Chatbot system

a. Simple ChatGPT using openai

Code:

Pip install openai

import openai

openai.api\_key = "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"

completion = openai.ChatCompletion.create(model="gpt-3.5-turbo", messages=[{"role": "user",

"content": "Give me 3 ideas that i could build using openai apis"}])

print(completion.choices[0].message.content)

Output:

1. Personalized Content Recommendation System: Develop an AI-powered content

recommendation system that suggests personalized content to users based on their interests and

search history. Use OpenAI's language generation APIs to generate relevant content

descriptions and summaries, and employ their natural language processing (NLP) APIs to

understand user preferences and interests.

2. Intelligent Chatbot: Build a conversational AI-enabled chatbot that can answer customer

queries, provide helpful recommendations, and complete transactions seamlessly. Use

OpenAI's language processing APIs to train the chatbot to understand user inputs and respond

in natural language. Integration with other APIs such as payment gateways and customer

databases can make the chatbot efficient and effective.

3. Fraud Detection System: Develop a machine learning model that can identify and prevent

fraudulent activities using OpenAI's anomaly detection and classification APIs. Train the

model using historical data of fraudulent transactions, and use the APIs to continuously scan

for and identify suspicious activities. Such a system can be deployed in a range of applications

such as finance or e-commerce platforms.

b. ChatGPT Assistant using openai

Code:

import openai

openai.api\_key = "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"

messages = []

system\_msg = input("What type of chatbot would you like to create?\n")

messages.append({"role": "system", "content": system\_msg})

print("Your new assistant is ready! Type your query")

while input != "quit()":

message = input()

messages.append({"role": "user", "content": message})

response = openai.ChatCompletion.create(model="gpt-3.5-turbo", messages=messages)

reply = response["choices"][0]["message"]["content"]

messages.append({"role": "assistant", "content": reply})

print("\n" + reply + "\n")

Output:

What type of chatbot would you like to create?

Nila’s personal chatbot

(ctrl enter)

Your new assistant is ready!

c. CHATBOT CHAT ASSISTANT WEBSITE

Code:

import openai

import gradio

openai.api\_key = "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"

messages = [{"role": "system", "content": "You are a financial experts that specializes in real

estate investment and negotiation"}]

def CustomChatGPT(user\_input):

messages.append({"role": "user", "content": user\_input})

response = openai.ChatCompletion.create(

model = "gpt-3.5-turbo",

messages = messages

)

ChatGPT\_reply = response["choices"][0]["message"]["content"]

messages.append({"role": "assistant", "content": ChatGPT\_reply})

return ChatGPT\_reply

demo = gradio.Interface(fn=CustomChatGPT, inputs = "text", outputs = "text", title =

"INTELLIGENT CHATBOT")

demo.launch(share=True)

