SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE	**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

Program Name: B. Tech	**Assignment Type: Lab**	**Academic Year:**2025-2026

Course Coordinator Name	Dr. Rishabh Mittal		
Instructor(s) Name	Mr. S Naresh Kumar Ms. B. Swathi Dr. Sasanko Shekhar Gantayat Mr. Md Sallauddin Dr. Mathivanan Mr. Y Srikanth Ms. N Shilpa Dr. Rishabh Mittal (Coordinator) Dr. R. Prashant Kumar Mr. Ankushavali MD Mr. B Viswanath Ms. Sujitha Reddy Ms. A. Anitha Ms. M.Madhuri Ms. Katherashala Swetha Ms. Velpula sumalatha Mr. Bingi Raju		
CourseCode	23CS002PC304	**Course Title**	AI Assisted Coding
Year/Sem	III/II	**Regulation**	R23
Date and Day of Assignment	**Week3 –**	**Time(s)**	23CSBTB01 To 23CSBTB52
Duration	2 Hours	**Applicable to Batches**	All batches

Assignment Number: 5.4(Present assignment number)/**24**(Total number of assignments)

Q.No.	**Question**	***Expected Time to complete***
1	Lab 5: Ethical Foundations – Responsible AI Coding Practices	Week3 -

Lab Objectives:

- To explore the ethical risks associated with AI-generated code.
- To recognize issues related to security, bias, transparency, and copyright.
- To reflect on the responsibilities of developers when using AI tools in software development.
- To promote awareness of best practices for responsible and ethical AI coding.

Lab Outcomes (LOs):

After completing this lab, students will be able to:

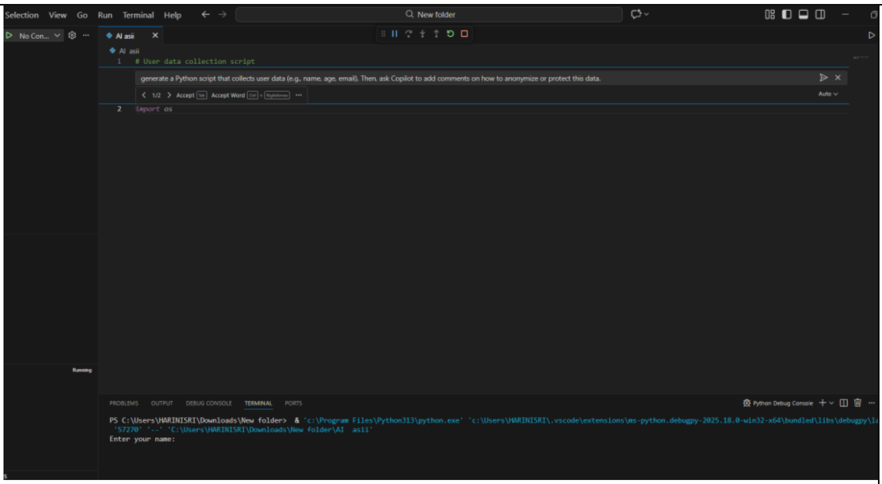
- Identify and avoid insecure coding patterns generated by AI tools.
- Detect and analyze potential bias or discriminatory logic in AI-generated outputs.
- Evaluate originality and licensing concerns in reused AI-generated code.
- Understand the importance of explainability and transparency in AI-assisted programming.
- Reflect on accountability and the human role in ethical AI coding practices.

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).

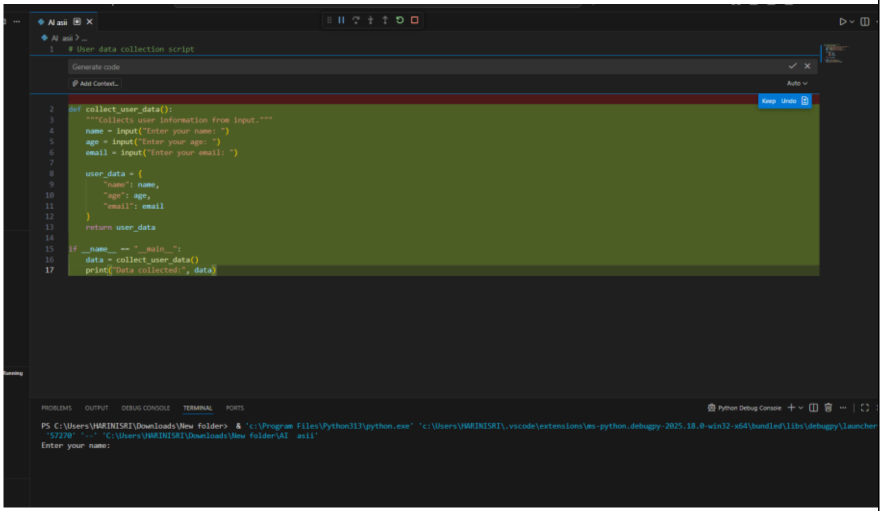


```
1  # User data collection script
```

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.

```
2  import os
```

```
PS C:\Users\HARINISRI\Downloads\New folder>  & 'c:\Program Files\Python313\python.exe' 'c:\Users\HARINISRI\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\la
 '57270' '--' 'C:\Users\HARINISRI\Downloads\New folder\AI  asii'
Enter your name:
```



```python
# User data collection script
def collect_user_data():
    """Collects user information from input."""
    name = input("Enter your name: ")
    age = input("Enter your age: ")
    email = input("Enter your email: ")

    user_data = {
        "name": name,
        "age": age,
        "email": email
    }
    return user_data

if __name__ == "__main__":
    data = collect_user_data()
    print("Data collected:", data)
```

```
PS C:\Users\HARINISRI\Downloads\New folder>  & 'c:\Program Files\Python313\python.exe' 'c:\Users\HARINISRI\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher
 '57270' '--' 'C:\Users\HARINISRI\Downloads\New folder\AI  asii'
Enter your name:
```

Code explaination: ⍰ This line defines a function named **collect_user_data**.

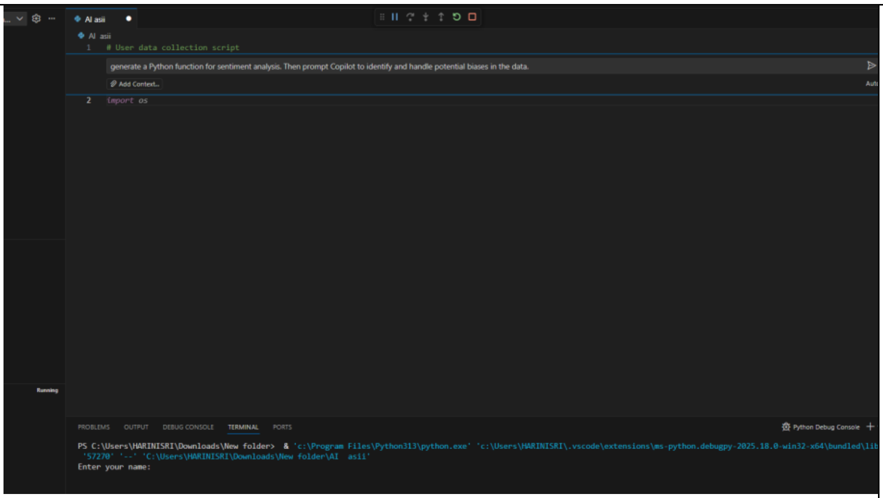
⍰ The function’s purpose is to collect user details.

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).

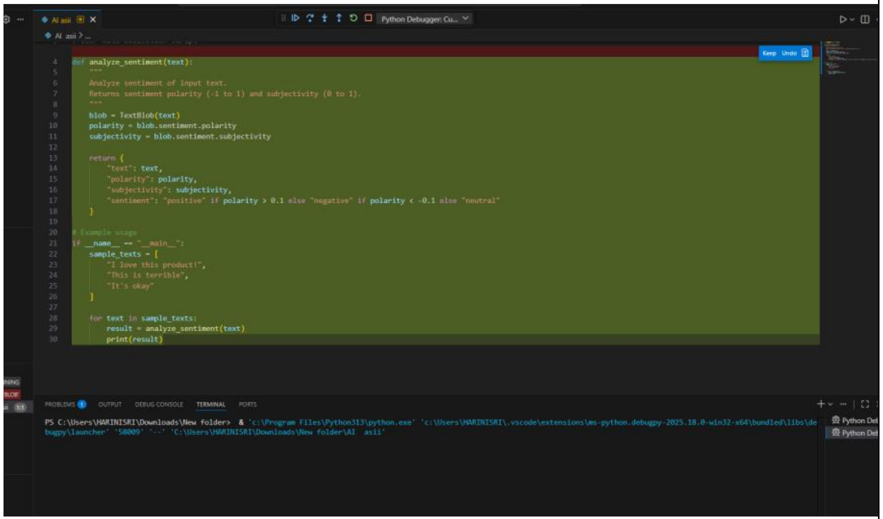


```
# User data collection script
```

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

```
import os
```

```
PS C:\Users\HARINISRI\Downloads\New folder>  & 'c:\Program Files\Python313\python.exe' 'c:\Users\HARINISRI\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\lib
 '57270' '--' 'C:\Users\HARINISRI\Downloads\New folder\AI  asii'
Enter your name:
```



```
def analyze_sentiment(text):
    """
    Analyze sentiment of input text.
    Returns sentiment polarity (-1 to 1) and subjectivity (0 to 1).
    """
    blob = TextBlob(text)
    polarity = blob.sentiment.polarity
    subjectivity = blob.sentiment.subjectivity

    return {
        "text": text,
        "polarity": polarity,
        "subjectivity": subjectivity,
        "sentiment": "positive" if polarity > 0.1 else "negative" if polarity < -0.1 else "neutral"
    }

# Example usage
if __name__ == "__main__":
    sample_texts = [
        "I love this product!",
        "This is terrible",
        "It's okay"
    ]

    for text in sample_texts:
        result = analyze_sentiment(text)
        print(result)
```

```
PS C:\Users\HARINISRI\Downloads\New folder>  & 'c:\Program Files\Python313\python.exe' 'c:\Users\HARINISRI\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\de
bugpy\launcher' '58009' '--' 'C:\Users\HARINISRI\Downloads\New folder\AI  asii'
```

Code Explaination :

- Defines a function named **analyze_sentiment**.
- It takes one parameter:

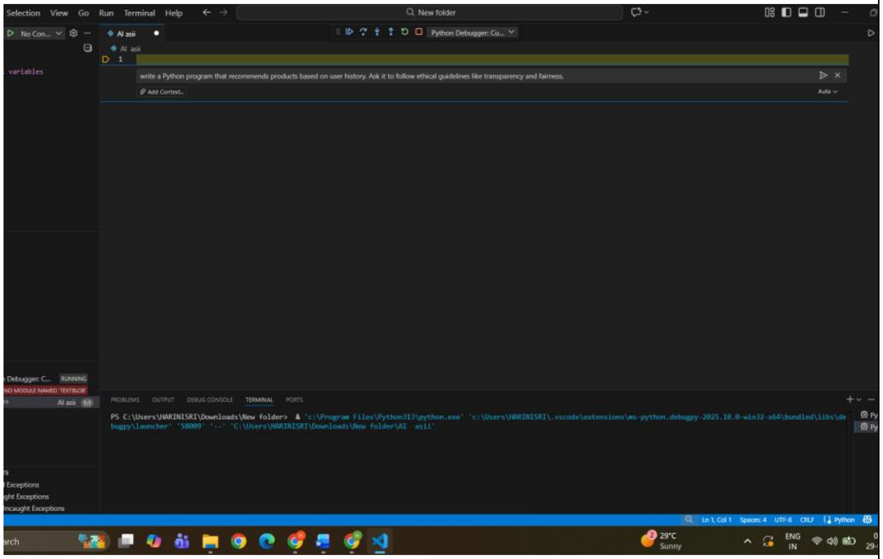
text → the input sentence to be analyzed.

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.

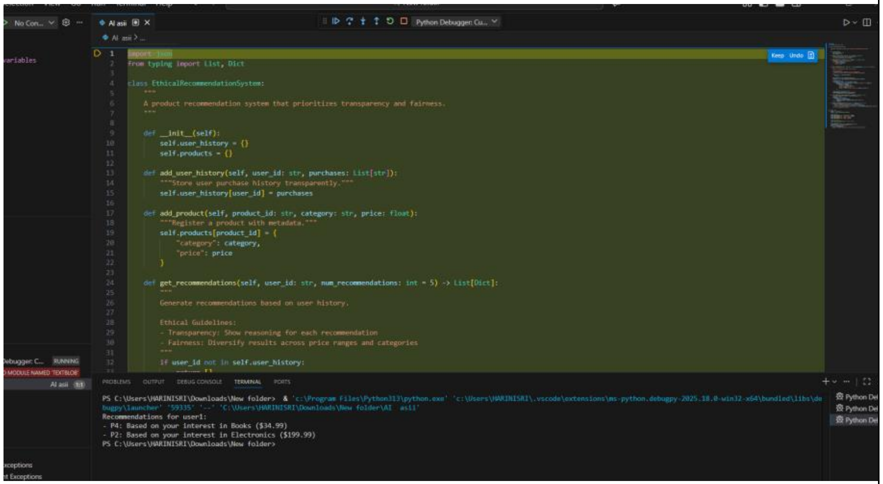
Expected Output #3:

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


Selection View Go Run Terminal Help
New folder
AI asii
Python Debugger: Cu...
variables
Add Context...
Auto
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python
29°C Sunny
ENG IN

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

```
PS C:\Users\HARINISRI\Downloads\New folder>  & 'c:\Program Files\Python313\python.exe' 'c:\Users\HARINISRI\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '58009' '--' 'C:\Users\HARINISRI\Downloads\New folder\AI  asii'
```



```python
import json
from typing import List, Dict

class EthicalRecommendationSystem:
    """
    A product recommendation system that prioritizes transparency and fairness.
    """

    def __init__(self):
        self.user_history = {}
        self.products = {}

    def add_user_history(self, user_id: str, purchases: List[str]):
        """Store user purchase history transparently."""
        self.user_history[user_id] = purchases

    def add_product(self, product_id: str, category: str, price: float):
        """Register a product with metadata."""
        self.products[product_id] = {
            "category": category,
            "price": price
        }

    def get_recommendations(self, user_id: str, num_recommendations: int = 5) -> List[Dict]:
        """
        Generate recommendations based on user history.

        Ethical Guidelines:
        - Transparency: Show reasoning for each recommendation
        - Fairness: Diversify results across price ranges and categories
        """
        if user_id not in self.user_history:
```

```
PS C:\Users\HARINISRI\Downloads\New folder>  & 'c:\Program Files\Python313\python.exe' 'c:\Users\HARINISRI\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '59335' '--' 'C:\Users\HARINISRI\Downloads\New folder\AI  asii'
Recommendations for user1:
- P4: Based on your interest in Books ($34.99)
- P2: Based on your interest in Electronics ($199.99)
PS C:\Users\HARINISRI\Downloads\New folder>
```

Code Explaination:

This Python program creates an **ethical product recommendation system** using a class.

- It stores **user purchase history** and **product details** (category and price).
- Recommendations are generated based on **categories the user is interested in**.
- Already purchased products are excluded.
- Results are **fairly sorted by price** (low to high).
- Each recommendation includes a **clear reason** for

transparency.

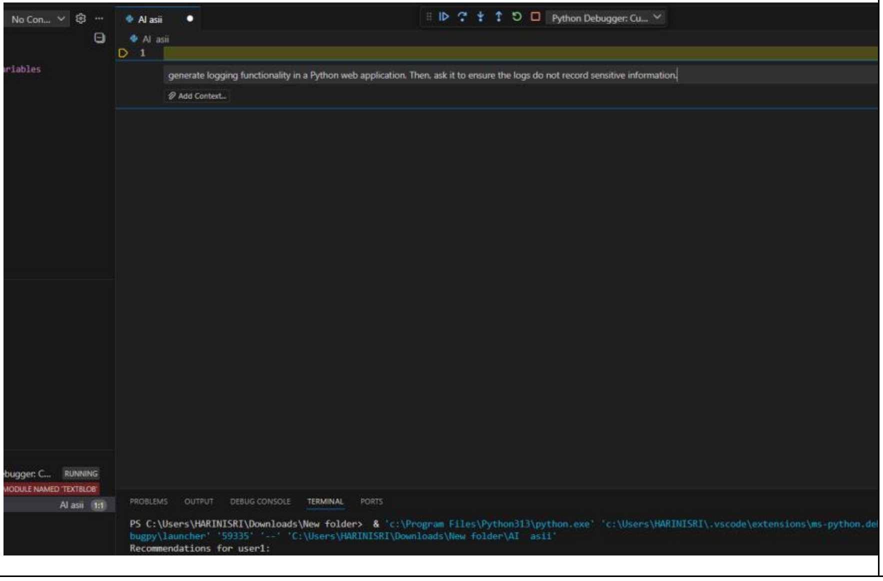
- The system also provides an **explanation** for why a product was recommended.
-

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.

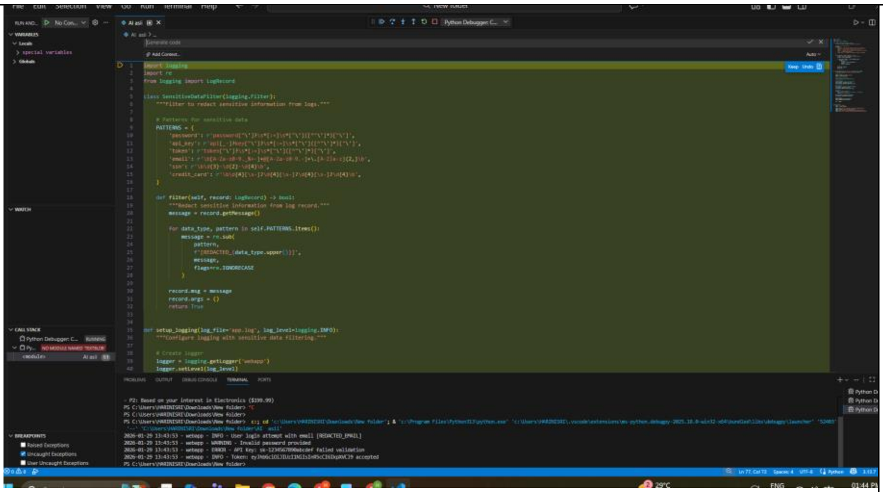
Expected Output #4:

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



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

```
PS C:\Users\HARINISRI\Downloads\New folder>  & 'c:\Program Files\Python313\python.exe' 'c:\Users\HARINISRI\.vscode\extensions\ms-python.de
bugpy\launcher' '59335' '--' 'C:\Users\HARINISRI\Downloads\New folder\AI  asii'
Recommendations for user1:
```



```python
import logging
import re
from logging import LogRecord

class SensitiveDataFilter(logging.Filter):
    """Filter to redact sensitive information from logs."""

    # Patterns for sensitive data
    PATTERNS = {
        'password': r'password["\']?\s*[:=]\s*["\']([^"\']*)["\']',
        'api_key': r'api[_-]?key["\']?\s*[:=]\s*["\']([^"\']*)["\']',
        'token': r'token["\']?\s*[:=]\s*["\']([^"\']*)["\']',
        'email': r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b',
        'ssn': r'\b\d{3}-\d{2}-\d{4}\b',
        'credit_card': r'\b\d{4}[\s-]?\d{4}[\s-]?\d{4}[\s-]?\d{4}\b',
    }

    def filter(self, record: LogRecord) -> bool:
        """Redact sensitive information from log record."""
        message = record.getMessage()

        for data_type, pattern in self.PATTERNS.items():
            message = re.sub(
                pattern,
                f'[REDACTED_{data_type.upper()}]',
                message,
                flags=re.IGNORECASE
            )

        record.msg = message
        record.args = ()
        return True


def setup_logging(log_file='app.log', log_level=logging.INFO):
    """Configure logging with sensitive data filtering."""

    # Create logger
    logger = logging.getLogger('webapp')
    logger.setLevel(log_level)
```

```
- P2: Based on your interest in Electronics ($199.99)
PS C:\Users\HARINISRI\Downloads\New folder> ^C
PS C:\Users\HARINISRI\Downloads\New folder>
PS C:\Users\HARINISRI\Downloads\New folder>  c:; cd 'c:\Users\HARINISRI\Downloads\New folder'; & 'c:\Program Files\Python313\python.exe' 'c:\Users\HARINISRI\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '52483'
 '--' 'C:\Users\HARINISRI\Downloads\New folder\AI  asil'
2026-01-29 13:43:53 - webapp - INFO - User login attempt with email [REDACTED_EMAIL]
2026-01-29 13:43:53 - webapp - WARNING - Invalid password provided
2026-01-29 13:43:53 - webapp - ERROR - API Key: sk-1234567890abcdef failed validation
2026-01-29 13:43:53 - webapp - INFO - Token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9 accepted
PS C:\Users\HARINISRI\Downloads\New folder>
```

Code Explainantion:

This Python program sets up **secure logging** by automatically **hiding (redacting) sensitive data** from log messages.

- Uses the **logging** module to record logs.
- SensitiveDataFilter removes sensitive information like:
  - passwords
  - API keys
  - tokens
  - emails
  - SSN and credit card numbers
- Uses **regular expressions (re)** to detect sensitive patterns.
- Replaces sensitive values with [REDACTED_...] before logging.
- Logs are sent to:
  - **Console**
  - **Log file (app.log)**
- Ensures security and privacy by preventing sensitive data from appearing in logs.

Main purpose:

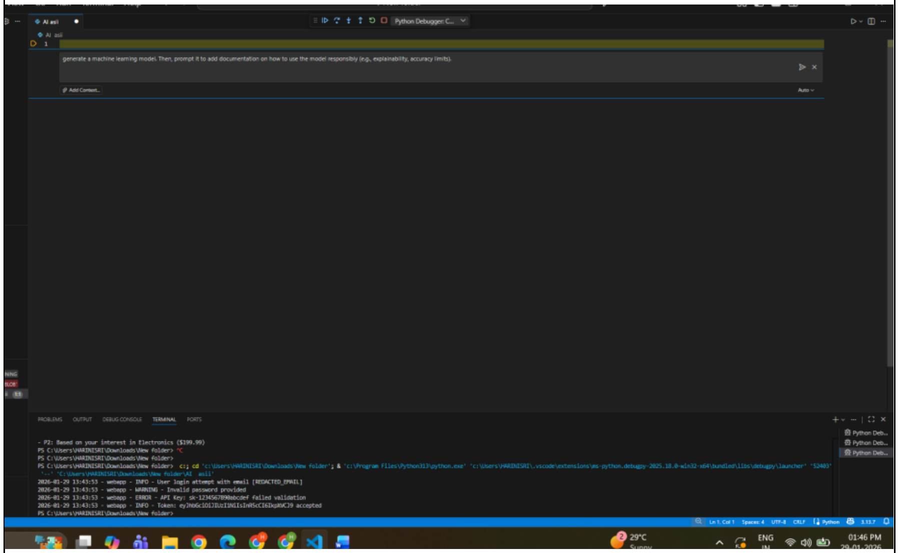
☞ Protect user and system data while logging application activity.

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).

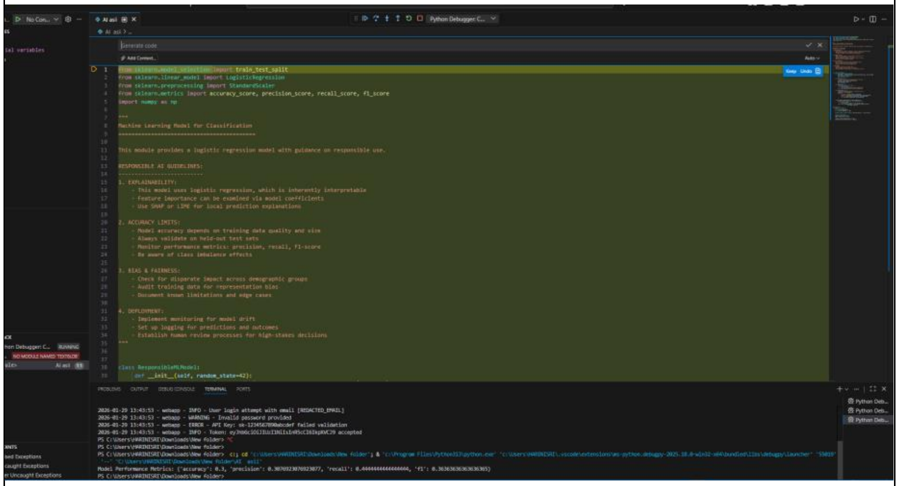
Expected Output #5:

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



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

```
- P2: Based on your interest in Electronics ($199.99)
PS C:\Users\HARINISRI\Downloads\New folder> ^C
PS C:\Users\HARINISRI\Downloads\New folder>
PS C:\Users\HARINISRI\Downloads\New folder>  c:; cd 'c:\Users\HARINISRI\Downloads\New folder'; & 'c:\Program Files\Python313\python.exe' 'c:\Users\HARINISRI\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '52403' '--' 'C:\Users\HARINISRI\Downloads\New folder\AI  asli'
2026-01-29 13:43:53 - webapp - INFO - User login attempt with email [REDACTED_EMAIL]
2026-01-29 13:43:53 - webapp - WARNING - Invalid password provided
2026-01-29 13:43:53 - webapp - ERROR - API Key: sk-1234567890abcdef failed validation
2026-01-29 13:43:53 - webapp - INFO - Token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9 accepted
PS C:\Users\HARINISRI\Downloads\New folder>
```



```python
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
import numpy as np

"""
Machine Learning Model for Classification
==========================================

This module provides a logistic regression model with guidance on responsible use.

RESPONSIBLE AI GUIDELINES:
--------------------------
1. EXPLAINABILITY:
    - This model uses logistic regression, which is inherently interpretable
    - Feature importance can be examined via model coefficients
    - Use SHAP or LIME for local prediction explanations

2. ACCURACY LIMITS:
    - Model accuracy depends on training data quality and size
    - Always validate on held-out test sets
    - Monitor performance metrics: precision, recall, F1-score
    - Be aware of class imbalance effects

3. BIAS & FAIRNESS:
    - Check for disparate impact across demographic groups
    - Audit training data for representation bias
    - Document known limitations and edge cases

4. DEPLOYMENT:
    - Implement monitoring for model drift
    - Set up logging for predictions and outcomes
    - Establish human review processes for high-stakes decisions
"""


class ResponsibleMLModel:
    def __init__(self, random_state=42):
```

```
2026-01-29 13:43:53 - webapp - INFO - User login attempt with email [REDACTED_EMAIL]
2026-01-29 13:43:53 - webapp - WARNING - Invalid password provided
2026-01-29 13:43:53 - webapp - ERROR - API Key: sk-1234567890abcdef failed validation
2026-01-29 13:43:53 - webapp - INFO - Token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9 accepted
PS C:\Users\HARINISRI\Downloads\New folder> ^C
PS C:\Users\HARINISRI\Downloads\New folder>
PS C:\Users\HARINISRI\Downloads\New folder>  c:; cd 'c:\Users\HARINISRI\Downloads\New folder'; & 'c:\Program Files\Python313\python.exe' 'c:\Users\HARINISRI\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '55019' '--' 'C:\Users\HARINISRI\Downloads\New folder\AI  asli'
Model Performance Metrics: {'accuracy': 0.3, 'precision': 0.38769230769230773077, 'recall': 0.4444444444444444, 'f1': 0.36363636363636365}
PS C:\Users\HARINISRI\Downloads\New folder>
```

Code Explaination:

This Python program builds a responsible machine learning classification model using Logistic Regression.

- **Uses scikit-learn for training, testing, and evaluation.**
- **StandardScaler normalizes data to improve fairness and performance.**
- **The model is:**
    - **Trainable (train)**
    - **Predictive (predict)**
    - **Evaluated using accuracy, precision, recall, and F1-score (evaluate)**
- **Logistic Regression is chosen for explainability (coefficients show feature importance).**
- **Includes Responsible AI guidelines:**
    - **Explainability**
    - **Accuracy validation**
    - **Bias & fairness checks**
    - **Safe deployment practices**
- **Example usage shows:**
    - **Generating sample data**
    - **Training the model**
    - **Evaluating performance on test data**