HealthCare Chatbot

1. Introduction:

The healthcare chatbot project aims to develop an AI-powered chatbot that assists users with health-related queries and provides relevant information about different diseases. The chatbot is built using the Rasa framework for natural language understanding and Flask for creating a web interface.

2. Project Objectives:

- Collect relevant data to provide accurate and helpful information about various health-related topics and diseases.
- Develop conversational interface for users to interact with the chatbot.
- Implement natural language understanding and dialogue management using Rasa.
- Create a user-friendly web interface using Flask for easy accessibility.

3. Technologies Used:

- Rasa: An open-source conversational AI framework for building chatbots and virtual assistants.
- Flask: A lightweight web framework used for creating the web interface of the chatbot.
- HTML/CSS: Used for designing and styling the chatbot user interface.
- JavaScript: Used for handling user interactions and making API requests.

4. System Architecture:

The healthcare chatbot system consists of the following components:

- Rasa NLU: Responsible for natural language understanding and extracting intents and entities from user messages.
- Rasa Core: Handles the dialogue management and generates appropriate responses based on the conversation context.
- Flask Server: Acts as the backend server to receive user messages, interact with Rasa, and send back bot responses.
- User Interface: A web interface where users can interact with the chatbot.

5. Implementation Steps:

- Gathering health-related information on various diseases and compiling it into a CSV file.

- **Setting up Rasa:** Install Rasa and create the necessary training data files.
  In this step, our team defined the intents, entities, and dialogue stories and rules required for the healthcare chatbot. We created training data files in the Rasa format (nlu, stories and rules YML files) providing examples of user messages, corresponding intents, and entities.

- **Training the Chatbot:**
  We used Rasa NLU (Natural Language Understanding) to train the chatbot's language understanding capabilities and configured the NLU pipeline by specifying the language processing components, such as tokenizers, featurizers, and classifiers. The NLU model was trained on the provided training data to enable intent classification and entity recognition.
  The following components were included in the pipeline:
    - ❖ WhitespaceTokenizer: This component tokenizes the user input by splitting the message on whitespace.
    - ❖ RegexFeaturizer: It applies regular expressions to extract specific patterns or entities from the user message.
    - ❖ LexicalSyntacticFeaturizer: This component captures lexical and syntactic features of the user message to improve intent classification.
    - ❖ CountVectorsFeaturizer: It converts the text into numerical vectors using character-based n-grams. The specified parameters include the minimum and maximum n-gram lengths.
    - ❖ DIETClassifier: This component is a neural network-based classifier that combines intent classification and entity recognition. It is trained for a specified number of epochs and can constrain similarities between similar intents.
    - ❖ RegexEntityExtractor: It extracts entities from user messages using predefined lookup tables.
    - ❖ EntitySynonymMapper: This component maps synonymous entities to a common representation.

- **Configuring Rasa Core:**
  We defined the dialogue flow of the chatbot by creating a dialogue management model using Rasa Core and specifying the policies in the Rasa configuration file to determine how the chatbot selects actions and responds to user inputs. The dialogue management model was trained using the training data, which includes dialogue examples, actions, and corresponding bot responses.
  The following policies were included:
    - ❖ MemoizationPolicy: This policy uses the history of user inputs and the corresponding bot responses to predict the next action. It memorizes successful action sequences to improve performance.

    - ❖ TEDPolicy: The TED (Transformer Embedding Dialogue) policy is a neural network-based policy that uses self-attention mechanisms to capture context from previous user inputs. It considers a specified maximum history of user inputs and is trained for a specified number of epochs.
    - ❖ RulePolicy: This policy enables rule-based responses to specific user inputs. It allows for predefined rules to be defined and triggers corresponding actions or responses.

- **Action Server:**
  Our team implemented an action server using Rasa SDK to handle custom actions that require external computations. We defined a custom action in the action server and implemented the necessary logic to retrieve medical information from the CSV file. The action server was connected to the Rasa Core model to enable the execution of the custom action during the dialogue flow.

- **Flask Integration:** Implement the Flask server to handle incoming user messages, interact with the Rasa model, and send bot responses.

- **Designing the User Interface:** Create an HTML template, CSS and JavaScript for the chatbot interface, including message containers and input form.

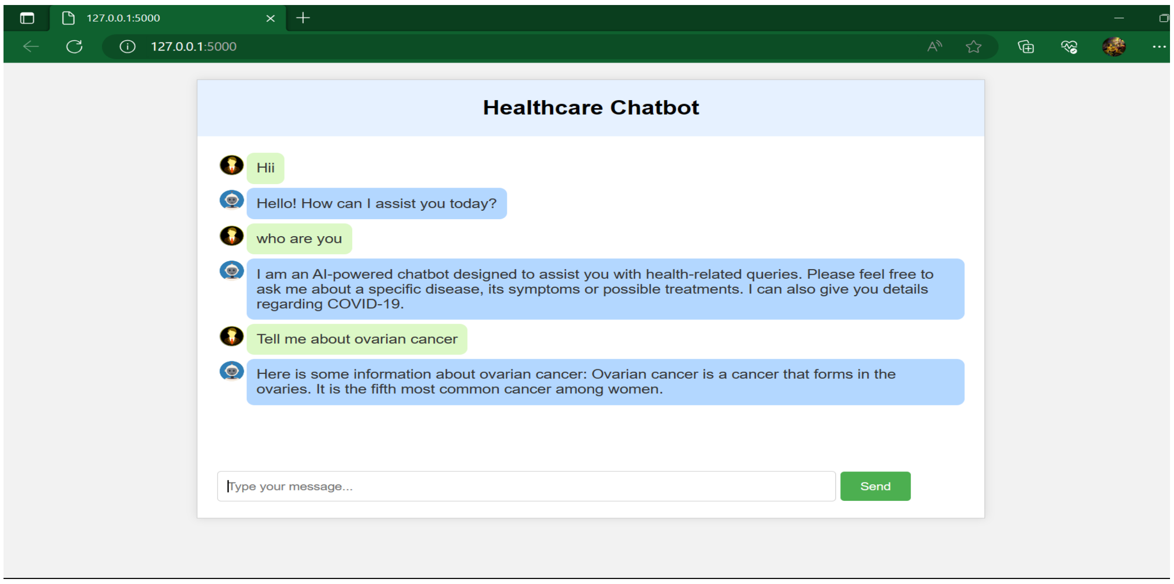
6. Results and Achievements:

- Successfully developed a healthcare chatbot with conversational capabilities.
- Implemented natural language understanding using Rasa NLU to extract user intents and entities.
- Implemented dialogue management using Rasa Core to generate appropriate responses based on the conversation context.
- Created a user-friendly web interface using Flask, allowing users to interact with the chatbot easily.
- Achieved the project objective of assisting users with health-related queries and providing relevant information about diseases.

7. Future Enhancements:

- Implementing user authentication and personalized recommendations based on user profiles and medical history.
- Enhancing the natural language understanding capabilities to handle complex medical queries and nuances.
- Expanding the knowledge base of the chatbot by continuously updating and adding new medical information.
- Deploying the chatbot on a cloud platform for scalability and availability.

8. Output:



Healthcare Chatbot

Hii

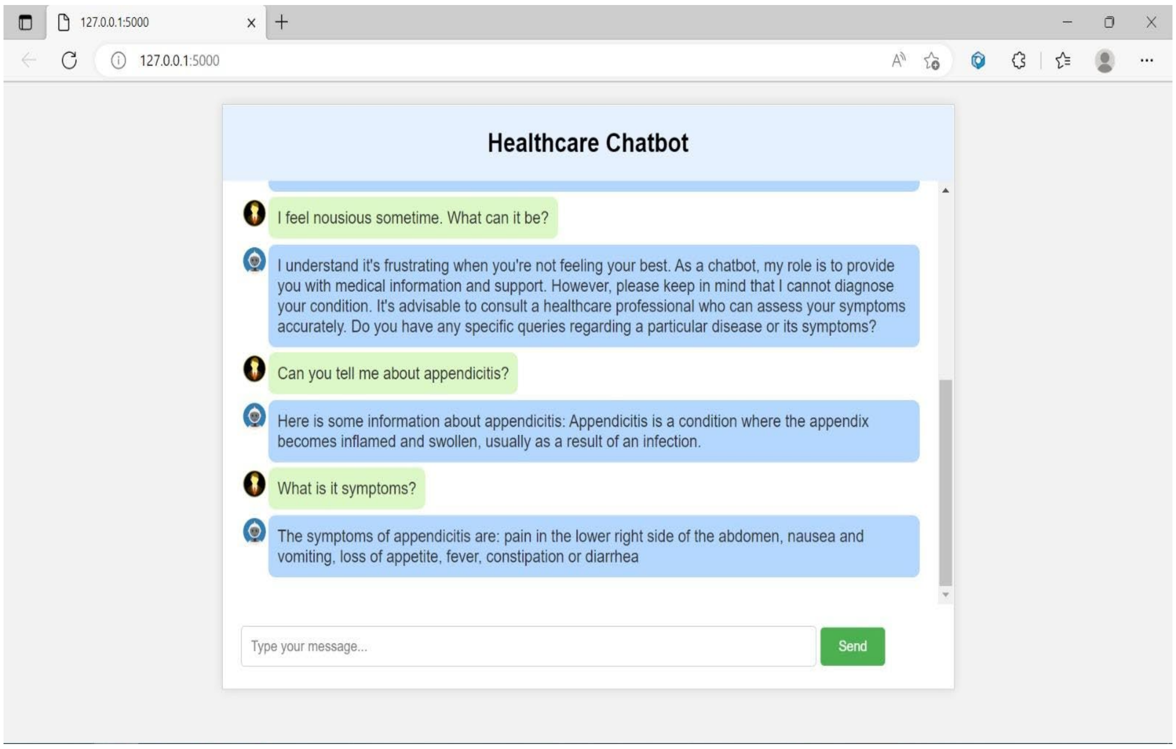
Hello! How can I assist you today?

who are you

I am an AI-powered chatbot designed to assist you with health-related queries. Please feel free to ask me about a specific disease, its symptoms or possible treatments. I can also give you details regarding COVID-19.

Tell me about ovarian cancer

Here is some information about ovarian cancer: Ovarian cancer is a cancer that forms in the ovaries. It is the fifth most common cancer among women.



Healthcare Chatbot

I feel nousious sometime. What can it be?

I understand it's frustrating when you're not feeling your best. As a chatbot, my role is to provide you with medical information and support. However, please keep in mind that I cannot diagnose your condition. It's advisable to consult a healthcare professional who can assess your symptoms accurately. Do you have any specific queries regarding a particular disease or its symptoms?

Can you tell me about appendicitis?

Here is some information about appendicitis: Appendicitis is a condition where the appendix becomes inflamed and swollen, usually as a result of an infection.

What is it symptoms?

The symptoms of appendicitis are: pain in the lower right side of the abdomen, nausea and vomiting, loss of appetite, fever, constipation or diarrhea

9. Conclusion:

Our team successfully developed an AI-powered healthcare chatbot using Rasa and Flask, giving users a reliable resource for medical questions. The chatbot is a helpful tool for finding out medical information because of its conversational capabilities and user-friendly interface. The chatbot can continue to advance and give users in the healthcare industry more individualized and precise support with additional improvements and refinements.

10. References:

- Rasa Documentation: https://rasa.com/docs/
- Flask Documentation: https://flask.palletsprojects.com/

11. Github:

Click Here: **Healthcare Chatbot**