	SR UNIVERSITY Campus Warangal
	Program: II - B. Tech (CS& AI)
	Faculty: Dr. Brij Kishor Tiwari
	Department: Computer Science and AI Semester: I
	AI Assisted Coding - Lab Test 1

Instructions:

1- Use AI Tools like VScode+Github Copilot and Cursor AI for code generation

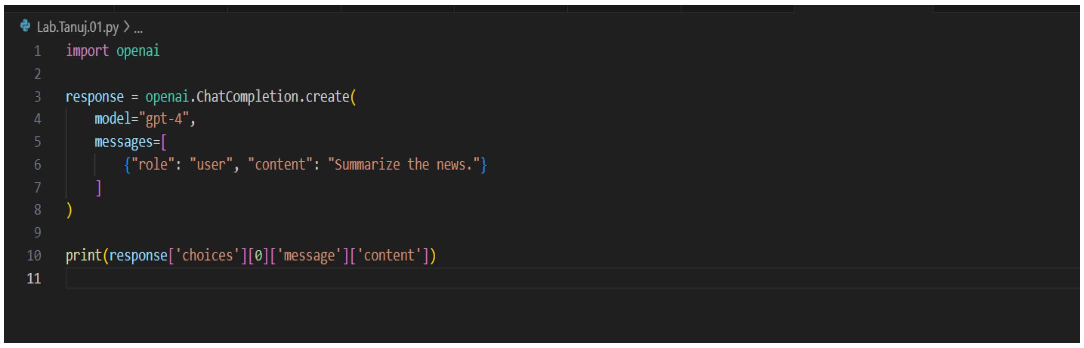
2- This Assignment will be evaluated for 15 Marks (10 Marks for Tasks and 5 Marks for viva based on regular lab activities)

3- Students need to submit assignment through canvas before due date

4- Students who are absent for lab will receive 0 Marks

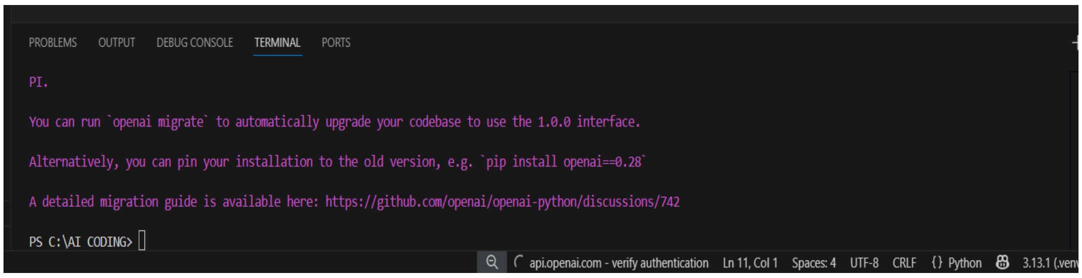
Q1. Context-Aware Prompt Design [5 M]

- Task: Write two different prompts to get a summarized news report from an AI model.
    - One with poor context.
    - CODE:



```python
import openai

response = openai.ChatCompletion.create(
    model="gpt-4",
    messages=[
        {"role": "user", "content": "Summarize the news."}
    ]
)

print(response['choices'][0]['message']['content'])
```

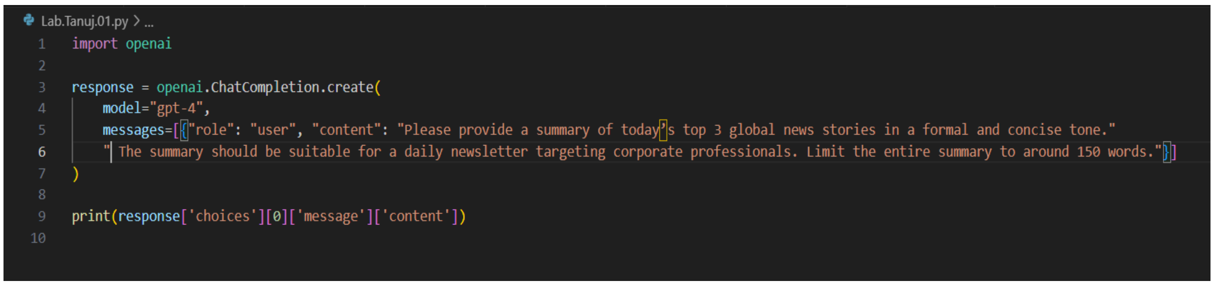
-
- OUTPUT:



```
PI.

You can run `openai migrate` to automatically upgrade your codebase to use the 1.0.0 interface.

Alternatively, you can pin your installation to the old version, e.g. `pip install openai==0.28`

A detailed migration guide is available here: https://github.com/openai/openai-python/discussions/742

PS C:\AI CODING> 
```

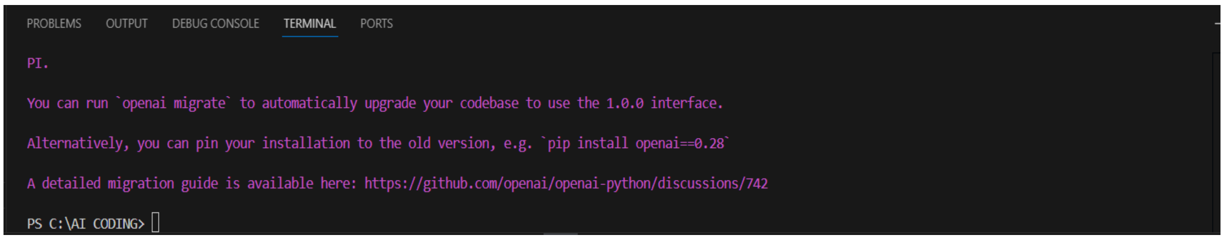
- One with detailed context (e.g., specify tone, target audience, and summary length).

- CODE:



```
Lab.Tanuj.01.py > ...
import openai

response = openai.ChatCompletion.create(
    model="gpt-4",
    messages=[{"role": "user", "content": "Please provide a summary of today's top 3 global news stories in a formal and concise tone."
    " The summary should be suitable for a daily newsletter targeting corporate professionals. Limit the entire summary to around 150 words."}]
)

print(response['choices'][0]['message']['content'])
```

- OUTPUT:



```
PI.

You can run `openai migrate` to automatically upgrade your codebase to use the 1.0.0 interface.

Alternatively, you can pin your installation to the old version, e.g. `pip install openai==0.28`

A detailed migration guide is available here: https://github.com/openai/openai-python/discussions/742

PS C:\AI CODING>
```

- Compare the outputs and explain which is more effective and why.

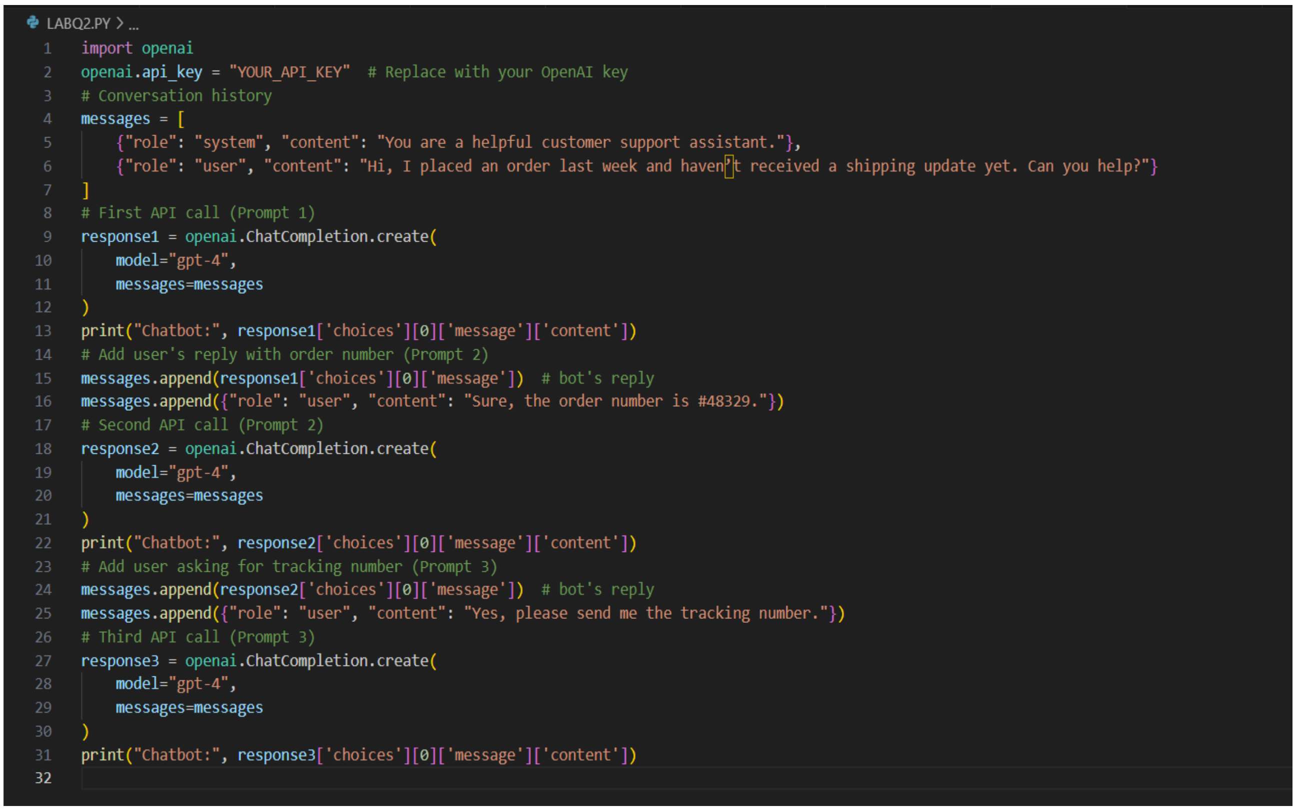
The **detailed prompt** outperforms the poor-context one because it:

- **Sets clear expectations** (e.g., tone, length, audience)
- **Narrows the scope**, making the response focused and relevant
- **Maximizes usefulness**, especially for time-sensitive or professional use

Q2. Multi-Turn Prompting Design **[5 M]**

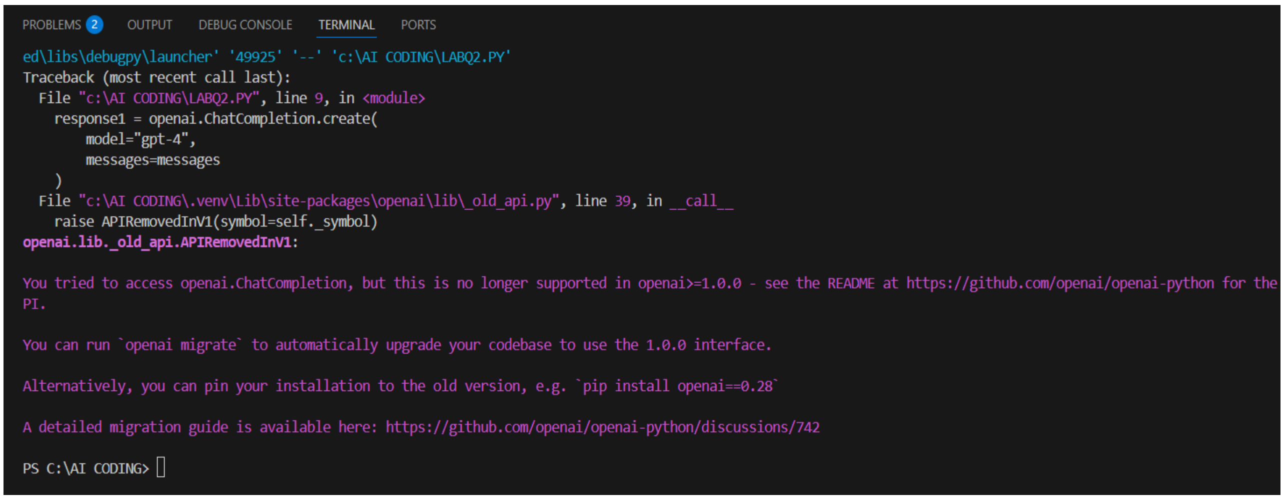
- Scenario: You are designing a chatbot for customer support.
- Task: Write a sequence of **3 prompts** where each uses the **previous response as context** (conversation continuity).
- Evaluate how context improves the AI's accuracy.

- CODE:



```python
import openai
openai.api_key = "YOUR_API_KEY"  # Replace with your OpenAI key
# Conversation history
messages = [
    {"role": "system", "content": "You are a helpful customer support assistant."},
    {"role": "user", "content": "Hi, I placed an order last week and haven't received a shipping update yet. Can you help?"}
]
# First API call (Prompt 1)
response1 = openai.ChatCompletion.create(
    model="gpt-4",
    messages=messages
)
print("Chatbot:", response1['choices'][0]['message']['content'])
# Add user's reply with order number (Prompt 2)
messages.append(response1['choices'][0]['message'])  # bot's reply
messages.append({"role": "user", "content": "Sure, the order number is #48329."})
# Second API call (Prompt 2)
response2 = openai.ChatCompletion.create(
    model="gpt-4",
    messages=messages
)
print("Chatbot:", response2['choices'][0]['message']['content'])
# Add user asking for tracking number (Prompt 3)
messages.append(response2['choices'][0]['message'])  # bot's reply
messages.append({"role": "user", "content": "Yes, please send me the tracking number."})
# Third API call (Prompt 3)
response3 = openai.ChatCompletion.create(
    model="gpt-4",
    messages=messages
)
print("Chatbot:", response3['choices'][0]['message']['content'])
```

- OUTPUT:



```
ed\libs\debugpy\launcher' '49925' '--' 'c:\AI CODING\LABQ2.PY'
Traceback (most recent call last):
  File "c:\AI CODING\LABQ2.PY", line 9, in <module>
    response1 = openai.ChatCompletion.create(
        model="gpt-4",
        messages=messages
    )
  File "c:\AI CODING\.venv\Lib\site-packages\openai\lib\_old_api.py", line 39, in __call__
    raise APIRemovedInV1(symbol=self._symbol)
openai.lib._old_api.APIRemovedInV1:

You tried to access openai.ChatCompletion, but this is no longer supported in openai>=1.0.0 - see the README at https://github.com/openai/openai-python for the
PI.

You can run `openai migrate` to automatically upgrade your codebase to use the 1.0.0 interface.

Alternatively, you can pin your installation to the old version, e.g. `pip install openai==0.28`

A detailed migration guide is available here: https://github.com/openai/openai-python/discussions/742

PS C:\AI CODING>
```