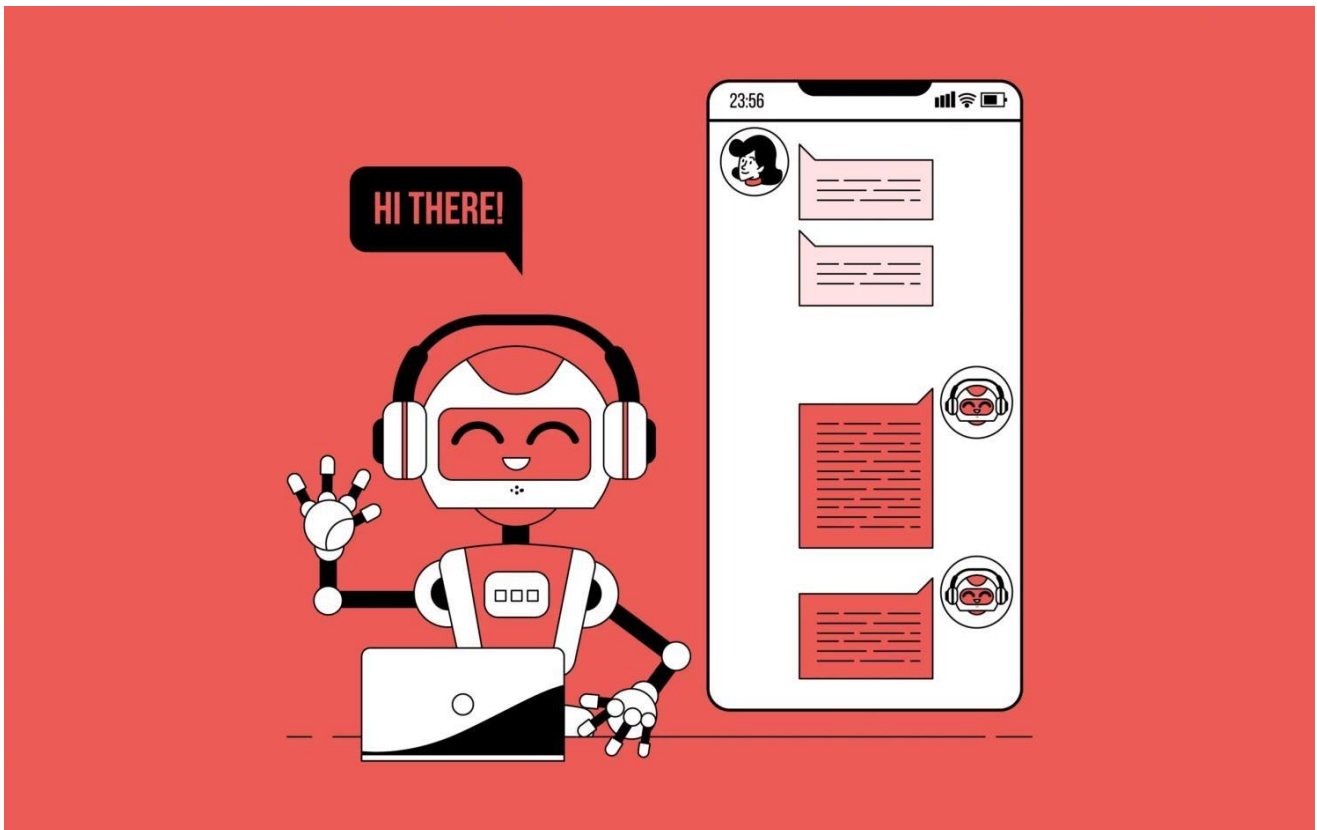# <u>CREATE A CHATBOT IN PYTHON</u>

Domain Name: Artificial Intelligence

Phase-5 Document Submission

Project: To create a Chatbot in Python that provides exceptional answering user queries (diabetes) on a website.

<u>Introduction:</u>

- Chatbot is a technology that arises to be the way of human to interact with computer using natural language (spoken human language).
- They are at the forefront of creating intelligent systems that can understand, process, and generate human language, paving the way

- for applications like chatbots, language translation, sentiment analysis, and more.
- we propose an architectural design of a chatbot that will have the ability to remember the whole conversation flow to be used by diabetic patients for their daily diabetes control activities.
- RNNs can be used to create intelligent conversational agents capable of understanding and generating human-like responses in a dynamic and context-aware manner.
- A friendly and informative companion to help you navigate the world of diabetes. Whether you have questions about diabetes types, management, or just need some guidance, our chatbot is here to provide you with answers and support. Simply start a conversation, and our chatbot will assist you with valuable information and guidance related to diabetes.
- Whether you're looking for advice on diabetes management, information on symptoms and types, or simply need a knowledgeable companion to converse with, our chatbot is here to help.
- We address deployment options and considerations, whether I want to integrate in chatbot into a website, messaging platform, or standalone application. By the end of this guide, It will have the knowledge and skills to embark on my chatbot development journey and create very conversational AI.

Problem Description:

Diabetes is a prevalent and chronic medical condition with serious health consequences. Early detection and proactive management are crucial for preventing complications. The goal of this project is to create a predictive model that assesses an individual's risk of developing diabetes based on their medical data.

Frequent Urination: People with Type 2 diabetes may urinate more frequently than usual, especially at night.

Increased Thirst: If you may find yourself feeling unusually thirsty and drinking more fluids.

Fatigue: Many individuals with Type 2 diabetes experience persistent fatigue and a lack of energy.

Blurred Vision: High blood sugar levels can cause temporary vision problems.

Key Objectives:

1.Early Risk Assessment:

Build a machine learning model capable of analyzing medical data and accurately predicting the likelihood of an individual developing diabetes.

2.Personalized Prevention:

Provide personalized recommendations and preventive measures based on the prediction to help individuals reduce their diabetes risk. These measures may include dietary changes, exercise routines, and monitoring blood sugar levels.

3.User-Friendly Interface:

Develop a user-friendly interface, such as a web application or mobile app, that allows users to input their medical data and receive predictions and recommendations in a clear and understandable format.

4.Data Privacy and Security:

Ensure strict adherence to data privacy regulations and implement robust security measures to protect users' sensitive medical information.

5.Continuous Improvement:

Establish a feedback loop for users to provide information about the effectiveness of recommendations and predictions, allowing for continuous improvement of the system.

Scope:

The scope of this project includes:

- Data collection and preprocessing:

Gathering relevant medical data, cleaning, and preparing it for model training.

- Machine learning model development:

Creating an accurate predictive model using appropriate algorithms and techniques.

- User interface development:

Designing and implementing a user-friendly interface for data input and result presentation.

- Personalized recommendations:

Developing algorithms for generating personalized preventive measures.

- Data privacy and security:

Ensuring the secure handling and storage of user data.

- Continuous improvement:

Implementing mechanisms for user feedback and system enhancement.

Natural Language Processing (NLP):

Implement NLP techniques to understand user queries and generate appropriate responses. This involves tokenization, part-of-speech tagging,

entity recognition, and sentiment analysis.

Example:

```
pip install nltk

import nltk

from nltk.chat.util import Chat, reflections

# Define patterns and responses for the chatbot

patterns = [

    (r'(.*)tell me about Type 2 diabetes(.*)', [

        "Type 2 diabetes is a chronic condition that affects how your body metabolizes

glucose.",

        "Common risk factors include genetics, obesity, and a sedentary lifestyle.",

        "Symptoms may include increased thirst, frequent urination, and fatigue.",

        "Managing blood sugar levels through diet, exercise, and medication is crucial

for treatment.",

    ]),

    (r'(.*)symptoms of diabetes(.*)', [

        "The symptoms of diabetes can include increased thirst, frequent urination,

fatigue, and more.",

        "It's essential to consult a healthcare professional for a proper diagnosis if you
```

```
        suspect diabetes.",

        ]),

        (r'(.*)how to manage diabetes(.*)', [

            "Managing diabetes involves monitoring blood sugar levels, eating a
balanced

diet, and regular exercise.",

            "Medications and insulin may also be prescribed by a healthcare
provider.",

        ]),


        (r'(.*)help(.*)', [

            "I can provide information about diabetes. Just ask me a specific
question, and

I'll do my best to help!",

        ]),

    ]

# Create and start the chatbot

def diabetes_chat():

    print("Hello! I'm your diabetes chatbot. How can I assist you today?")

    chatbot = Chat(patterns, reflections)

    chatbot.converse()

if _name_ ==  "_main_":

    nltk.download("punkt")

    diabetes_chat()
```

Data Preparation:

- Load the dataset into a suitable data structure (e.g., Pandas DataFrame).
- Examine the dataset to understand its structure and distribution.
- Preprocess the data by removing unnecessary characters, converting text to
  lowercase, and handling any missing values.

Example:

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

Input data files are available in the read-only "../input/" directory

For example, running this (by clicking run or pressing Shift+Enter) will list all files

under the input directory

import os

for dirname, _, filenames in os.walk('/kaggle/input'):

   for filename in filenames:

     print(os.path.join(dirname, filename))

You can write up to 20GB to the current directory (/kaggle/working/) that gets

preserved as output when you create a version using "Save & Run All"

You can also write temporary files to /kaggle/temp/, but they won't be saved

outside of the current session

```
import json

with                    open('/kaggle/input/mental-health-conversational-
data/intents.json', 'r') as f:
    data = json.load(f)

df = pd.DataFrame(data['intents'])

df
```

tag patterns responses

0 greeting [Hi, Hey, Is anyone there?, Hi there, Hello, H... [Hello there. Tell me how are you feeling toda...

1 morning [Good morning] [Good morning. I hope you had a good night's s...

2 afternoon [Good afternoon] [Good afternoon. How is your day going?]

3 evening [Good evening] [Good evening. How has your day been?]

4 night [Good night] [Good night. Get some proper sleep, Good night...

... ...

... ...

... ...

75 fact-28 [What do I do if I'm worried about my mental h... [The most important thing is to talk to someon...

76 fact-29 [How do I know if I'm unwell?] [If your beliefs , thoughts , feelings or beha...

77 fact-30 [How can I maintain social connections? What i... [A lot of people are alone right now, but we d...

78 fact-31 [What's the difference between anxiety and str... [Stress and anxiety are often used interchange...

79 fact-32 [What's the difference between sadness and dep... [Sadness is a normal reaction to a loss, disap...

80 rows × 3 columns

```python
dic = {"tag":[], "patterns":[], "responses":[]}
for i in range(len(df)):
    ptrns = df[df.index == i]['patterns'].values[0]
    rspns = df[df.index == i]['responses'].values[0]
    tag = df[df.index == i]['tag'].values[0]
    for j in range(len(ptrns)):
        dic['tag'].append(tag)
        dic['patterns'].append(ptrns[j])
        dic['responses'].append(rspns)
    df = pd.DataFrame.from_dict(dic)
df
```

tag patterns responses

0 greeting Hi [Hello there. Tell me how are you feeling toda...

1 greeting Hey [Hello there. Tell me how are you feeling toda...

2 greeting Is anyone there? [Hello there. Tell me how are you feeling toda...

3 greeting Hi there [Hello there. Tell me how are you feeling toda...

4 greeting Hello [Hello there. Tell me how are you feeling toda...

... ...

… … … …

… … … …

227 fact-29 How do I know if I'm unwell? [If your beliefs , thoughts , feelings or beha...

228 fact-30 How can I maintain social connections? What if... [A lot of people are alone right now, but we d...

229 fact-31 What's the difference between anxiety and stress? [Stress and anxiety are often used interchange...

230 fact-32 What's the difference between sadness and depr... [Sadness is a normal reaction to a loss, disap...

231 fact-32 difference between sadness and depression [Sadness is a normal reaction to a loss, disap...

232 rows × 3 columns

df['tag'].unique()

array(['greeting', 'morning', 'afternoon', 'evening', 'night', 'goodbye',
       'thanks', 'no-response', 'neutral-response', 'about', 'skill',
       'creation', 'name', 'help', 'sad', 'stressed', 'worthless',
       'depressed', 'happy', 'casual', 'anxious', 'not-talking', 'sleep',
       'scared', 'death', 'understand', 'done', 'suicide', 'hate-you',
       'hate-me', 'default', 'jokes', 'repeat', 'wrong', 'stupid',
       'location', 'something-else', 'friends', 'ask', 'problem',
       'no-approach', 'learn-more', 'user-agree', 'meditation',
       'user-meditation', 'pandora-useful', 'user-advice',
       'learn-mental-health', 'mental-health-fact', 'fact-1', 'fact-2',

'fact-3', 'fact-5', 'fact-6', 'fact-7', 'fact-8', 'fact-9',

'fact-10', 'fact-11', 'fact-12', 'fact-13', 'fact-14', 'fact-15',

'fact-16', 'fact-17', 'fact-18', 'fact-19', 'fact-20', 'fact-21',

'fact-22', 'fact-23', 'fact-24', 'fact-25', 'fact-26', 'fact-27',

'fact-28', 'fact-29', 'fact-30', 'fact-31', 'fact-32'],

dtype=object)

NoSQL Databases:

- MongoDB: A widely-used NoSQL database that stores data in a flexible, JSON-like format. It's suitable for handling unstructured or semi-structured data, making it useful for chatbots that manage diverse data types, such as user-generated content, logs, or chat histories.
- MongoDB is a popular NoSQL database that you can use to store data for my chatbot. To interact with MongoDB in Python, it will typically use the pymongo library. Below is a simple example of how to connect to a MongoDB database and perform basic CRUD (Create, Read, Update, Delete) operations in Python.

Install pymongo:

```
pip install pymongo
```

Sample Python code for MongoDB:

```python
import pymongo

# Connect to the MongoDB server

client = pymongo.MongoClient("mongodb://localhost:27017/") # Replace with your MongoDB server URL

# Create or select a database (e.g., "chatbot_db")
```

```
db = client["chatbot_db"]

# Create or select a collection (e.g., "chat_messages")

collection = db["chat_messages"]

# Insert a chat message into the collection

message = {

    "user_id": 1,

    "message": "Hello, chatbot!",

    "timestamp": "2023-10-30T14:00:00"

}

inserted_message = collection.insert_one(message)

# Query chat messages

messages = collection.find({"user_id": 1})

for message in messages:

    print(f"User {message['user_id']} said: {message['message']} at
{message['timestamp']}")

# Close the MongoDB connection

client.close()
```

User Interface:

1. **Web-Based Chat Interface:**

 A web-based chat interface using HTML, CSS, and JavaScript. This is a versatile option, allowing users to interact with the chatbot through a web browser on desktop or mobile devices.

Example:

```
    <div id="chat-container">
```

```
<div id="chat-output"></div>

<input type="text" id="user-input" placeholder="Type a message...">

</div>
```

2. Messaging Platforms Integration(specification):

We can integrate the chatbot with popular messaging platforms like Facebook Messenger, WhatsApp, or Slack. This approach leverages the user-friendly interfaces of these platforms and allows users to interact with your chatbot within their favorite apps.

3. Voice User Interface (VUI):

If the chatbot supports voice interactions, we can create a voice user interface using speech recognition and synthesis technologies. Platforms like Amazon Alexa or Google Assistant can be used for voice-based chatbot interfaces.

4. Mobile App Integration:

Develop a mobile app that includes the chatbot as a feature. Users can interact with the chatbot within the app, and you can design the interface to fit your app's style and design.

5. Command-Line Interface (CLI):

For more technical or developer-oriented chatbots, we can create a command-line interface where users interact with the chatbot by entering text-based commands.

6. AR/VR Interfaces:

If you're building a chatbot for augmented reality (AR) or virtual reality (VR) applications, you can design a 3D or immersive chat interface that aligns with the AR/VR environment.

Best Security Practices When Working With a Chatbot

1. End-to-End Encryption

- It is very important to have E2EE; this ensures that the whole conversation will be encrypted. Data in transit can also be tampered and spoofed; different protocols exist to provide encryption while addressing these issues — we will take a look at them later. Encryption, in some cases, might need to be used with authentication and integrity protection schemes.

- As you may know, chatbots can also connect to channels like Facebook Messenger, Telegram, Slack, etc. In this case, the ideal mitigation for E2E is to only allow chatbots to connect to channels that support data encryption. Article 32 (a) of the General Data Protection Regulation (GDPR) specifically requires that companies take measures to pseudonymize and encrypt personal data.

- The good news is that, thanks to regulations like GDPR, more companies are paying attention to data encryption and, therefore, giving chatbots more channels to connect to in a secure manner. For example: in 2016, Facebook Messenger introduced the new feature called "Secret Conversations" that enables E2E based on Signal Protocol developed by Open Whisper Systems.

2. User Identity Authentication and Authorization

- User authentication is used to identify that a user is verified with valid and secure login credentials, such as a username and password. The credentials are exchanged for a secure authentication token; this token is used throughout the user's session.

- One of the security measures used in chatbots, especially on the banking cases, is authentication timeouts, when the generated token can only be used for a preset amount of time. After that, the system will be forced to make a new one.

- Two-factor authentication is another way to verify the identity of a user by asking them to verify their account through an email and text message. This authentication technique also helps with authorization

by granting the right person access and keeping the information in the right hands.

3. Self-Destructive Messages

- When Sensitive PII (Personally identifiable information) is being transmitted, the message with this information will be destroyed after a set amount of time. This type of security measure is crucial when working with banking and other financial chatbots.

- Article 5 (e) of the General Data Protection Regulation (GDPR) states that personal data shall be kept for no longer than is necessary for the purposes for which it is being processed.

- Another GDPR compliance measure is to have an "intent level" of privacy. The user can exchange personal information but the data will not be revealed even from the backend, only the intents from the user are logged and kept for audit purposes.

4. Secure Protocols

- We know that network security can often be a tough topic; you can think of the purpose of secure protocols as to transfer data from point A to point B without being intercepted, read by unauthorized individuals, altered or deleted.

- HTTPS is the web protocol ensuring the privacy and integrity of our data. This protocol transfers data over Hypertext transfer protocol (HTTP) through an encrypted connection by Secure Sockets Layer (SSL) or Transport Layer Security (TSL).

Diabetes Symptoms:

- Urinate (pee) a lot, often at night
- Are very thirsty
- Lose weight without trying
- Are very hungry
- Have blurry vision
- Have numb or tingling hands or feet
- Feel very tired
- Have very dry skin

- Have sores that heal slowly
- Have more infections than usual

Symptoms of Type 1 Diabetes:

People who have type 1 diabetes may also have nausea, vomiting, or stomach pains. Type 1 diabetes can be diagnosed at any age, and symptoms can develop in just a few weeks or months and can be severe.

Symptoms of Type 2 Diabetes:

Type 2 diabetes symptoms often take several years to develop. Some people don't notice any symptoms at all. Type 2 diabetes usually starts when you're an adult, though more and more children and teens are developing it. Because symptoms are hard to spot, it's important to know the risk factors for type 2 diabetes. Make sure to visit your doctor if you have any of them.

Symptoms of Gestational Diabetes:

Gestational diabetes (diabetes during pregnancy) usually doesn't have any symptoms. If you're pregnant, your doctor should test you for gestational diabetes between 24 and 28 weeks of pregnancy. If needed, you can make changes to protect your health and your baby's health.

Conclusion:

- The availability of a well-structured dataset encompassing various conversations related to diabetes health provides a valuable resource for training chatbot models to offer emotional support to individuals dealing with anxiety and depression.

- By utilizing intents, patterns, and responses, the models can learn to understand user messages and generate empathetic and relevant replies.

- In an age where healthcare information and support are critical, the Diabetes Chatbot stands as a valuable companion on your journey towards understanding, managing, and living well with diabetes.

- This intelligent conversational agent has been designed to offer information, answer questions, and provide guidance to individuals seeking clarity on diabetes-related matters.

- A chatbot for diabetes is a valuable initiative for providing information and support to individuals living with diabetes or seeking information about this medical condition.

- This chatbot offers an accessible, convenient, and responsive way to engage with users in discussions related to diabetes management, types, symptoms, and more.

- Overall, the dataset and the subsequent training of chatbot models enable the creation of innovative tools that bridge the gap in Diabetes health care, providing individuals with a readily available resource for emotional support and guidance.