**Ganadipathy Tulsi’s Jain Engineering College**

Kaniyambadi – Vellore – 632102

**IBM – PROJECT**

**Phase 5:** Final Submission

**CREATE A CHATBOT IN PYTHON**

**TEAM ID: 223927(TEAM-2)**

Submitted by,

Team Members,

Manojh R

Monikasri M

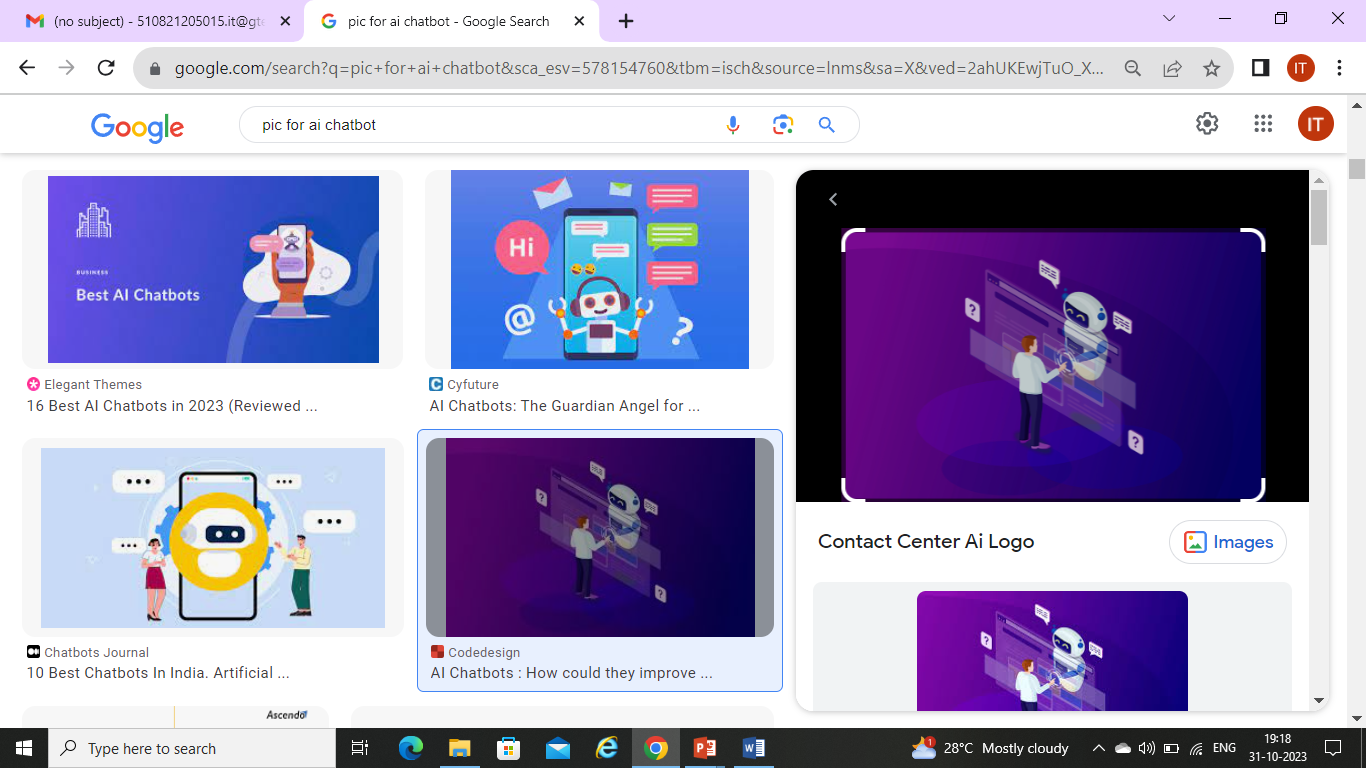
Shabitha S

Vijayadhithya M

**Create a Chatbot in Python**

**INTRODUCTION:**

* The problem is to develop an AI-powered chatbot that can predict the likelihood of an individual developing diabetes by analysing medical data.
* The chatbot aims to provide early risk assessment and personalized preventive measures, enabling individuals to take proactive actions to manage their health effectively.

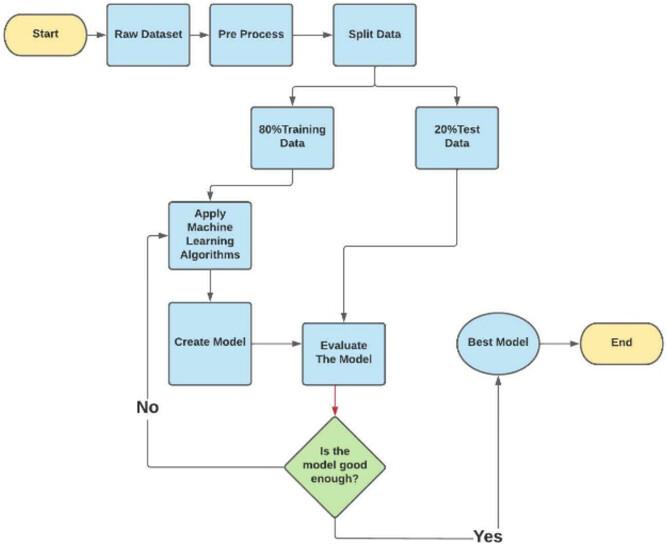


* Chatbots are computer programs that can simulate conversation with humans. They are often used in customer service applications, but they can also be used for other purposes, such as education, entertainment, and companionship.
* A chatbot web app that is powered by the GPT-2 language model. The web app allows users to interact with the chatbot by typing in their questions and requests. The chatbot will then respond in a natural and informative way.

**Solution Overview:**

The proposed solution involves creating an AI-powered chatbot that integrates machine learning algorithms for diabetes prediction. The chatbot will be designed to provide a user-friendly interface, analyze user input, offer accurate predictions, and recommend personalized preventive measures.

**FLOWCHART:**

****

**Flowchart Description:**

**1. User Interaction**: Users interact with the chatbot through a website or app interface.

**2.** **Input Processing (Chatbot)**: The chatbot processes user input, which may include medical data and queries related to diabetes risk assessment.

**3.** **Diabetes Prediction (Chatbot)**: Machine learning algorithms are used to analyze the medical data provided by the user. The chatbot predicts the likelihood of the user developing diabetes based on the data.

**4.** **Response Generation (Chatbot)**: The chatbot generates responses that include the diabetes risk assessment and personalized preventive measures.

**5.** **Medical Data (Data Source)**: The chatbot may refer to medical data from a trusted source to enhance the accuracy of predictions.

**6.** **Website/App Integration**: The chatbot is integrated into a website or app, ensuring a seamless user experience.

**7.** **User Testing**: Continuous testing is conducted to evaluate the chatbot's performance and user satisfaction.

**8.** **Feedback**: User feedback is collected to identify areas for improvement.

**Key Features and Functionalities:**

* User-friendly interface for easy interaction.
* Natural Language Processing (NLP) techniques for understanding user queries
* Machine learning algorithms for diabetes prediction.
* Personalized responses with risk assessment and preventive measures.

**DESIGN THINKING:**

* **Functionality:** The chatbot will answer questions, provide guidance, and direct users to relevant resources.
* **User interface:** We'll integrate the chatbot into a website or app with a user-friendly interface.
* **Natural language processing (NLP):** Implement NLP techniques for conversational interactions.
* **Responses:** Plan accurate responses, suggestions, and assistance.
* **Integration:** Decide how the chatbot will be integrated.
* **Testing and improvement:** Continuously refine performance based on user interactions.

**PHASES OF DEVELOPMENT:**

There are seven steps to Develop a Chatbot for your business.

* Define the problem. First of all, it is important to decide why you would need a chatbot.
* Create the conversation flow.
* Selecting the chatbot platform.
* Integrating the chatbot.
* Testing the chatbot.
* Analysing the conversations.
* Improve the chatbot.

**Transformation Steps:**

* Data collection and preprocessing.
* Machine learning model development.
* Chatbot development.
* User testing.
* Integration and deployment.
* Continuous improvement.
* Security and privacy.

**Data Collection and Preprocessing:**

* + Gather relevant medical data including patient records, lab results, lifestyle information, and genetic factors.
  + Anonymize and preprocess the data to remove any identifying information and ensure it's suitable for machine learning.

**Machine Learning Model Development:**

* + Choose appropriate machine learning algorithms for prediction, such as logistic regression, decision trees, or deep learning.
  + Train the model using historical data to predict the likelihood of diabetes development.
  + Validate and fine-tune the model to ensure accuracy.

**Chatbot Development:**

* + Develop the chatbot interface and integrate it into the designated platform (website, app).
  + Implement NLP techniques for natural language understanding and generation.
  + Connect the chatbot to the trained machine learning model for predictive capabilities.

**User Testing:**

* + Launch a beta version of the chatbot for user testing.
  + Collect user feedback on the chatbot's functionality, usability, and accuracy in predicting diabetes risk.
  + Address any issues or shortcomings identified during testing.

**Integration and Deployment:**

* + Integrate the chatbot into the chosen platform, ensuring a seamless user experience.
  + Ensure that the system complies with relevant regulations regarding medical data and AI in healthcare.

**Continuous Improvement:**

* + Monitor the chatbot's performance and user interactions.
  + Regularly update the machine learning model with new data to improve prediction accuracy.
  + Implement user feedback to enhance the chatbot's responses and capabilities.

**Security and Privacy:**

* + Implement robust security measures to protect user data and ensure confidentiality.
  + Comply with data privacy regulations (e.g., GDPR, HIPAA) to safeguard medical information.

**Features:**

The chatbot web app has the following features:

* It can answer a wide range of questions, including factual, open-ended, and challenging questions.
* It can generate different creative text formats of text content, like poems, code, scripts, musical pieces, email, letters, etc.
* It can translate languages.
* It can be used in different ways, such as customer service, education, entertainment, and companionship.

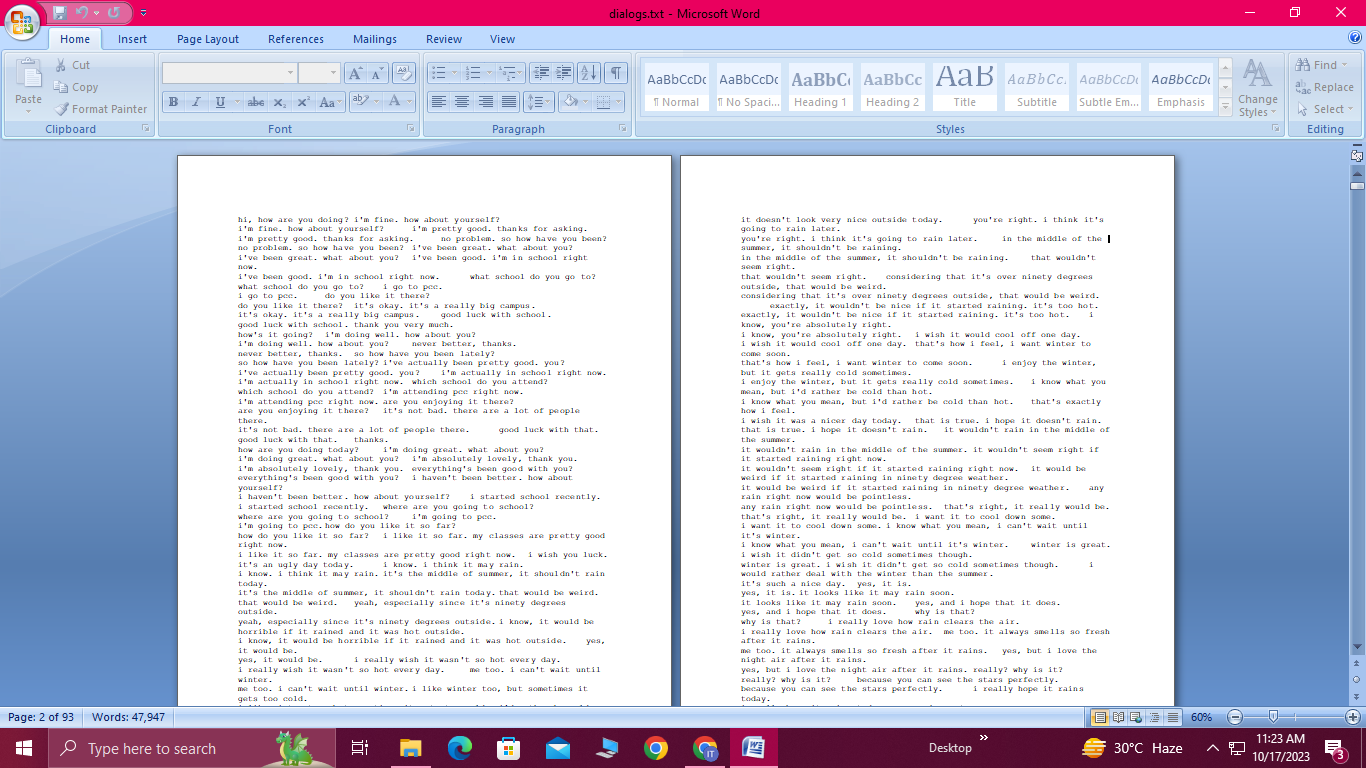
**STEPS TO RUN THE CHATBOT :**

* Create a virtual environment using terminal using #python -m venv myenv,
* Activate the virtual environment using #myenv/scipts/Activate,
* Install the required libraries after activating the virtual environmrnt,
* In the installed library "FLASK" we need to insert our HTML file as a template in the templates model for web application,
* After all the 4 steps completed the chatbot is ready to run for deployment,
* To run and deploy the chatbot web application #python chatbot.py .on the terminal,
* Now the chatbot will deployed on the webpage , and user can ask the quries to the chatbot and get the response .

**DATASET LINK:**

[www.kaggle.com/datasets/grafstor/simple-dialogs-for-chatbot](http://www.kaggle.com/datasets/grafstor/simple-dialogs-for-chatbot)

**GIVEN DATASET:**

****

**Import Libraries:**

Start by importing the required libraries.

**Program:**

import re,

import pandas as pd,

import spacy,

from flask import Flask, render\_template, request,

from transformers import GPT2LMHeadModel, GPT2Tokenizer

**DESCRIPTION OF THE LIBRARIES AND MODULES:**

**1. `re`:**

This module provides support for regular expressions. In the code, it is used to perform text preprocessing by removing non-alphabetic characters and reducing multiple consecutive spaces to a single space in text data.

**2. `pandas` (imported as `pd`):**

Pandas is a powerful data manipulation and analysis library for Python. In this code, it is used to read and manipulate the dataset stored in a CSV file ("dialogs.txt"). The dataset is loaded into a Pandas DataFrame for further processing.

**3. `spacy`:**

SpaCy is a natural language processing library. It is used to load the English language model ("en\_core\_web\_sm") for various text processing tasks.

**4. `Flask`:**

Flask is a lightweight and web microframework for Python. It is used to create a web application for the chatbot. The Flask application is configured, and routes are defined for serving web pages and handling user input.

**5. `transformers`:**

This library is part of the Hugging Face Transformers library, which provides easy access to pre-trained NLP models. In this code, it is used to load the GPT-2 model and tokenizer for generating responses to user input.

- `GPT2LMHeadModel` is a class for the GPT-2 language model.

- `GPT2Tokenizer` is used to tokenize text input for the GPT-2 model.

**6. `dataset` (Pandas DataFrame):**

This variable holds the dataset loaded from the "dialogs.txt" file. The dataset contains pairs of questions and answers, which are used for both user input matching and generating responses.

**7. `clean\_text` function:**

This is a user-defined function that performs text preprocessing by removing non-alphabetic characters, reducing multiple spaces, and converting text to lowercase.

**8. `remove\_repeating\_sentences` function:**

This user-defined function removes repeating sentences from the dataset by keeping only unique questions and their corresponding answers.

**9. Flask routes:**

**- `@app.route('/')`:** This route is for the homepage and is associated with the `index` function, which renders the main chat interface.

- **`@app.route('/chat')`:** This route is for handling user input and generating bot responses. It is associated with the `chat` function.

**- `@app.route('/dataset')`:** This route displays the dataset in a separate web page and is associated with the `show\_dataset` function.

**Importance of loading and preprocessing dataset:**

* Loading and preprocessing the dataset is an important step in building any machine learning model.
* Because, it can improve data quality and the resulting data quality can have directimpact on learning classification models.
* By loading and preprocessing the dataset, we can ensure that the machine learning algorithm is able to learn from the data effectively and accurately.

**Load the Dataset:**

Load your dataset into a Pandas DataFrame. You can typically find diabetes prediction datasets in CSV format, but you can adapt this code to other formats as needed.

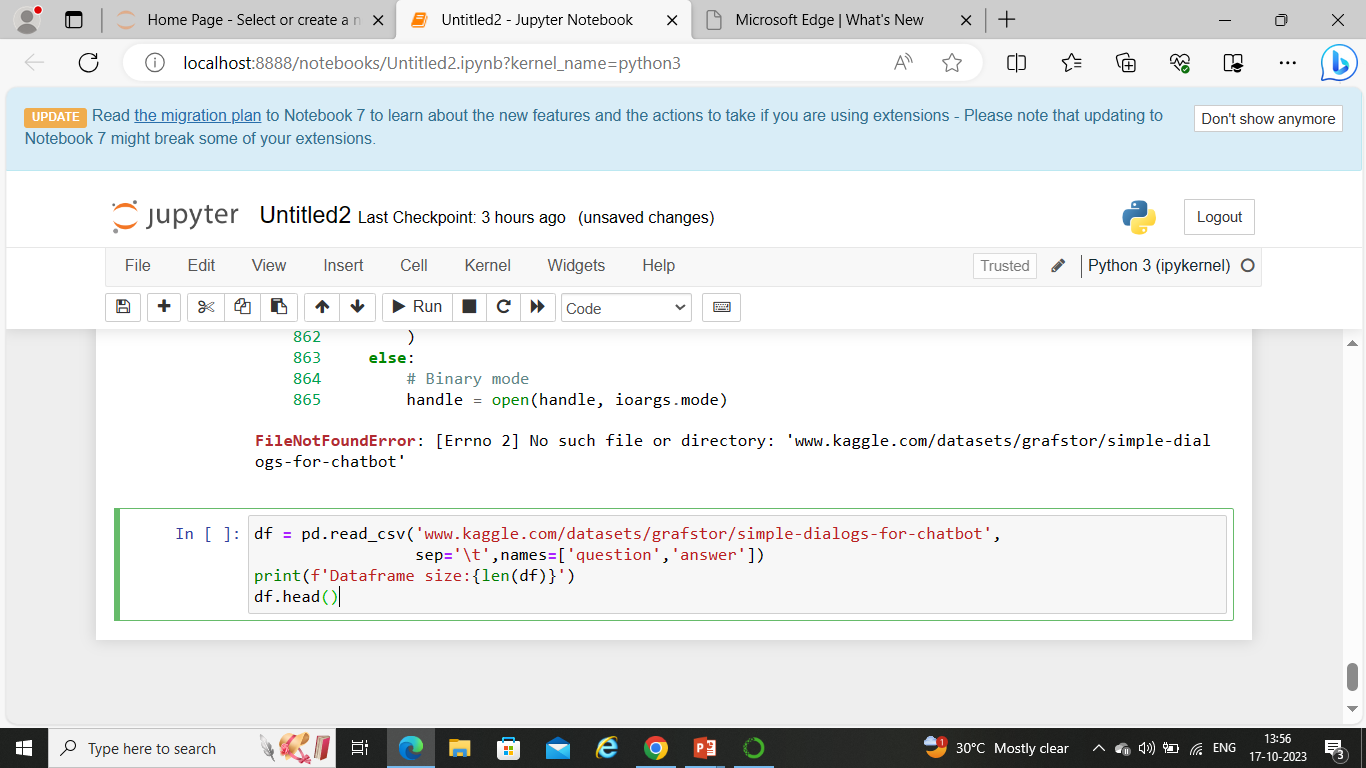
**Program:**

df = pd.read\_csv(‘/kaggle/input/simple-dialogs-for-chatbot/dialogs.txt’)

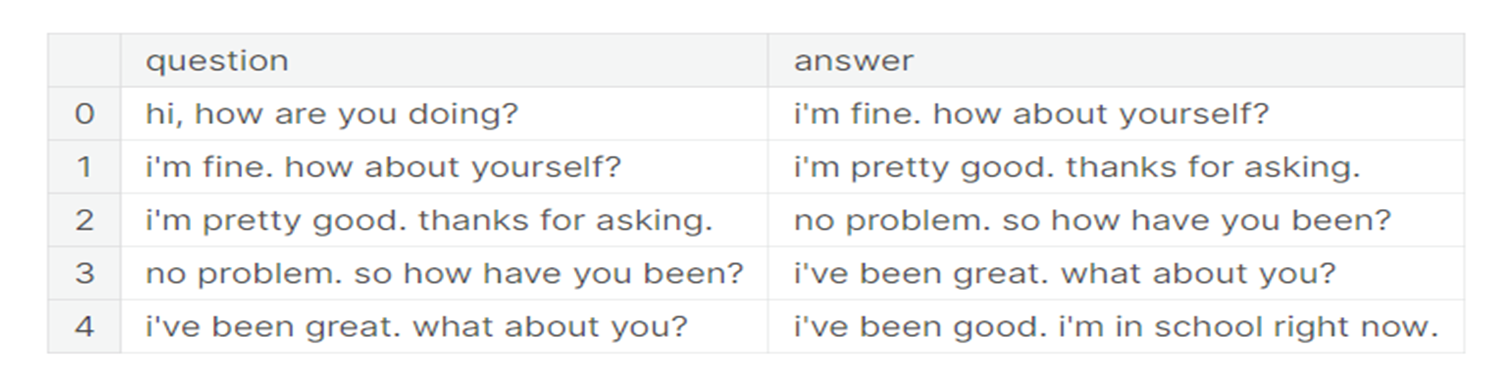
**Explore the dataset:**

Before preprocessing it is essential to get an overview of the dataset.

**Program:**



**Output:**

****

**DATA PREPROCESSING:**

**Steps:**

The preprocessing steps described in the code are as follows:

* Load the dataset.txt file. This file contains the dialogs in plain text format, with one question-answer pair per line.
* Clean the text. This involves removing any punctuation, symbols, or other non-alphanumeric characters. It also involves converting the text to lowercase.
* Create a tokenizer. A tokenizer is a tool that splits text into individual tokens, or words. The code creates a tokenizer with a vocabulary of 2500 words. This means that the tokenizer will only keep the 2500 most common words in the dataset.
* Encode the questions and answers. The tokenizer is used to encode the questions and answers into sequences of integers. Each integer in a sequence represents a word in the vocabulary.
* Pad the sequences to the same length. This is done to ensure that all of the sequences have the same length, which is required by some chatbot training algorithms. The code pads the sequences with zeros to the maximum length of 30 words.
* Save the preprocessed dataset. The preprocessed dataset is saved to a file called preprocessed\_dataset.txt. This file can then be used to train the chatbot.

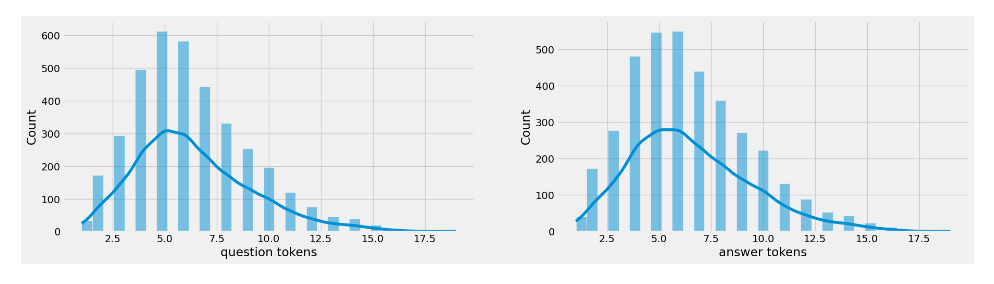
**Data Visualization:**

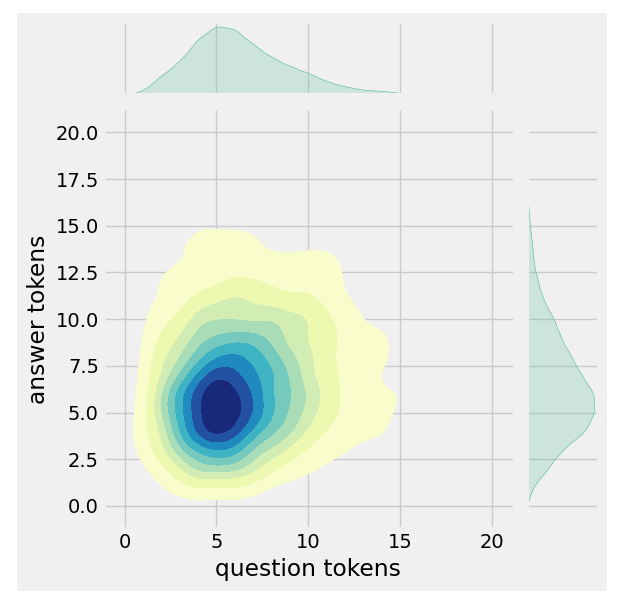
Data visualization allows business users to gain insight into their vast amounts of data. It benefits them to recognize new patterns and errors in the data. Making sense of these patterns helps the users pay attention to areas that indicate red flags or progress. This process, in turn, drives the business ahead.

**Program:**



**Output:**

****

****

**Text Cleaning:**

**1.** A function called `clean\_text` is defined to preprocess text. It performs the following tasks:

**-** Converts text to lowercase.

**-** Replaces hyphens with spaces.

**-** Adds spaces around various punctuation marks, digits, and special characters (., 1-9, 0, ,, ?, !, $, &, /, :, ;,\*, ', ").

**-** Replaces tab characters with spaces.

**2.** The DataFrame 'df' has columns 'answer tokens' and 'question tokens' dropped, effectively removing these columns from the DataFrame.

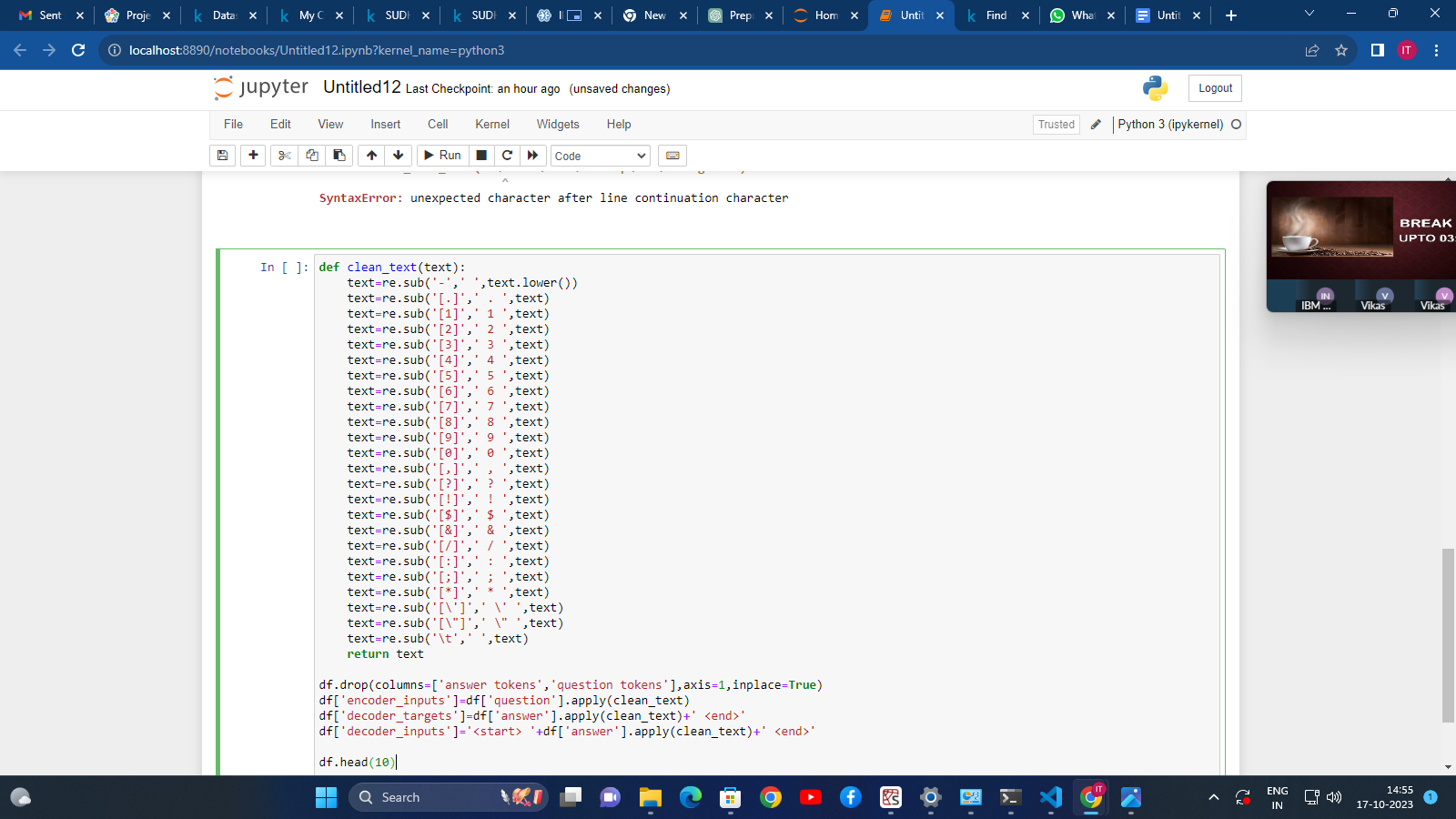
**3.** The 'clean\_text' function is applied to the 'question' column, and the result is stored in a new 'encoder\_inputs' column in the DataFrame.

**4.** The 'clean\_text' function is applied to the 'answer' column, and '<end>' is appended to the end of each processed answer. The result is stored in a new 'decoder\_targets' column in the DataFrame.

**5.** The 'clean\_text' function is applied to the 'answer' column, '<start>' is added at the beginning, and '<end>' is appended at the end. The result is stored in a new 'decoder\_inputs' column in the DataFrame.

**6.** Finally, the first 10 rows of the DataFrame are displayed using `df.head(10)`.

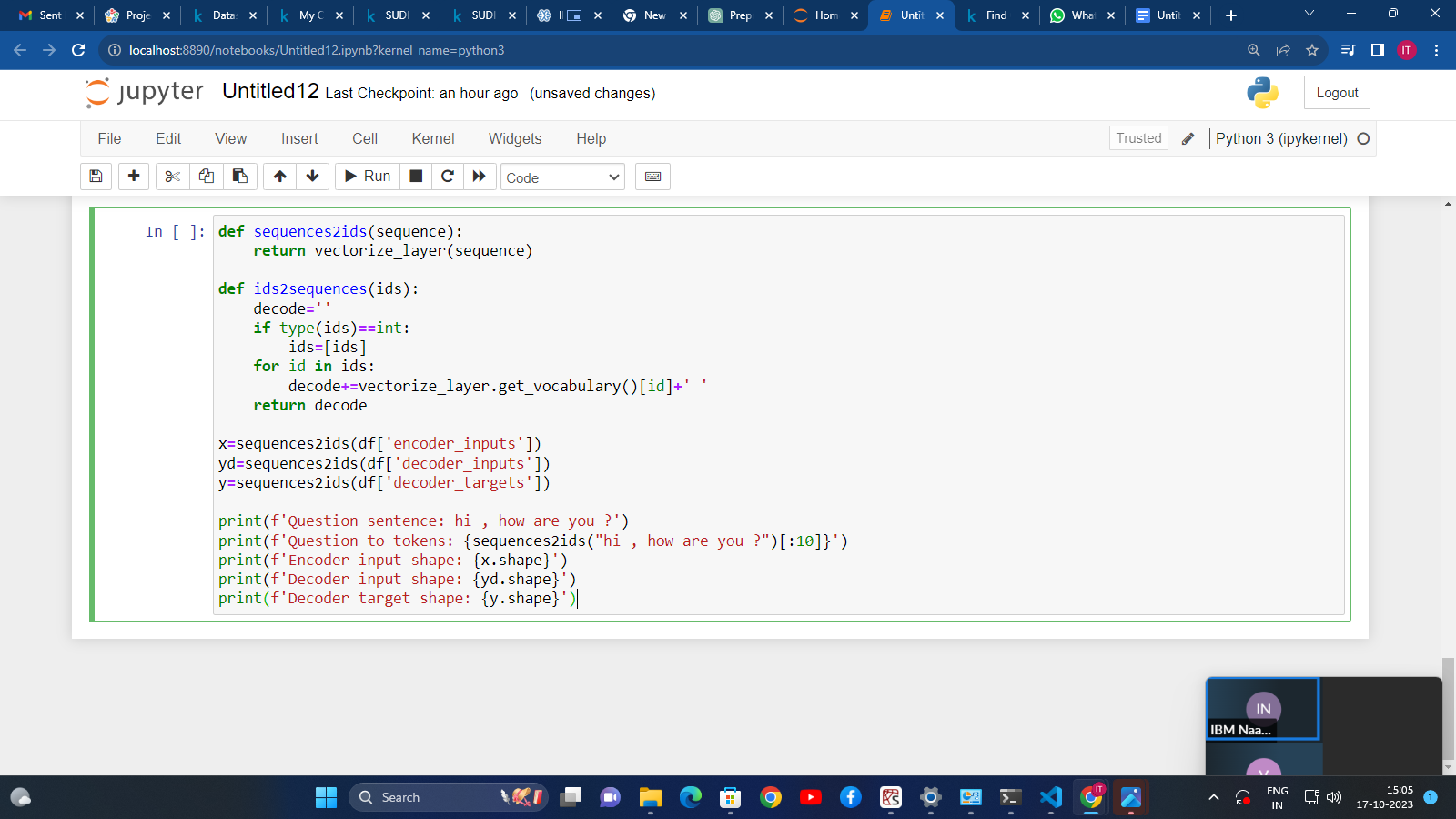
**Program:**

****

**Output**

**Tokenization:**

It calculates the number of tokens (words or symbols) in three specific columns: 'encoder\_inputs', 'decoder\_inputs', and 'decoder\_targets'.

**Program:**

**Output:**

****

**Benefits of Preprocessing:**

There are several benefits to preprocessing a dialog dataset for AI chatbot application:

* Improves the accuracy of the chatbot. By cleaning the text and removing any noise, the preprocessing steps help to ensure that the chatbot is trained on high-quality data. This leads to a more accurate and informative chatbot.
* Reduces the training time. By encoding the text into a format that the chatbot can understand, the preprocessing steps help to speed up the training process.
* Makes the chatbot more robust. By padding the sequences to the same length, the preprocessing steps help to make the chatbot more robust to variations in the length of the input sequences. This makes the chatbot less likely to generate errors when responding to user queries.

**How to Use?**

To use the chatbot web app, simply type in your question or request in the text box and press the "Ask" button. The chatbot will then respond in a natural and informative way.

**Demo:**

The following is a demo of the chatbot web app:

**User:** What is the capital of France?

**Chatbot:** The capital of France is Paris.

**User:** Translate this sentence into Spanish: "I love you."

**Chatbot:** The Spanish translation of the sentence "I love you" is "Te amo."

**User:** Thank you!

**Chatbot:** You're welcome!

**HOW TO CHATBOT INTERACTS WITH THE USERS?**

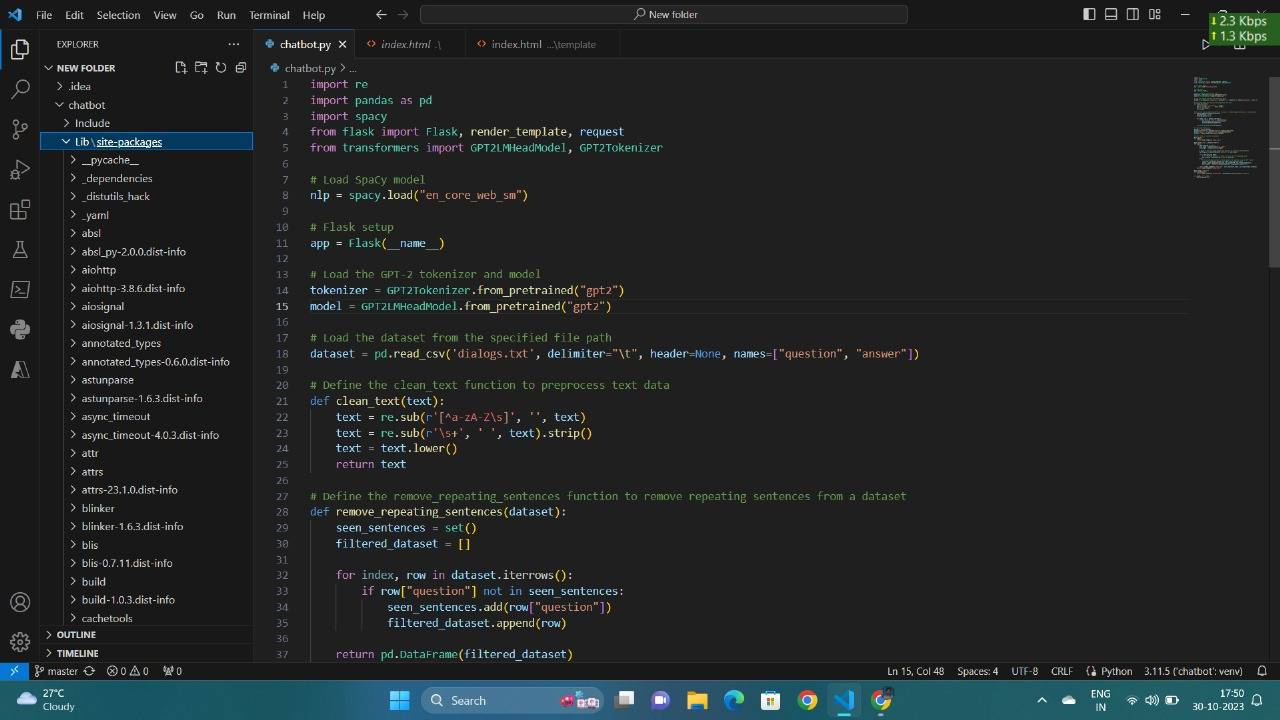
A chatbot can be defined as a developed program capable of having a discussion/conversation with a human.

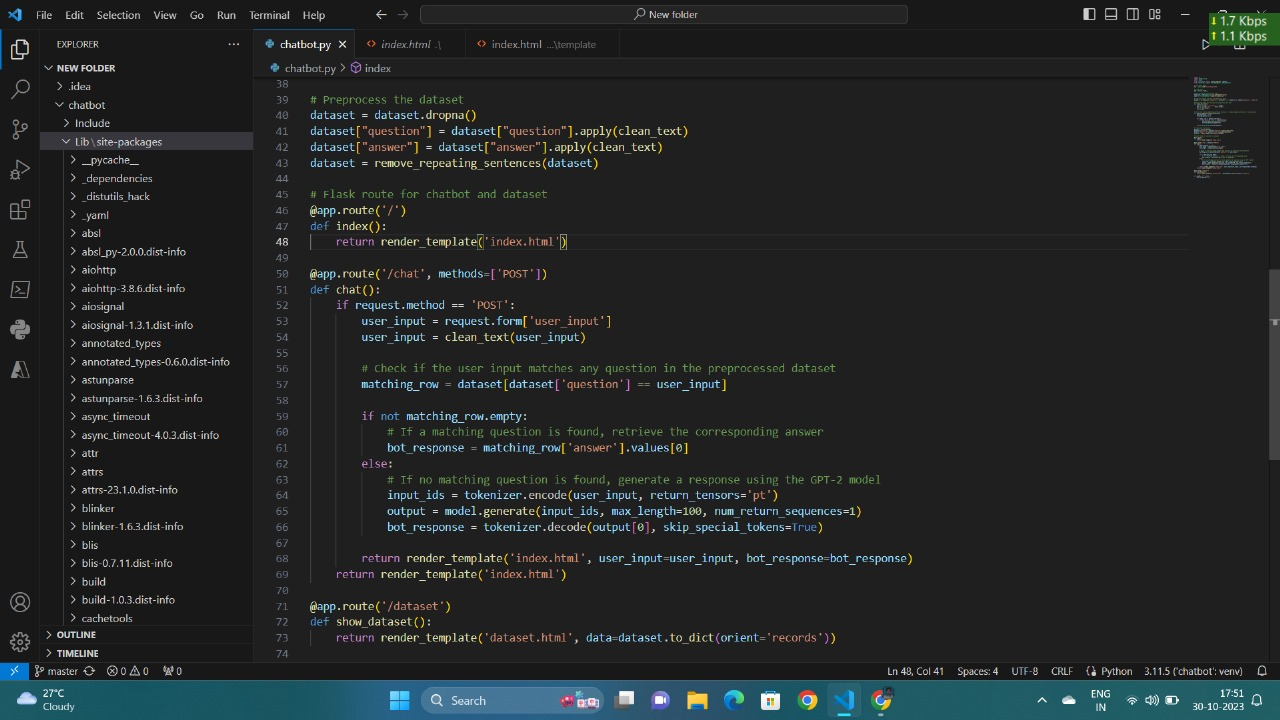
**For example,** ask the bot a question or make a statement, and the bot would answer or perform an action as necessary.

A chatbot communicates similarly to instant messaging.

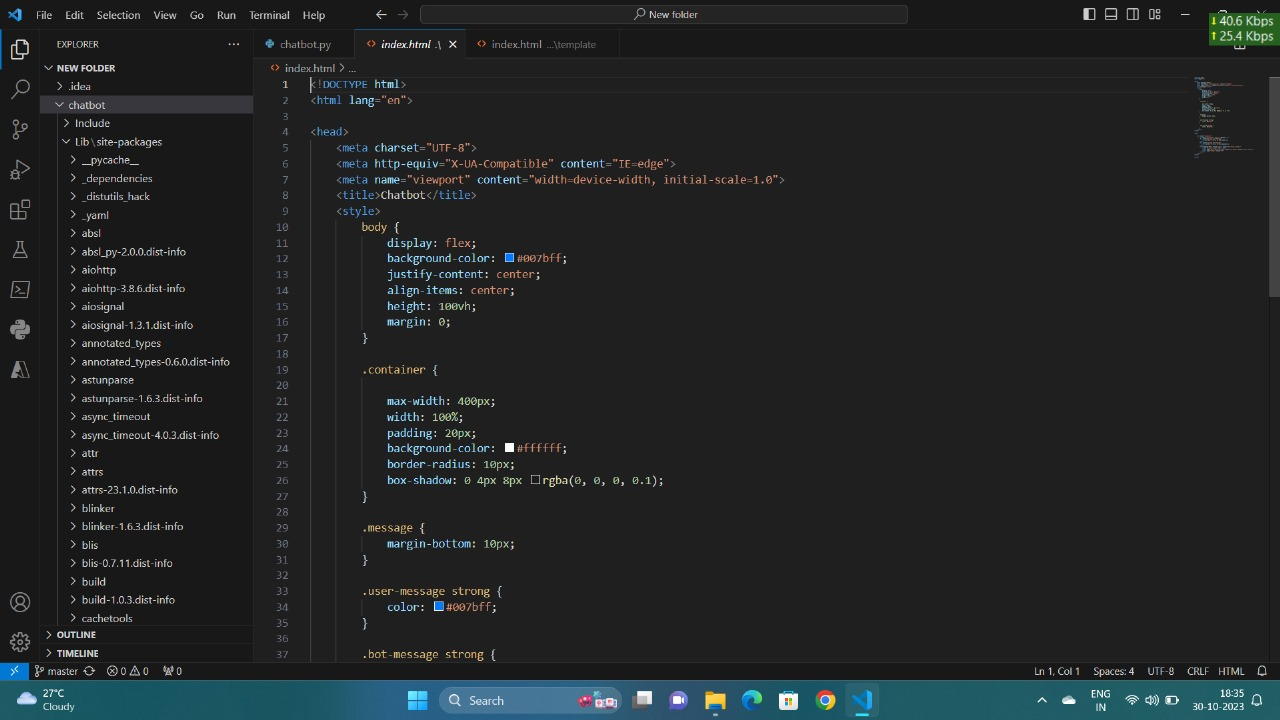
**CODE TO CREATE A AI-CHATBOT:**

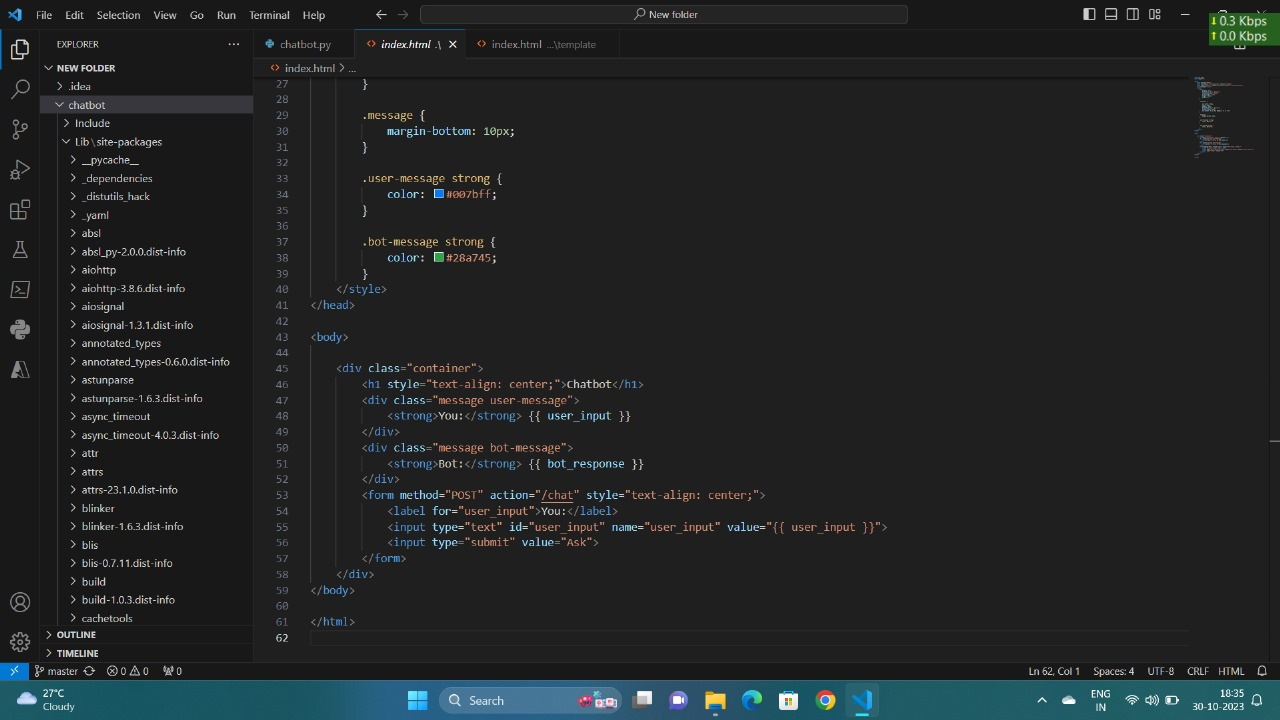
**chatpot.py:**



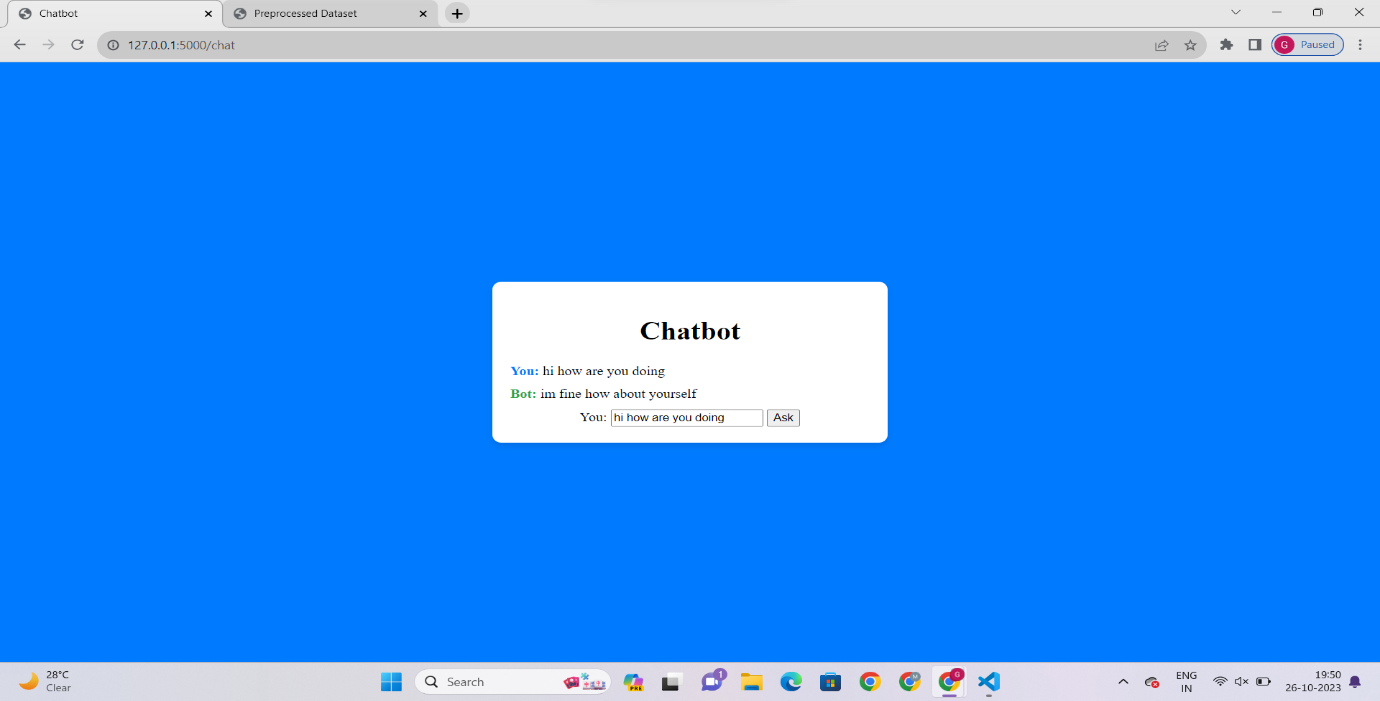


**index.html:**





**Output:**

****

**FUTURE SCOPE:**

The chatbot industry is projected to reach a market size of $3.62 billion by 2030, with an annual growth rate of 23.9%.

This speaks to the increasing popularity of chatbots and their potential as powerful customer service tools.

**Customer service:**

Chatbots can provide instant assistance to customers, which can help reduce wait times and improve customer satisfaction.

In the future, chatbots may become even more sophisticated and be able to handle more complex customer service interactions

.

**ADVANTAGES:**

The chatbot web app offers the following benefits:

* It can provide users with 24/7 customer support.
* It can help users to learn new things and to be more productive.
* It can provide users with companionship and entertainment.
* Instant response and minimized wait time
* Consistent responses
* Hyper-personalized experience
* Multilingual support
* Streamlined solutions across multiple industries

**DISADVANTAGES:**

* **Lack of empathy:** AI chatbots may not be able to understand a customer's emotions, tone, or intent. This can lead to the chatbot appearing emotionally insensitive.
* **S queries:** AI chatbots may have difficulty with complex queries.
* **Raising concerns about data privacy and security:** AI chatbots may raise concerns about data privacy and security.
* **Not being able to answer all queries:** Chatbots can't answer all queries.
* **Being poor in processing:** Certain chatbots are poor in processing and take time to filter results.

**CONLUSION:**

The chatbot web app is a powerful and versatile tool that can be used for a variety of purposes.

It is easy to use and offers a number of benefits, including 24/7 customer support, educational resources, and companionship.