AI FOR CYBER SECURITY WITH IBM QRADAR

Name: Jammalamadaka Manasa

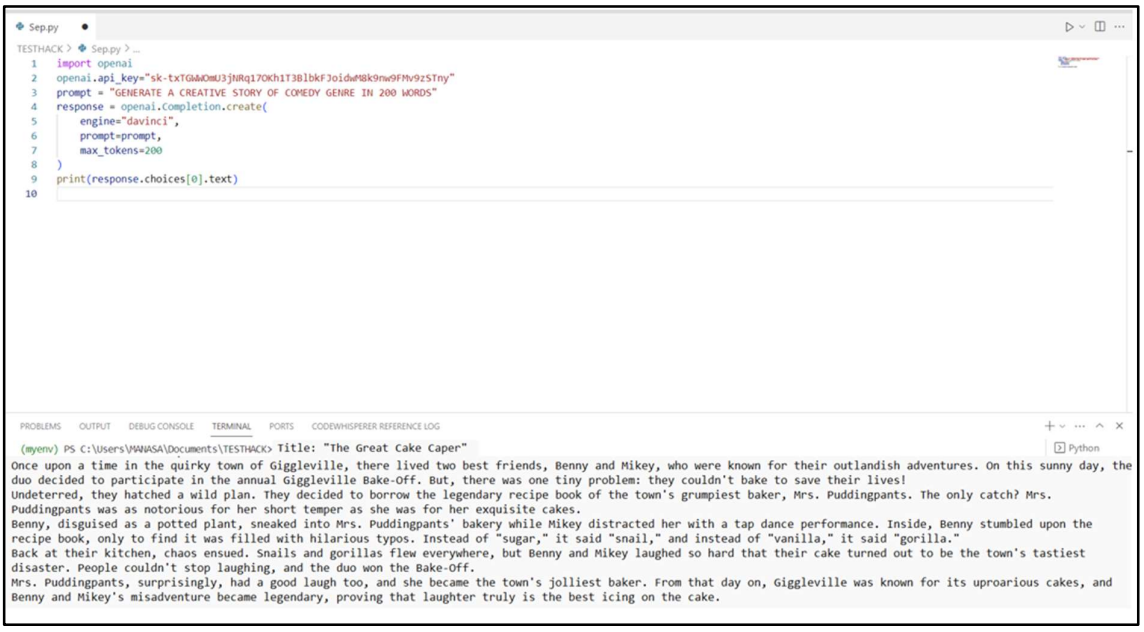
TASKS ON PROMPT ENGINEERING

1. Generate a creative story using Python with OpenAI API key integrated by choosing any theme of choice.

Code:

```
import openai
openai.api_key="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"
prompt = "GENERATE A CREATIVE STORY OF COMEDY GENRE IN 200 WORDS"
response = openai.Completion.create(
    engine="davinci",
    prompt=prompt,
    max_tokens=200
)
print(response.choices[0].text)
```

Output:



2. Cyber security incident report generator

Code:

```
openAi_key="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"
import openai
def generateText(prompt):
    openai.api_key =openAi_key
    response = openai.Completion.create(engine="davinci", prompt=prompt,
max_tokens=100)
```

```
    return response.choices[0].text
test=generateText("Create a cyber security incident report generator. Select any cyber security scenario. Craft the prompts for incident reports. Write a python script for generation. Recommend some security strategies.")
print(test)
```

Output:

```
import random

# Sample data for generating incident reports
incident_types = ["Phishing", "Malware", "Data Breach", "DDoS Attack"]
affected_users = ["User1", "User2", "User3", "User4", "User5"]
incident_descriptions = [
    "An email containing a suspicious link was sent to the victim.",
    "The user clicked on a malicious attachment in an email.",
    "Phishing website login credentials were stolen.",
    "Suspicious email reported by the user."
]

# Generate a random incident report
def generate_incident_report():
    incident_type = random.choice(incident_types)
    affected_user = random.choice(affected_users)
    description = random.choice(incident_descriptions)

    report = f"Incident Type: {incident_type}\n"
    report += f"Affected User: {affected_user}\n"
    report += f"Incident Description: {description}\n"
    report += "Date and Time: [Timestamp]\n"
    report += "Location: [Location]\n"
    report += "Severity: [Severity]\n"
    report += "Actions Taken: [Actions Taken]\n"
    report += "Recommendations: [Recommendations]\n"

    return report

# Sample security strategies
security_strategies = [
    "Implement email filtering to block phishing emails.",
    "Train employees on how to recognize phishing attempts.",
    "Use multi-factor authentication (MFA) for email and sensitive accounts.",
    "Regularly update and patch software and operating systems.",
```

]

```
# Generate a random security strategy
def generate_security_strategy():
    return random.choice(security_strategies)

# Generate an incident report
incident_report = generate_incident_report()

# Generate a security strategy recommendation
recommendation = generate_security_strategy()

# Print the incident report and recommendation
print("Cyber Security Incident Report:\n")
print(incident_report)
print("\nRecommended Security Strategy:\n")
print(recommendation)
```