**CREATE A CHATBOT USING PYTHON**

**STEP 1: SET UP A DEVELOPMENT ENVIRONMENT**

Setting up a development environment for creating a chatbot in Python involves several steps, including installing the necessary tools and libraries. Here's a step-by-step guide to help you set up your environment:

**1. Install Required Libraries:**

You'll need various Python libraries for building a chatbot. The most common library for chatbot development is nltk (Natural Language Toolkit) and pytorch. You can install these packages using pip.

Next, we need to install the following packages:

* pandas
* numpy
* string
* sklearn

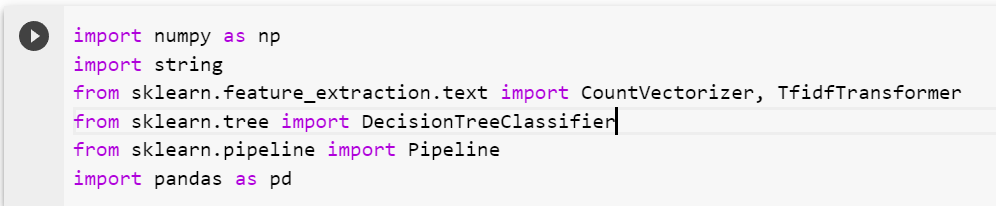
You can install these packages by running the following commands in your terminal or command prompt:

pip install numpy

pip install string

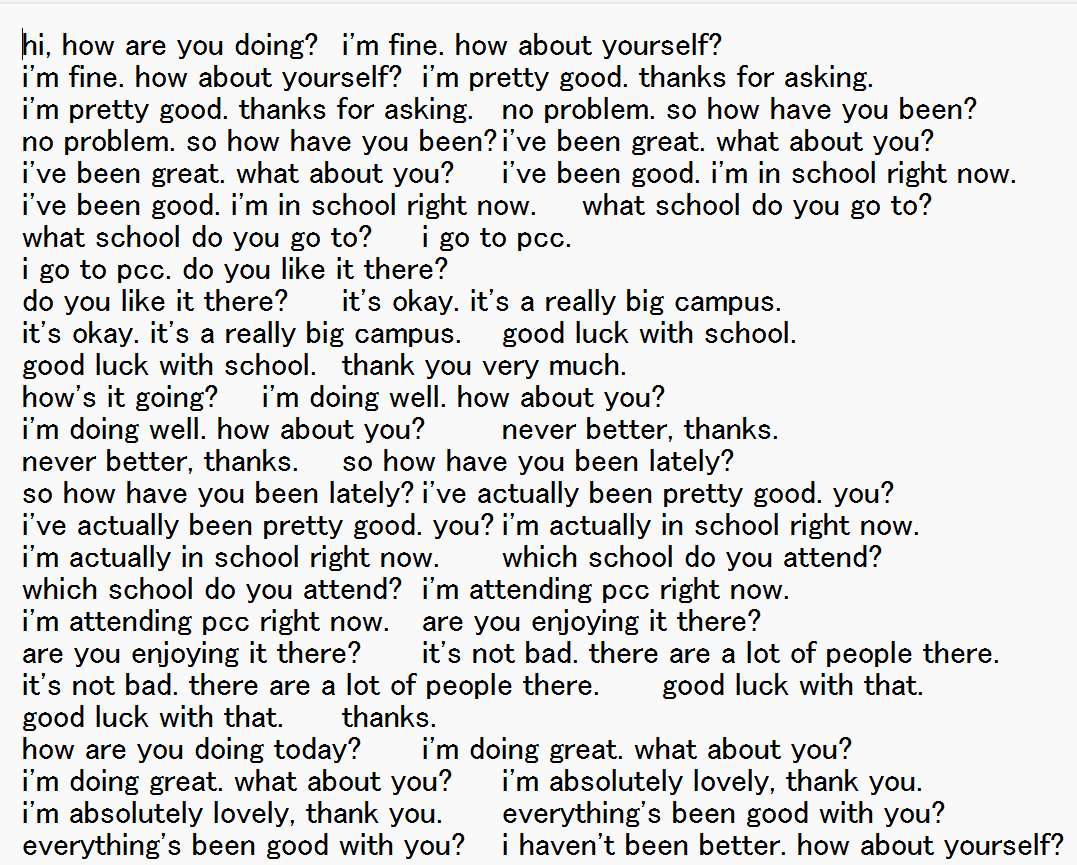
pip install pandas

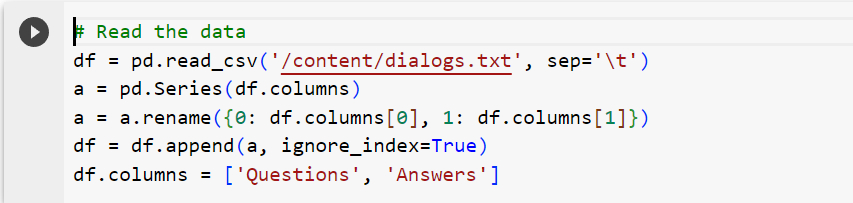
pip install sklearn



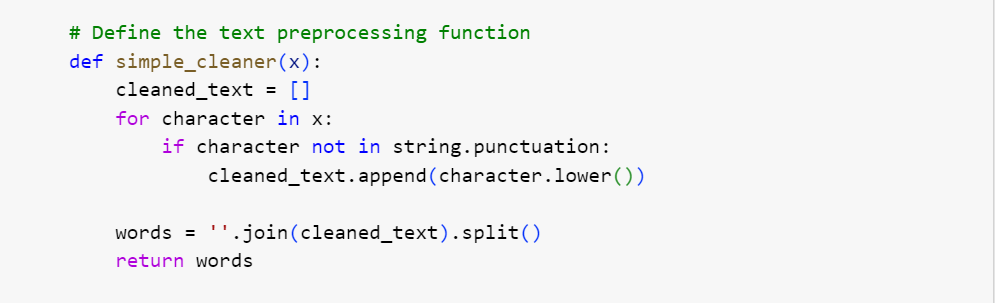
**STEP 2: COLLECT AND PREPROCESS DATA:**

The dataset is from kaggle.com which is known for providing a variety of dataset for machine learning and artificial intelligence.

****

****

**Lowercases all words:** This is done to ensure that all words are treated equally, regardless of case. The words are transformed into lowercase in the above simple\_cleaner().



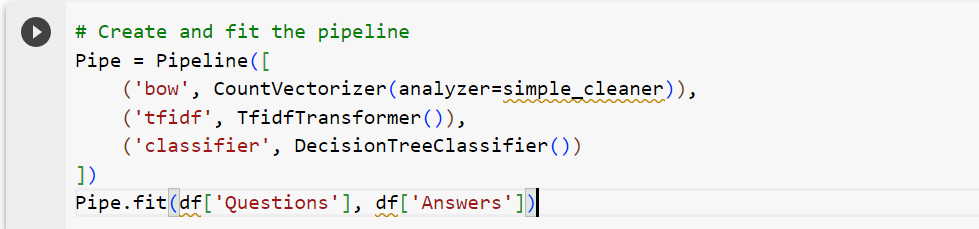
**Removes stopwords and punctuation:** Stopwords are common words that do not add much meaning to text, such as "the", "is", and "of". Punctuation can also be noisy and irrelevant to the meaning of the text. Therefore, both stopwords and punctuation are removed using the set(stopwords.words('english')) and string.punctuation variables. Removing punctuation is done in the above code.

# **STEP 3: TRAIN A MACHINE LEARNING MODEL**

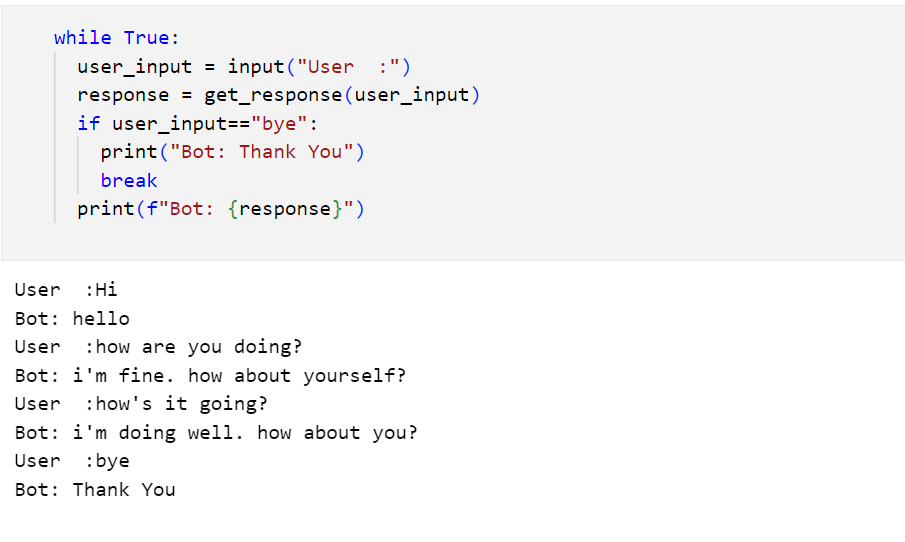
In this machine learning model, we use sklearn’s CountVectorizer , TfidfTransformer and DecisionTreeClassifier.

The machine learning model is build using pipeline that combine multiple machine learning algorithms into a single pipeline.

* **CountVectorizer** object to convert the text data into a bag-of-words (BOW) representation.
* **TfidfTransformer** object to transform the BOW representation into a TF-IDF representation.
* **DecisionTreeClassifier** object to classify the TF-IDF representation into different categories.



**STEP 4: TESTING THE CHATBOT**

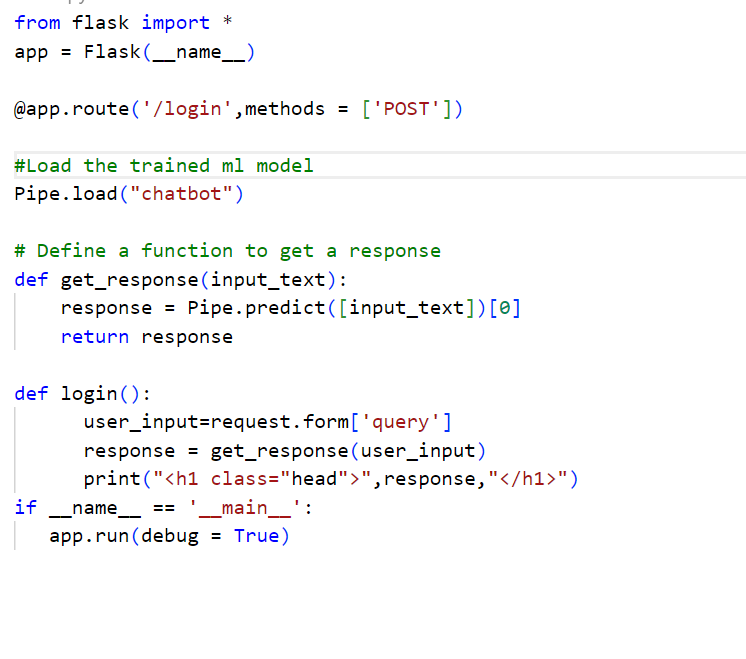


The chatbot machine learning model is designed in such a way that it can handle any kind of query that it is been asked from the given dataset and verified successfully.

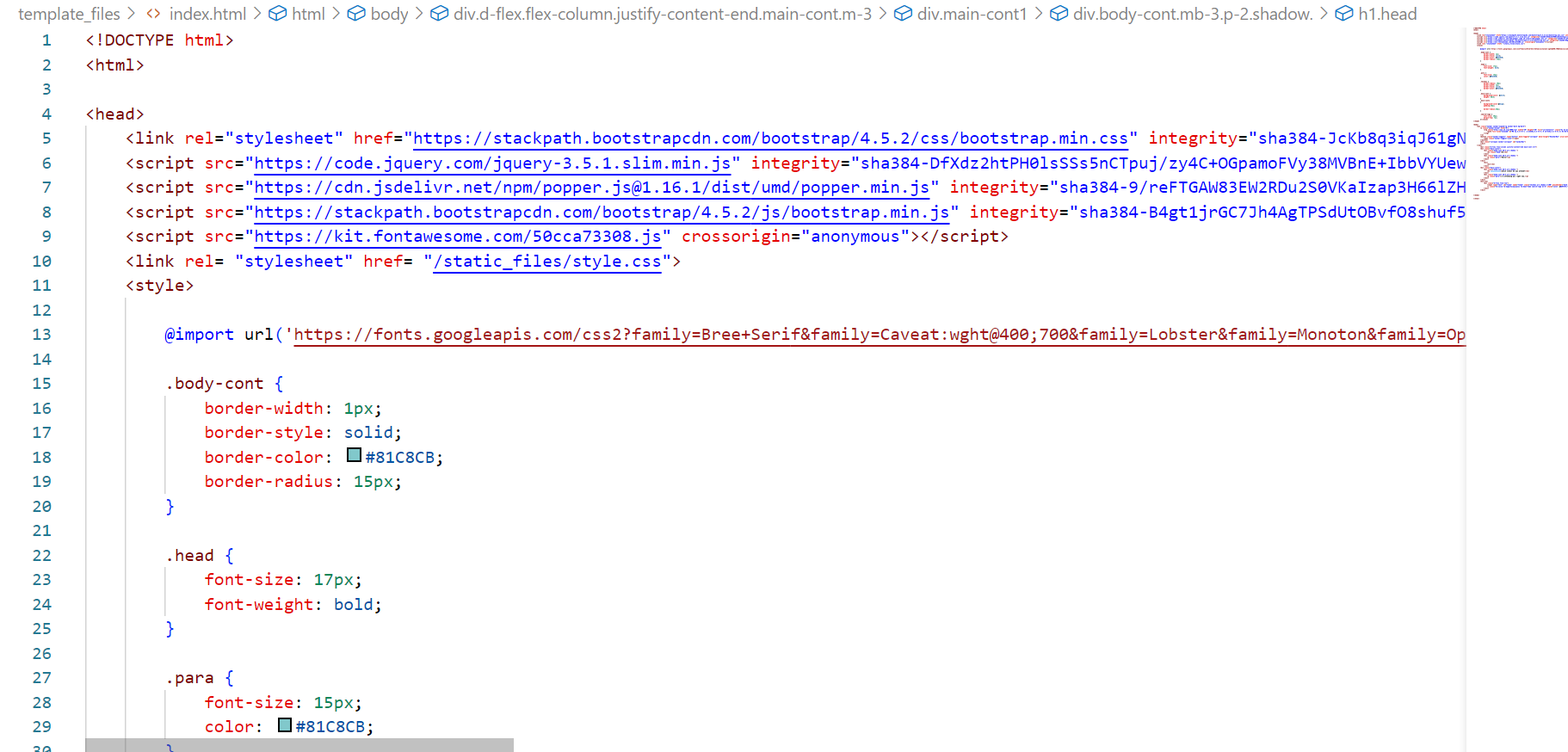
The model is works good and efficient in generating a response based on the input query.

**STEP 5: CREATE A WEB INTERFACE USING FLASK**

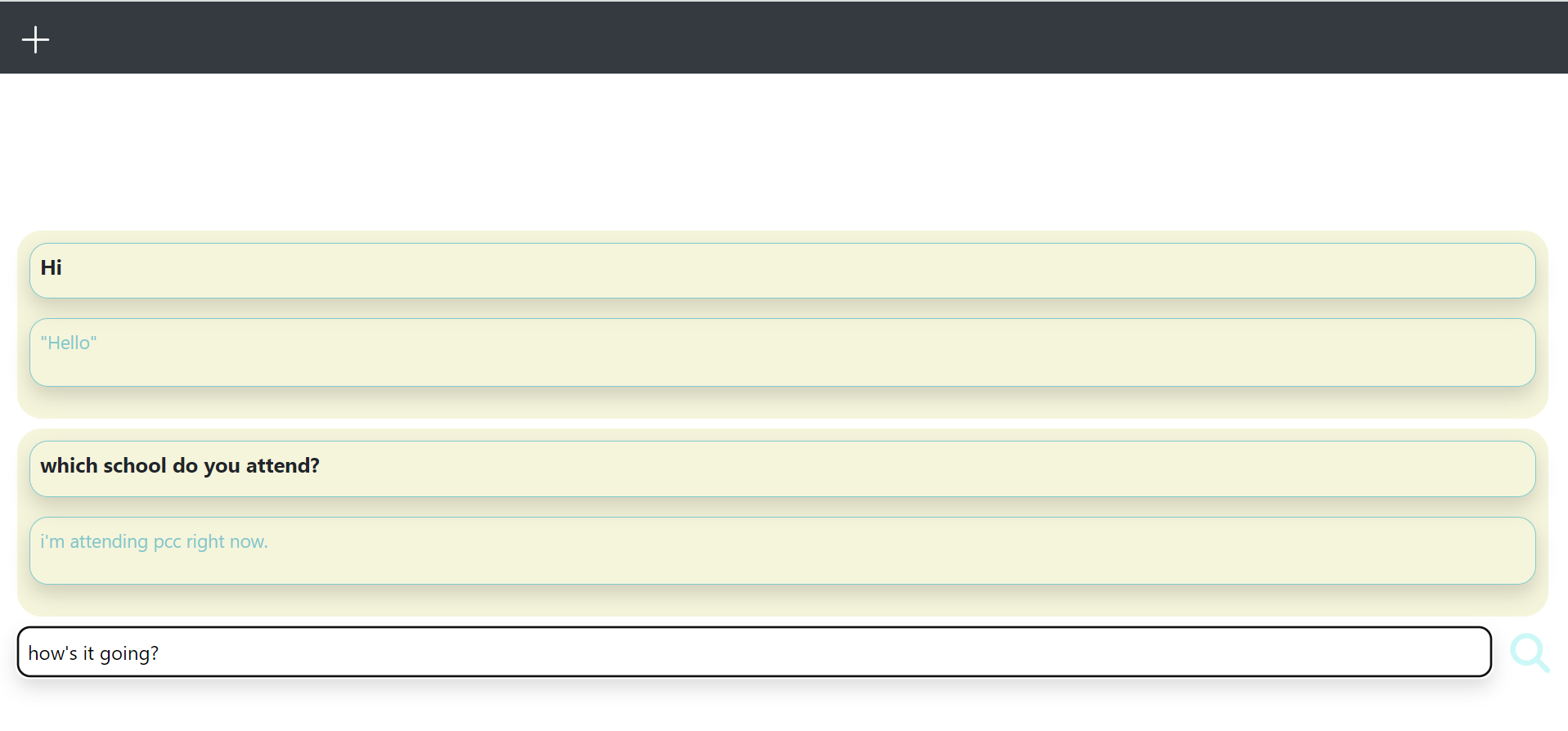
Flask is a micro web framework for building web applications in Python. It is lightweight, easy to use, and allows developers to create web applications quickly and efficiently.

****

The webpage simple responsive one that is built using html and css elements and some bootstrap components.



The web interface would look like the following picture.



# **CONCLUSION:**

# Creating a chatbot in Python is a multifaceted endeavour. It requires data collection, NLP expertise, machine learning model selection, rigorous testing, and a commitment to continuous improvement. User experience and ethical considerations are paramount. Python's versatility and rich ecosystem make it an excellent choice for building intelligent and user-friendly chatbots.