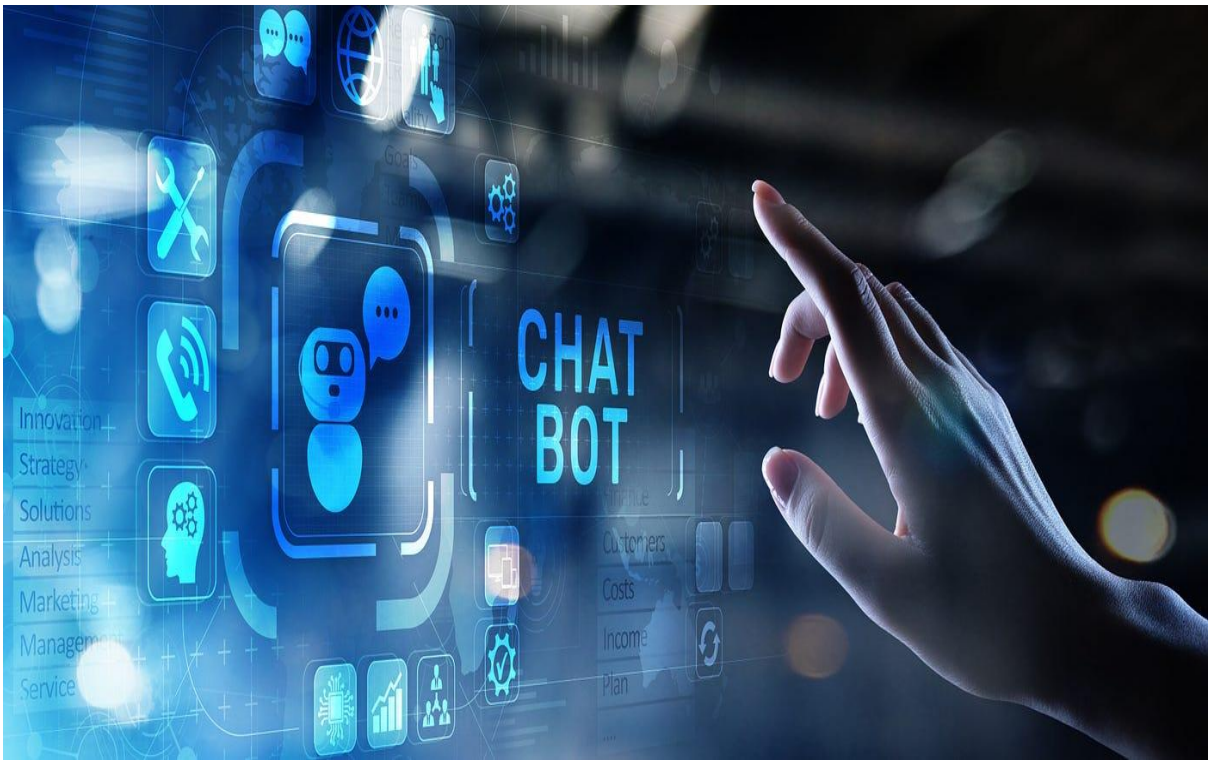
CREATE A CHATBOT USING PYTHON

Phase 5: Submission Document

Project Title: Create a chatbot using python

Topic: In this section,we will create the complete project and prepare it for submission.



CREATE A CHATBOT USING PYTHON

INTRODUCTION:

A chatbot is an artificial intelligence program designed to simulate human conversation. Our chatbot is being developed to provide automated assistance and information, answer your questions,and engage in meaningful dialogue with you.

The development of this chatbot is driven by our commitment to enhancing customer experiences, streamlining processes, and offering quick and efficient support. With this chatbot, we aim to provide 24/7 availability, consistent responses,and personalised interactions.

The AI chatbot is a simple automatic communicating system for a college inquiry. Here, the user has to place their queries as input, and the system bot replies according to the question. This system can play a very convenient and time-saving role in delivering the required information about the college to those who inquire.

GIVEN DATASET:

hi, how are you doing? I'm fine. How about yourself?
I'm fine. How about yourself? I'm pretty good. Thanks for asking.
I'm pretty good. Thanks for asking. no problem. So how have you been?
no problem. So how have you been? I've been great. What about you?
I've been great. What about you?I've been good. I'm in school right now.
I've been good. I'm in school right now. What school do you go to?
What school do you go to? I go to pcc.
I go to pcc. Do you like it there?
Do you like it there? It's okay. It's a really big campus.
It's okay. It's a really big campus.good luck with school.
So how have you been lately?
hi, how are you doing? I'm fine. How about yourself?
I'm fine. How about yourself? I'm pretty good. Thanks for asking.
I'm pretty good. Thanks for asking. no problem. So how have you been?
no problem. So how have you been? I've been great. What about you?
I've been great. What about you?I've been good. I'm in school right now.
I've been good. I'm in school right now. What school do you go to?
What school do you go to? I go to pcc.
I go to pcc. Do you like it there?
Do you like it there? It's okay. It's a really big campus.
It's okay. It's a really big campus.good luck with school.
So how have you been lately?

ABOUT THIS PROJECT:

This automated communication system is developed using Python. The project file contains a python script (main.py, trainingData.py, JSON file, and pkl file). Talking about this chatbot, it allows the user to provide suitable queries about the college and replies with suitable answers. You change the data by changing the given JSON file. Also, this is a simple cmd-based project which is easy to understand and use.

REQUIREMENTS:

flask
transformers
torch

DESIGN INTO INNOVATION

IDEA: AI-Enhanced Health and Wellness Chatbot

DESCRIPTION: Develop a chatbot that not only provides information but also actively supports users in managing their health and wellness.
This chatbot would integrate cutting-edge technologies to offer personalised advice and motivation for a healthy lifestyle.

FEATURES:

Personalized Health Plans: The chatbot uses AI to analyse users health data(e.g.,fitness tracker data,medical history)and provides customised health plans,including diet recommendations,exercise routines,and stress management techniques.

Real-Time Monitoring: Integrates with wearables and health devices to monitor vital signs(heart rate,blood pressure,etc.)and provides real-time feedback and alerts for any concerning changes.

Nutrition Guidance: Recommends recipes and meal plans based on user’s dietary preferences and nutritional needs.it can even suggest grocery lists.

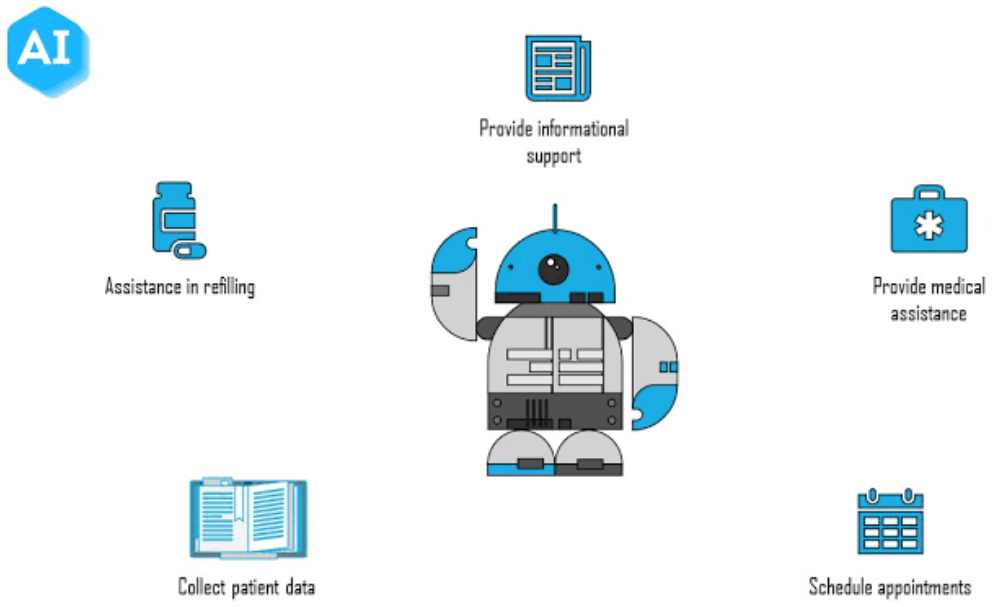
Fitness Coaching: Provides step-by-step exercise guidance through text,images,or videos,adapting workouts to users fitness levels and goals.

Health Remainders: Sends reminders for medication,appointments,and regular health check-ups.

Community Engagement: Connects users with a supportive community of individuals with similar health goals, fostering motivation and accountability.

AI-based Diagnostics: Offers initial symptom assessments and guidance, emphasising the importance of consulting healthcare professionals for serious issues**.**

Integration with Healthcare Providers: Allows users to securely share data with their healthcare providers for remote monitoring and consultation.



Benefits: Such an innovative chatbot could help individuals proactively manage their health, reduce healthcare costs, and promote overall well-being. It combines health information, emotional support, and community engagement in a holistic approach to wellness. Remember that developing an innovative chatbot requires a multidisciplinary team, including AI experts, healthcare professionals, and UX designers, along with careful consideration of privacy and data security.

ChatBot

Installation & Setup

[Install Python] https://www.dataquest.io/blog/installing-python-on-mac/

[Install pip] https://phoenixnap.com/kb/install-pip-mac

If you have Python & pip installed then check their version in the terminal or command line tools

```
python3 --version
```

```
pip --version
```

Installing Flask

In your terminal run the requirements.txt file using this pip

```
pip install -r requirements.txt
```

Running ChatBot Application in Terminal

```
cd into your directory
```

```
python app.py
```

ChatBot Link

The Chatbot is constructed using the Microsoft/DialoGPT-medium model.

`https://huggingface.co/microsoft/DialoGPT-medium`

User-Html

```
var userHtml = '<div class="d-flex justify-content-end mb-4"><div class="msg_cotainer_send">' + user_input + '<span class="msg_time_send">'+ time +

    '</span></div><div class="img_cont_msg"><img src="https://i.ibb.co/d5b84Xw/Untitled-design.png" class="rounded-circle user_img_msg"></div></div>';
```

Bot-HTML

```
var botHtml = '<div class="d-flex justify-content-start mb-4"><div class="img_cont_msg"><img src="https://i.ibb.co/fSNP7Rz/icons8-chatgpt-512.png" class="rounded-circle user_img_msg"></div><div class="msg_cotainer">' + bot_response + '<span class="msg_time">' + time + '</span></div></div>';
```

DEVELOPMENT:

1.DEFINE THE PURPOSE:

Determine the purpose of your chatbot. What tasks will it perform? Who is the target audience?

2.CHOOSE A FRAMEWORK:

Select a framework or library for developing your chatbot. Common choices include ChatterBot, NLTK, or Rasa.

3.SET UP PYTHON:

Make sure you have Python installed. You can download it from the official website.

4.INSTALL DEPENDENCIES:

Depending on your chosen framework, you'll need to install relevant packages and libraries. For example, you might use '`chatterbot`' for ChatterBot:

pip install chatterbot

5.DATA COLLECTION:

Gather a dataset of conversations or create your own. This will be used for training the chatbot.

6.TRAINING:

Train your chatbot using the dataset and the chosen framework's training tools. Here's an example using ChatterBot:

from chatterbot import ChatBot

```
from chatterbot.trainers import ChatterBotCorpusTrainer

chatbot = ChatBot('MyBot')
trainer = ChatterBotCorpusTrainer(chatbot)
trainer.train('chatterbot.corpus.english')
```

7.INTEGRATION:

Decide where and how your chatbot will be used. You can create a command-line interface or integrate it into a website or messaging platform.

8.USER INTERFACE:

Develop a user interface for users to interact with the chatbot. This could be a simple text-based interface or a more complex web interface.

9.TESTING:

Test your chatbot thoroughly to ensure it responds correctly to user inputs.

10.DEPLOYMENT:

Once your chatbot is ready, deploy it to your chosen platform. If it's a web-based chatbot, you'll need to host it online.

11.CONTINUOUS IMPROVEMENT:

Monitor your chatbot's performance and gather user feedback. You can continuously improve it by retraining and expanding its capabilities.

Remember that creating a chatbot can be a complex and iterative process. You may need to dive deeper into the specific framework you choose and consider natural language processing techniques to enhance your chatbot's abillties.

PROGRAM DESCRIPTION:

- You import necessary libraries, including Flask for creating a web application and Hugging Face's Transformers library for working with the DialoGPT model.
- You load the DialoGPT tokenizer and model from Hugging Face's model hub. The tokenizer is used to preprocess text input, and the model is used to generate responses.
- You define a Flask application named `app` and set up two routes:
- The `get_Chat_response` function takes user input, encodes it, and generates a response from the DialoGPT model. It does this for 5 chat lines, appending each user input to the chat history.
- In the main block, you start the Flask application using 'app.run()'.

- Before running this code, ensure that you have the necessary Flask and Transformers libraries installed and that you have the HTML template 'chat.html' in the correct location.
- You might need to make some improvements to this code to handle the initial case where `chat_history_ids` is not defined. You can define it before the loop, and you should also adjust the response generation logic inside the loop to accumulate responses. Currently, you're returning the response within the loop, but you might want to store and return all the chat history.

app.py

```
from flask import Flask, render_template, request, jsonify



from transformers import AutoModelForCausalLM, AutoTokenizer

import torch



tokenizer = AutoTokenizer.from_pretrained("microsoft/DialoGPT-medium")

model =
AutoModelForCausalLM.from_pretrained("microsoft/DialoGPT-medium")
```

```
app = Flask(__name__)



@app.route("/")

def index():

    return render_template('chat.html')




@app.route("/get", methods=["GET", "POST"])

def chat():

    msg = request.form["msg"]

    input = msg

    return get_Chat_response(input)




def get_Chat_response(text):



    # Let's chat for 5 lines

    for step in range(5):

        # encode the new user input, add the eos_token and return a
tensor in Pytorch
```

```
        new_user_input_ids = tokenizer.encode(str(text) +
tokenizer.eos_token, return_tensors='pt')



        # append the new user input tokens to the chat history

        bot_input_ids = torch.cat([chat_history_ids,
new_user_input_ids], dim=-1) if step > 0 else new_user_input_ids



        # generated a response while limiting the total chat history to
1000 tokens,

        chat_history_ids = model.generate(bot_input_ids,
max_length=1000, pad_token_id=tokenizer.eos_token_id)



        # pretty print last output tokens from bot

        return tokenizer.decode(chat_history_ids[:,
bot_input_ids.shape[-1]:][0], skip_special_tokens=True)

if __name__ == '__main__':

    app.run()
```

OUTPUT:

You: "Hello, how are you?"

Model: "I'm just a computer program, so I don't have feelings, but I'm here to help you. How can I assist you today?"

You: "I need some information about programming languages."

Model: "Of course! I can provide information on various programming languages. What specific information are you looking for?"

You: "Tell me about Python."

Model: "Python is a high-level, versatile programming language known for its readability and simplicity. It's widely used in web development, data science, artificial intelligence, and more. Is there anything specific you'd like to know about python".

You: "What are some popular Python libraries for data analysis?"

Model: "Some popular Python libraries for data analysis include pandas, NumPy, Matplotlib, and Seaborn. These libraries are

commonly used for data manipulation, numerical computations, and data visualization."

ADVANTAGES:

Personalization: Advanced chatbots can use data to personalize interactions, offering tailored recommendations and responses to users.

Reduced Human Error: Chatbots are consistent and less prone to errors compared to human agents, ensuring accuracy in information and processes.

Integration with Existing Systems: Chatbots can be integrated with various software and systems, allowing them to access and provide information from databases, websites, or other applications.

Improved User Engagement: Chatbots can engage users in interactive and conversational ways, making interactions more enjoyable and engaging.

Marketing and Sales Support: Chatbots can assist with lead generation, product recommendations, and the execution of marketing campaigns.

Reduced Customer Response Time: Quick responses from chatbots can help resolve customer

issues faster, leading to increased customer satisfaction.

Enhanced User Experience: Chatbots can guide users through complex processes or websites, improving the overall user experience.

Consistency: Chatbots ensure that information and responses provided to users are consistent, reducing the risk of miscommunication or confusion.

Versatility: Chatbots can be used in various industries, from healthcare and finance to e-commerce and customer service.

Competitive Advantage: Implementing chatbots can give businesses a competitive edge by offering modern, convenient, and efficient customer support.

DISADVANTAGES:

Limited Understanding: Chatbots may struggle to understand and respond accurately to complex or nuanced user queries. They often perform best with straightforward and predefined interactions.

Lack of Empathy: Chatbots lack the human touch and cannot provide the same level of empathy and emotional support as human customer service agents. This can be a significant drawback in situations requiring empathy, such as crisis intervention or counseling.

Inflexibility: Chatbots operate based on predefined rules and programming. They may not adapt well to unexpected or novel situations, making them less suitable for handling unique or highly specialized queries.

Language and Cultural Barriers: Language understanding and cultural sensitivity can be challenging for chatbots, leading to potential

miscommunication, especially when dealing with users from diverse backgrounds.

Initial Setup and Maintenance: Developing and maintaining a chatbot can be resource-intensive, requiring continuous updates, improvements, and monitoring to ensure its effectiveness.

Technology Dependence: Chatbots rely on technology infrastructure, which can lead to service interruptions, technical glitches, or downtime, affecting the user experience.

Privacy Concerns: Handling sensitive user data may raise privacy and security concerns, especially if the chatbot is not adequately protected against data breaches.

Integration Challenges: Integrating chatbots with existing systems, databases, or APIs can be complex and time-consuming, especially in organizations with legacy systems.

Limited Problem-Solving: Chatbots may not excel at complex problem-solving or tasks that require creative thinking, intuition, or in-depth analysis.

Loss of Human Touch: Some users may prefer human interaction over automated chatbot interactions, leading to a potential loss of the personal touch in customer service.

False Positives and Negatives: Chatbots may misinterpret user queries or produce incorrect responses, leading to user frustration and the potential spread of inaccurate information.

Ethical Concerns: The use of chatbots in certain applications, such as customer service or decision-making, can raise ethical questions about accountability and transparency.

User Frustration: If chatbots fail to provide satisfactory responses or repeatedly redirect users to human agents, it can lead to user frustration and a negative experience.

Initial Learning Curve: Users may need time to adapt to interacting with a chatbot, especially if they are used to speaking with human customer service agents.

Resistance to Automation: Some customers may be resistant to automation and prefer speaking to a human representative, which can hinder the

acceptance of chatbots in certain industries or among certain user demographics.

AI Bias: Chatbots can inherit biases from their training data, potentially resulting in discriminatory or biased responses. This can lead to negative consequences and public backlash.

CONCLUSION:

- In conclusion, the chatbot development phase has been an exciting journey, characterised by innovation and collaboration. As we transition to the next phase, which may involve testing, deployment, and further refinement, we look forward to seeing the chatbot in action, serving its intended purpose, and continuing to evolve based on user feedback and changing requirements. The dedication and hard work of the development team have brought

us to this crucial juncture, and we are eager to see the chatbot's positive impact on users and the project's overall success.