

# AI FOR CYBER SECURITY WITH IBM QRADAR

**Name:** Jammalamadaka Manasa

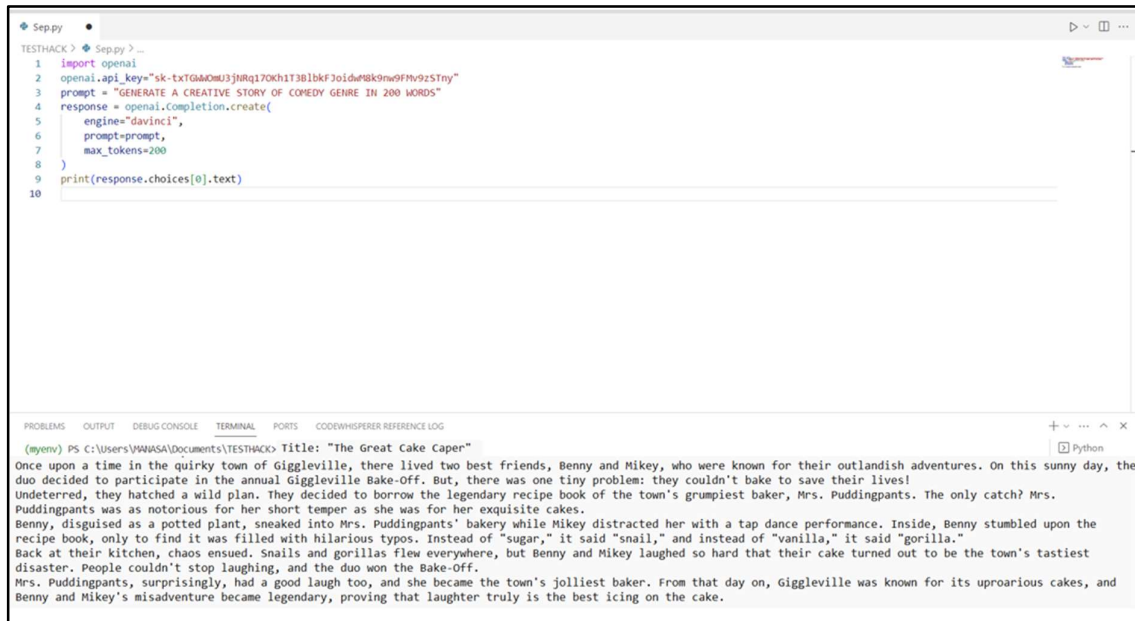
## TASKS ON PROMPT ENGINEERING

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:**



```
1 import openai
2 openai.api_key="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"
3 prompt = "GENERATE A CREATIVE STORY OF COMEDY GENRE IN 200 WORDS"
4 response = openai.Completion.create(
5     engine="davinci",
6     prompt=prompt,
7     max_tokens=200
8 )
9 print(response.choices[0].text)
10
```

(myenv) PS C:\Users\MANASA\Documents\TESTHACK> Title: "The Great Cake Caper"

Once upon a time in the quirky town of Giggleville, there lived two best friends, Benny and Mikey, who were known for their outlandish adventures. On this sunny day, the duo decided to participate in the annual Giggleville Bake-Off. But, there was one tiny problem: they couldn't bake to save their lives! Undeterred, they hatched a wild plan. They decided to borrow the legendary recipe book of the town's grumpiest baker, Mrs. Puddingpants. The only catch? Mrs. Puddingpants was as notorious for her short temper as she was for her exquisite cakes.

Benny, disguised as a potted plant, sneaked into Mrs. Puddingpants' bakery while Mikey distracted her with a tap dance performance. Inside, Benny stumbled upon the recipe book, only to find it was filled with hilarious typos. Instead of "sugar," it said "snail," and instead of "vanilla," it said "gorilla."

Back at their kitchen, chaos ensued. Snails and gorillas flew everywhere, but Benny and Mikey laughed so hard that their cake turned out to be the town's tastiest disaster. People couldn't stop laughing, and the duo won the Bake-Off.

Mrs. Puddingpants, surprisingly, had a good laugh too, and she became the town's jolliest baker. From that day on, Giggleville was known for its uproarious cakes, and Benny and Mikey's misadventure became legendary, proving that laughter truly is the best icing on the cake.

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)
```



```
task2.py
1 openai_key="sk-FytlvwG8T8sZKtSC3p7T3B184Fh4tcxUWqjby1xs6jYVW6"
2 import openai
3 def generateText(prompt):
4     openai.api_key = openai_key
5     response = openai.completion.create(engine="davinci", prompt=prompt, max_tokens=100)
6     return response.choices[0].text
7 test=generateText("Create a cyber security incident report generator. Select any cyber security scenario. Craft the prompts for incident reports. Write a python script fo
8 print(test)

Sep27.py
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"
```