemmatizer is used for the lemmatizer process.
- **Removing Stopwords:** Stopwords are commonly used words. The main idea is to extract symptoms and organs in which the patient has a problem. So we remove stop words as removing them will enhance the search. The list of all stopwords is taken from stopwords library. The words are then searched in the stopwords list. If the word is then dropped if it is present in the stopwords list.

The preprocessing flow chart of preprocessing is:



Symptom Extraction from preprocessed text : The symptom extraction process takes the preprocessed text and extracts the useful phrases and words from the preprocessed text.

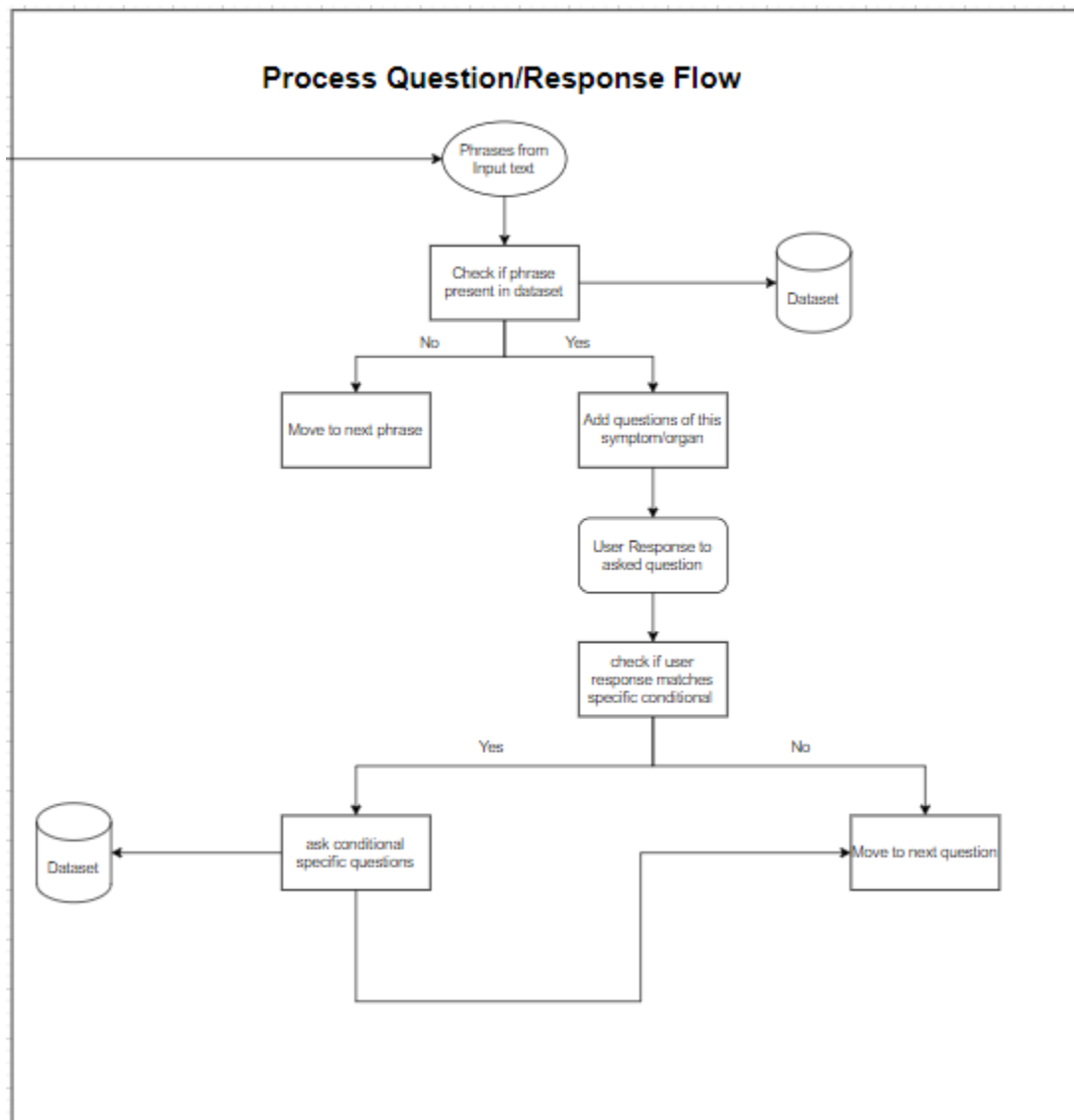
The **spacy** library is used to extract phrases and words. The **en_core_web_sm** pipeline is built using spacy. **en_core_web_sm** is a small english pipeline trained on written web text (blogs, news, comments), that includes vocabulary, syntax and entities. **TextRank** is in order to find the most relevant sentences in text and also to find keywords. TextRank is a graph-based ranking model for text processing that can be used to locate both keywords and the text's most pertinent sentences.

Words and Phrases matching in the dataset: The extracted phrases and words are then searched into the dataset. The searching is implemented using a **fuzzy search** approach so tackle the spelling mistake errors in the user input.

The python library fuzzywuzzy is used to implement fuzzysearch. Python's FuzzyWuzzy library is used for string matching. Finding strings that match a specified pattern is known as fuzzy string matching.

The search mechanism calculates the Levenshtein Distance between the two sequences and then calculates the similarity index. The similarity index is a score out of 100, based on the similarity of the strings. Here, we are using **85** as a similarity index threshold for searching symptoms in the dataset.

The Process question and response flow is as:



2. Interface to Mark pain

In this project we have screened the patient on various symptoms such as rashes, general body pain, burns, etc. Now in order to make our screening more efficient in collecting data from the patient we created an app in react to gather the location of the pain or sensation the patient is having on a 2D figure of the human body. This can be used by the doctor to accurately judge what type of condition could be possible for the given distribution .

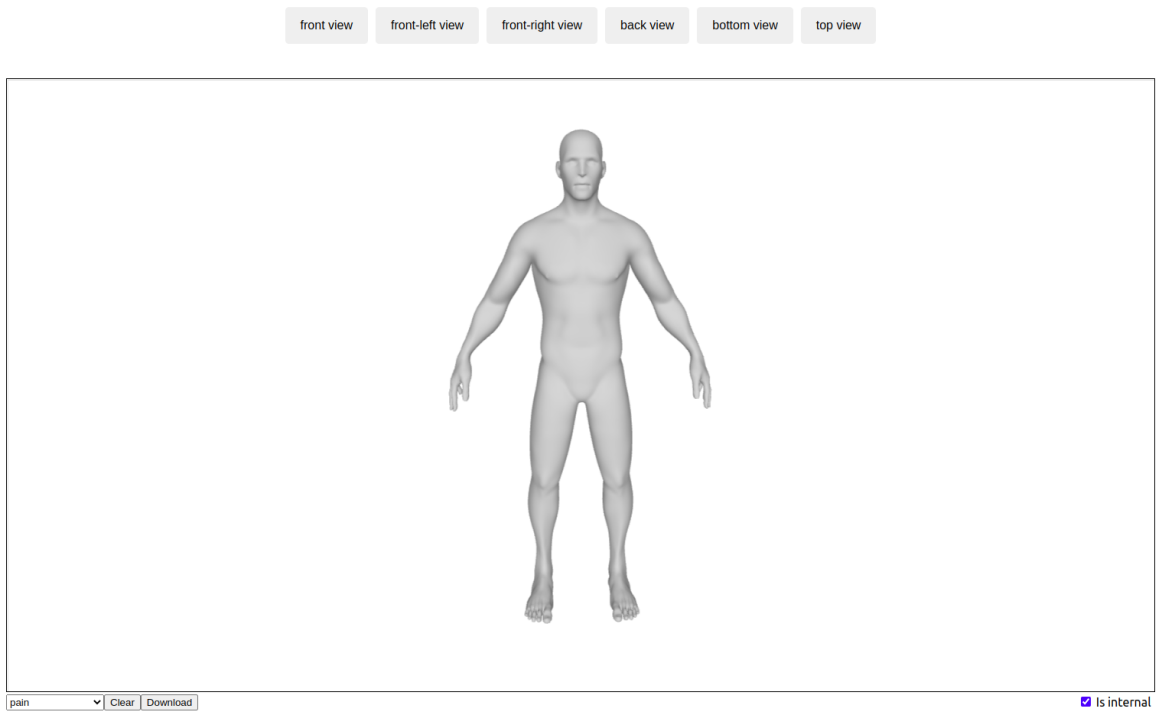
We Have provided the user with the following views

1. Front
2. Front - left
3. Front - right
4. Back
5. Top
6. Bottom

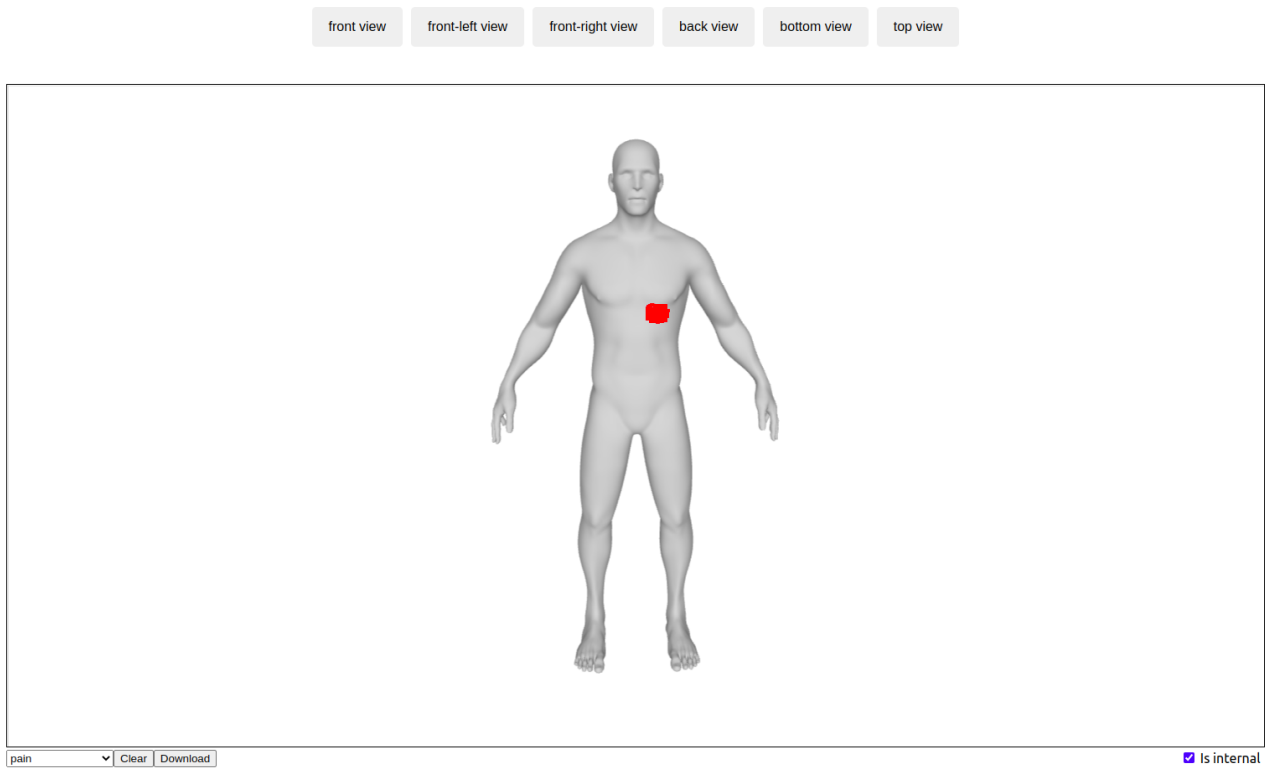
So in this way For example if the user has both Chest pain which is accompanied by pain in the lower back , then He/she can mark it on the dashboard and then the images would be saved for further scrutiny by the doctor .

Also in order to differentiate between the type of sensation for example (burn, pain , itching etc), we provided the small functionality to change the color of the marker . Also suppose the patient wants to differentiate between internal and external pain (in cases of rashes or itch the sensation is at the surface but in cases like chest pain the pain is internal) . For such cases the user can select the option for internal which will be saved along with the image .

Example



Example : patient with an internal pain in left upper side of abdomen



IV. Challenges

- **Data Collection**

Firstly our main objective was to Find examples of conversations that take place between doctors and patients . This was difficult to find as we were mainly finding proper conversations that didn't end in just a single line conversation with the patient describing his/her condition and the doctor suggesting some medicine or remedy .

We found such a database at https://huggingface.co/datasets/medical_dialog ,Which consisted of Single sentence question answer format conversation between the patient and the doctor .

We decided to solve the problem by using medical sources such as web - md for gathering information related to the symptoms which we were including in our screening chatbot

Another issue we faced is converting the gathered data regarding common conditions associated with various symptoms, to a set and sequence of questions which are medically correct . To accomplish that we consulted a doctor who guided us through the framing of the questions and validated our dataset .

V. Outcomes

We have collected data and designed questions that doctors ask if the patient is experiencing those symptoms. We then developed a chatbot using NLP that can be used as pre consultation screening of a patient by questionnaire. The response of the user is recorded and can be shown to the doctor. The chatbot can be used to reduce the time taken by a doctor in screening a patient. It will also utilize the waiting time of the patient. We have also designed an interface where a patient can mark the place along with the type of pain and it can be then shown to the doctor.

VI. Future work

We intend to further develop this project by adding more data points which would cover a wider range of conditions and cater to many more use cases. Also we would like to add the functionality to parse more complex patient input statements .

VII. References

1. Medical symptoms a visual guide, DK publishing
2. Web-md
3. <https://lemongrad.com/conversation-between-doctor-and-patient/>
4. <https://www.nhp.gov.in/disease-a-z>