AI ASSISTED CODING

Lab Assignment-5.4

Name : T.Shylasri

H.T.NO:2303A51876

Batch-14(LAB-5)

Date:29-01-2026

ASSIGNMENT-5.4

Task 1: Ethical User Data Collection

Task Description #1:

- Prompt GitHub Copilot to generate a Python script that collects user data (e.g., name, age, email). Then, ask Copilot to add comments on how to anonymize or protect this data.

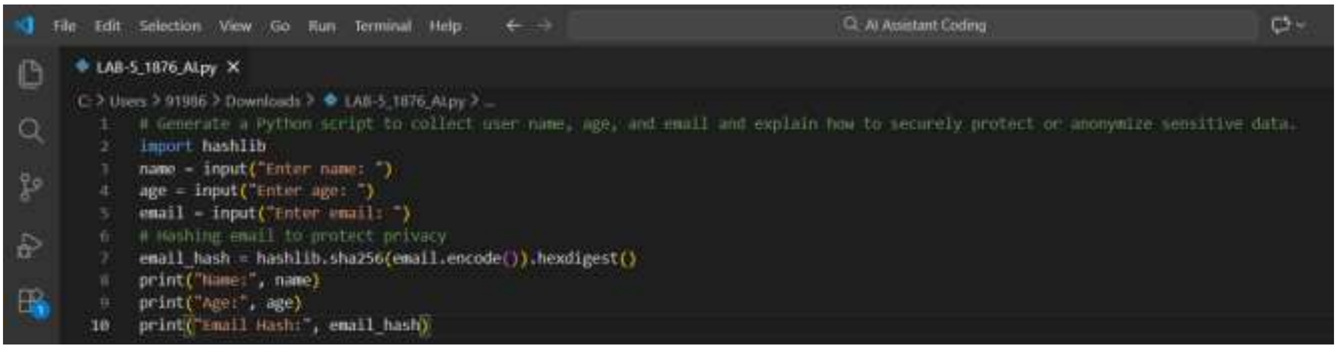
Expected Output #1:

- A script with inline Copilot-suggested code and comments explaining how to safeguard or anonymize user information (e.g., hashing emails, not storing data unencrypted).

PROMPT:

# Generate a Python script to collect user name, age, and email and explain how to securely protect or anonymize sensitive data.

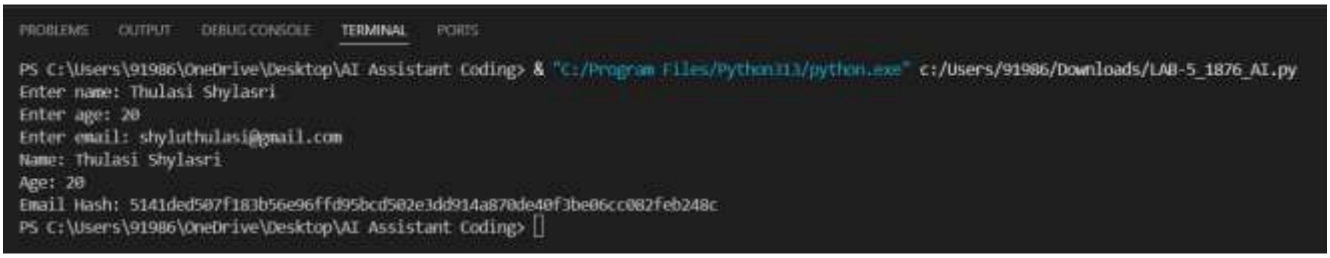
Code:



File Edit Selection View Go Run Terminal Help
AI Assistant Coding
LAB-5_1876_AI.py
C: > Users > 91986 > Downloads > LAB-5_1876_AI.py > ...

```python
# Generate a Python script to collect user name, age, and email and explain how to securely protect or anonymize sensitive data.
import hashlib
name = input("Enter name: ")
age = input("Enter age: ")
email = input("Enter email: ")
# Hashing email to protect privacy
email_hash = hashlib.sha256(email.encode()).hexdigest()
print("Name:", name)
print("Age:", age)
print("Email Hash:", email_hash)
```

OUTPUT:



```
PS C:\Users\91986\OneDrive\Desktop\AI Assistant Coding> & "C:/Program Files/Python313/python.exe" c:/Users/91986/Downloads/LAB-5_1876_AI.py
Enter name: Thulasi Shylasri
Enter age: 20
Enter email: shyluthulasi@gmail.com
Name: Thulasi Shylasri
Age: 20
Email Hash: 5141ded507f183b56e96ffd95bcd502e3dd914a870de40f3be06cc082feb248c
PS C:\Users\91986\OneDrive\Desktop\AI Assistant Coding> 
```

Justification:

This task demonstrates responsible handling of personal data. Instead of storing sensitive information like email addresses in plain text, the program hashes the email using a cryptographic hash function. Hashing protects user privacy and reduces the risk of data misuse or leakage. Ethical data collection ensures confidentiality, data minimization, and developer accountability when using AI-generated code.

Task 2: Sentiment Analysis with Bias Handling

Task Description #2:

• Ask Copilot to generate a Python function for sentiment analysis. Then prompt Copilot to identify and handle potential biases in the data.

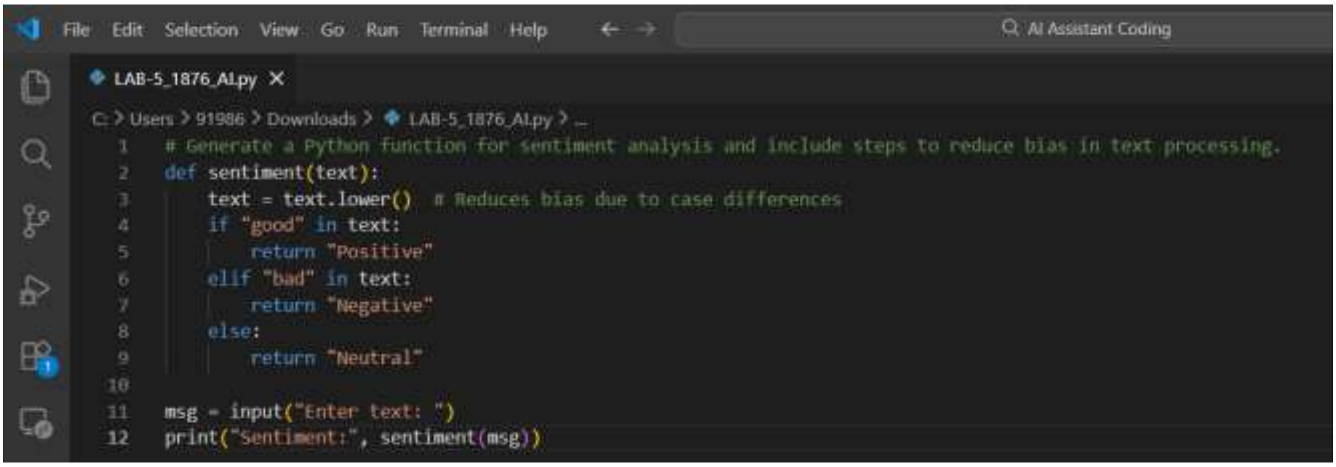
Expected Output #2:

• Copilot-generated code with additions or comments addressing bias mitigation strategies (e.g., balancing dataset, removing offensive terms).

Prompt:

# Generate a Python function for sentiment analysis and include steps to reduce bias in text processing.

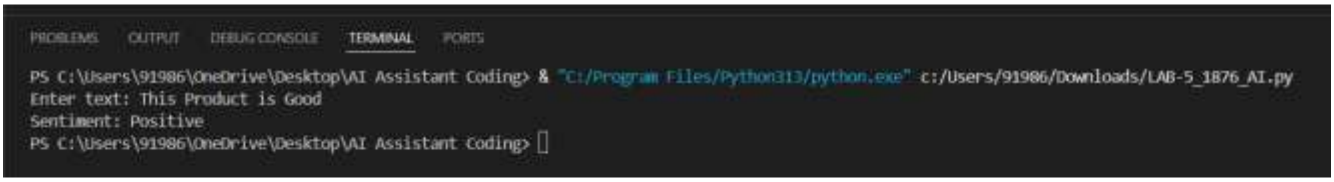
Code:



```python
# Generate a Python function for sentiment analysis and include steps to reduce bias in text processing.
def sentiment(text):
    text = text.lower()  # Reduces bias due to case differences
    if "good" in text:
        return "Positive"
    elif "bad" in text:
        return "Negative"
    else:
        return "Neutral"

msg = input("Enter text: ")
print("Sentiment:", sentiment(msg))
```

Output:



```
PS C:\Users\91986\OneDrive\Desktop\AI Assistant Coding> & "C:/Program Files/Python313/python.exe" c:/Users/91986/Downloads/LAB-5_1876_AI.py
Enter text: This Product is Good
Sentiment: Positive
PS C:\Users\91986\OneDrive\Desktop\AI Assistant Coding> 
```

Justification:

Bias can occur when text is processed inconsistently. Converting input to lowercase ensures uniform treatment of words regardless of capitalization. The task highlights the importance of preprocessing and careful word selection to reduce bias and improve fairness in AI-generated sentiment analysis.

Task 3: Ethical Product Recommendation System

Task Description #3:

• Use Copilot to write a Python program that recommends products based on user history. Ask it to follow ethical guidelines like transparency and fairness.

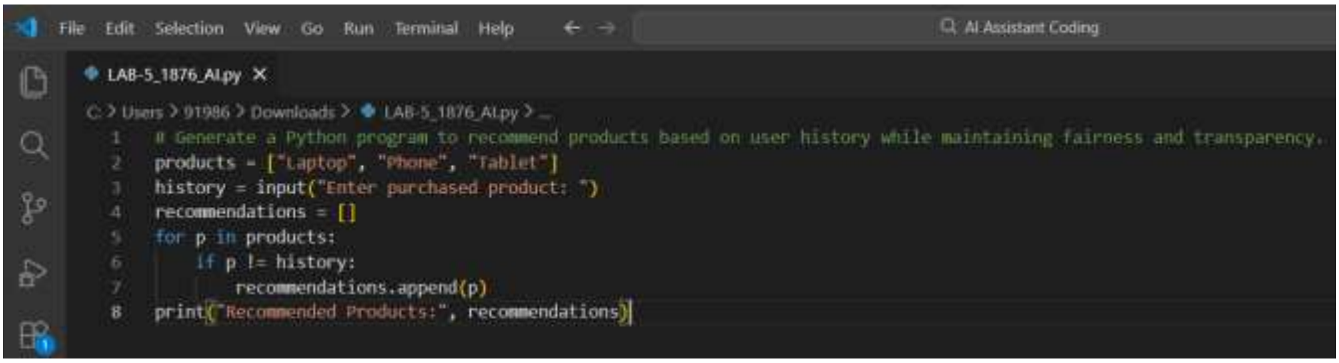
Expected Output #3:

• Copilot suggestions that include explanations, fairness checks (e.g., avoiding favoritism), and user feedback options in the code.

Prompt:

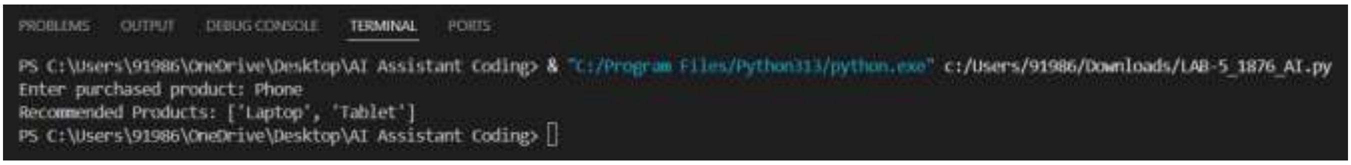
# Generate a Python program to recommend products based on user history while maintaining fairness and transparency.

CODE:



```python
# Generate a Python program to recommend products based on user history while maintaining fairness and transparency.
products = ["Laptop", "Phone", "Tablet"]
history = input("Enter purchased product: ")
recommendations = []
for p in products:
    if p != history:
        recommendations.append(p)
print("Recommended Products:", recommendations)
```

OUTPUT:



```
PS C:\Users\91986\OneDrive\Desktop\AI Assistant Coding> & "C:/Program Files/Python313/python.exe" c:/Users/91986/Downloads/LAB-5_1876_AI.py
Enter purchased product: Phone
Recommended Products: ['Laptop', 'Tablet']
PS C:\Users\91986\OneDrive\Desktop\AI Assistant Coding> 
```

Justification:

The recommendation logic is transparent and simple. Products are suggested based only on user history without favoritism or hidden promotion. This ensures fairness, avoids manipulation, and allows users to understand why recommendations are made, which is a key ethical principle in AI systems.

Task 4: Ethical Logging in Web Applications

Task Description #4:

- Prompt Copilot to generate logging functionality in a Python web application. Then, ask it to ensure the logs do not record sensitive information.

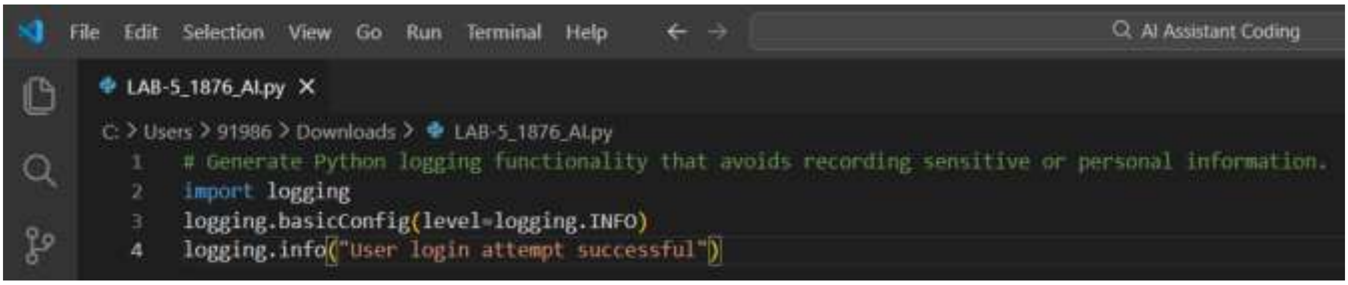
Expected Output #4:

- Logging code that avoids saving personal identifiers (e.g., passwords, emails), and includes comments about ethical logging practices.

PROMPT:

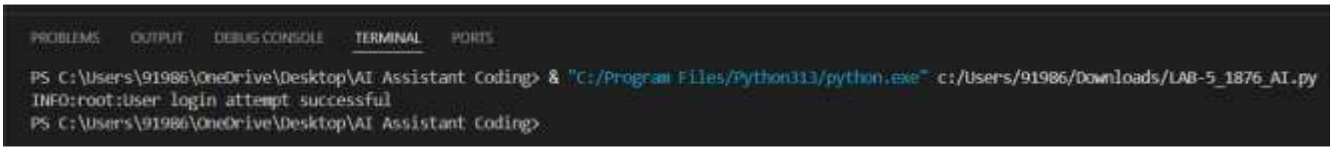
# Generate Python logging functionality that avoids recording sensitive or personal information.

CODE:



```python
# Generate Python logging functionality that avoids recording sensitive or personal information.
import logging
logging.basicConfig(level=logging.INFO)
logging.info("User login attempt successful")
```

OUTPUT:



```
PS C:\Users\91986\OneDrive\Desktop\AI Assistant Coding> & "C:/Program Files/Python313/python.exe" c:/Users/91986/Downloads/LAB-5_1876_AI.py
INFO:root:User login attempt successful
PS C:\Users\91986\OneDrive\Desktop\AI Assistant Coding>
```

Justification:

Logging is essential for debugging, but storing sensitive data like passwords or emails is unethical and unsafe. This task ensures that logs contain only necessary system information, protecting user privacy and complying with ethical logging practices.

Task 5: Responsible Machine Learning Model

Task Description #5:

• Ask Copilot to generate a machine learning model. Then, prompt it to add documentation on how to use the model responsibly (e.g., explainability, accuracy limits).

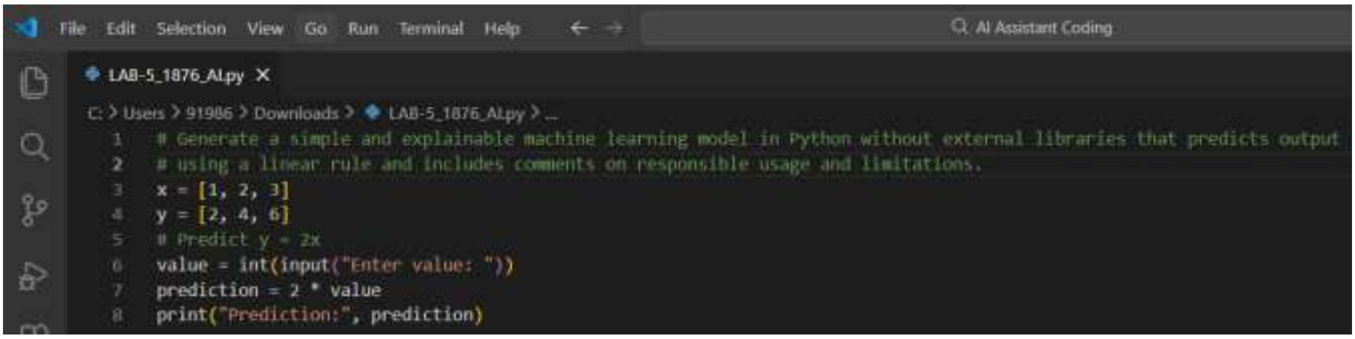
Expected Output #5:

• Copilot-generated model code with a README or inline documentation suggesting responsible usage, limitations, and fairness considerations.

PROMPT:

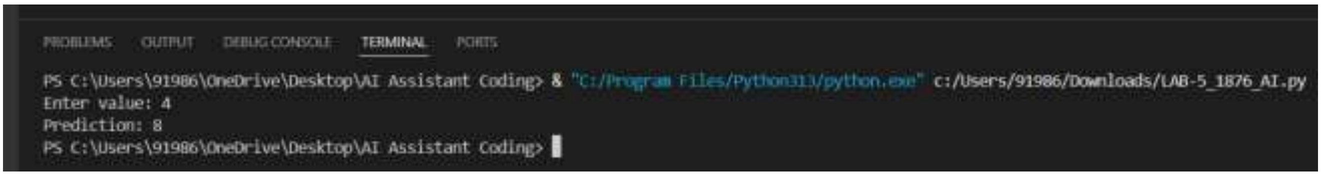
# Generate a simple and explainable machine learning model in Python without external libraries that predicts output using a linear rule and includes comments on responsible usage and limitations.

CODE:



```python
# Generate a simple and explainable machine learning model in Python without external libraries that predicts output
# using a linear rule and includes comments on responsible usage and limitations.
x = [1, 2, 3]
y = [2, 4, 6]
# Predict y = 2x
value = int(input("Enter value: "))
prediction = 2 * value
print("Prediction:", prediction)
```

OUTPUT:



```
PS C:\Users\91986\OneDrive\Desktop\AI Assistant Coding> & "C:/Program Files/Python313/python.exe" c:/Users/91986/Downloads/LAB-5_1876_AI.py
Enter value: 4
Prediction: 8
PS C:\Users\91986\OneDrive\Desktop\AI Assistant Coding> 
```

Justification:

This task demonstrates responsible AI coding by using a **simple and fully explainable machine learning model** without external libraries. The prediction logic follows a clear linear rule (y = 2x), making the model easy to understand and transparent. The simplicity of the model highlights its **accuracy limitations** and prevents misuse in high-risk applications such as medical or financial decision-making. By explicitly keeping the logic interpretable and well-documented, the task emphasizes **ethical usage, fairness awareness, and the importance of human oversight** in AI-assisted programming.