

2303A51739

Lab assignment 5.4

Lab 5: Ethical Foundations – Responsible AI Coding Practices

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.

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The left sidebar displays a file tree under 'EXPLORER' with various files and folders, including 'AI CODING', 'Assignment1(CP).pdf', 'Assignment2-4.pdf', 'assignment 3.4', 'Assignment2.pdf', 'assignment3.4.docx', 'factorial.py', 'jobs.py', 'jobsCP.py', 'lab assignment ...', 'lab assignment33.pdf', 'lab assignment5.4.py', 'lab1 HCP.pdf', 'matrixHCP.py', 'Untitled20.ipynb', and 'week2 HCP.pdf'. The main editor area shows a Python script named 'lab assignment5.4.py'. The script is a user data collection script with privacy protection features. It includes functions for collecting name, age, and email, validating email format, hashing sensitive data, and storing data securely. The code uses Python's standard library modules like re, hashlib, and sys. The script also includes docstrings and comments explaining its purpose and practices. Below the editor, the 'TERMINAL' tab is active, showing command-line output related to the script's execution. The bottom left corner shows the 'OUTLINE' and 'TIMELINE' tabs.

```
# User Data Collection Script with Privacy Protection
import re
from hashlib import sha256

def collect_user_data():
    """
    Collects user information with data protection best practices.
    """
    print("== User Information Form ==\n")

    # Collect name
    name = input("Enter your name: ").strip()

    # Collect age
    while True:
        try:
            age = int(input("Enter your age: "))
            if age < 0 or age > 150:
                print("Please enter a valid age.")
                continue
            break
        except ValueError:
            print("Please enter a valid number.")

    # Collect and validate email
    email = input("Enter your email: ").strip()
    if not is_valid_email(email):
        print("Invalid email format.")
        return None

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

def is_valid_email(email):
    """
    Validates email format before storage.
    """
    pattern = r"^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$"
    return re.match(pattern, email) is not None

def hash_sensitive_data(email):
    """
    ANONYMIZATION: Hash email for storage.
    One-way hashing prevents direct identification while allowing verification.
    """
    return sha256(email.encode()).hexdigest()

def store_data_securely(user_data):
    """
    BEST PRACTICES:
    - Store hashed identifiers, not raw emails
    - Separate personally identifiable info (PII) from analytics
    - Use encryption for storage and transmission
    """
    hashed_email = hash_sensitive_data(user_data["email"])

Zero-shot: This technique struggles with ambiguity in understanding emotions.
One-shot: This technique provides better clarity in emotional interpretation.
Few-shot: This technique achieves the best emotional accuracy by learning from examples.
PS D:\AI Coding> & C:/Users/ANJALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment5.4.py"
== User Information Form ==

Enter your name: anjali
Enter your age: 19
Enter your email: 2303a51924@sru.edu.in

✓ Data collected and processed securely.
PS D:\AI Coding>
```

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a set of navigation icons. The left sidebar has an Explorer view showing a folder named 'AI CODING' containing files like 'add.py', 'AI lab43.py', 'Assignment1(CP).pdf', etc., and a terminal tab showing command-line history.

The main editor area displays a Python script named 'lab assignment5.4.py'. The code defines two functions: `store_data_securely` and `categorize_age`. The `store_data_securely` function takes a dictionary `user_data` and processes it to remove sensitive information and store only necessary data securely. The `categorize_age` function takes an age value and returns a categorical group ('under_18', '18_34', '35_49', or '50_plus'). The script also includes a check for the main module and prints a message if run directly.

The bottom right corner of the editor shows a status bar with file paths and line numbers. The bottom panel features tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, showing the following session:

```
Zero-shot: This technique struggles with ambiguity in understanding emotions.  
One-shot: This technique provides better clarity in emotional interpretation.  
Few-shot: This technique achieves the best emotional accuracy by learning from examples.  
PS D:\AI Coding> & C:/Users/MDALI/AppData/Local/Programs/Python/Python313/python.exe "d:/AI Coding/lab assignment5.4.py"  
== User Information Form ==  
Enter your name: anjali  
Enter your age: 19  
Enter your email: 2303a51924@sru.edu.in  
✓ Data collected and processed securely.  
PS D:\AI Coding>
```

2) 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.

```

File Edit Selection View Go Run Terminal Help < > Q AI Coding
EXPLORER - lab assignment54.py & lab assignments...
    AI CODING
        add.py
        AI lab3.py
        Assignment1CP.pdf
        Assignment2-4.pdf
        assignment 34
        assignment 3.py
        Assignment2.pdf
        assignment34.docx
        factorial.py
        jobs.py
        jobsco.py
        lab assignment ...
        lab1 HCP.pdf
        matrixHCP.py
        Untitled20.pyrb
        week2 HCP.pdf
    JOBS
        jobs.py
        jobsco.py
    LAB ASSIGNMENT ...
        lab assignments...
    DOCUMENTS
        lab assignment54.py
        lab1 HCP.pdf
        matrixHCP.py
        Untitled20.pyrb
        week2 HCP.pdf
    PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
    Data collected and processed securely.
    PS D:\AI Coding & C:\Users\MDALI\Apdata\Local\Programs\Python\Python313\python.exe "d:\AI Coding\lab assignment54.py"
    Text: This product is amazing and wonderful!
    Result: {'score': 0.7, 'label': 'Positive', 'confidence': 'low'}
    Text: I hate this, it's terrible and awful.
    Result: {'score': -0.7, 'label': 'Negative', 'confidence': 'low'}
    Text: It's okay.
    Result: {'score': 0.0, 'label': 'Neutral', 'confidence': 'low'}
    PS D:\AI Coding []

```

```

File Edit Selection View Go Run Terminal Help < > Q AI Coding
EXPLORER - lab assignment54.py & lab assignments...
    AI CODING
        add.py
        AI lab3.py
        Assignment1CP.pdf
        Assignment 2-4.pdf
        assignment 34
        assignment 3.py
        Assignment2.pdf
        assignment34.docx
        factorial.py
        jobs.py
        jobsco.py
        lab assignment ...
        lab1 HCP.pdf
        matrixHCP.py
        Untitled20.pyrb
        week2 HCP.pdf
    JOBS
        jobs.py
        jobsco.py
    LAB ASSIGNMENT ...
        lab assignments...
    DOCUMENTS
        lab assignment54.py
        lab1 HCP.pdf
        matrixHCP.py
        Untitled20.pyrb
        week2 HCP.pdf
    PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
    Data collected and processed securely.
    PS D:\AI Coding & C:\Users\MDALI\Apdata\Local\Programs\Python\Python313\python.exe "d:\AI Coding\lab assignment54.py"
    Text: This product is amazing and wonderful!
    Result: {'score': 0.7, 'label': 'Positive', 'confidence': 'low'}
    Text: I hate this, it's terrible and awful.
    Result: {'score': -0.7, 'label': 'Negative', 'confidence': 'low'}
    Text: It's okay.
    Result: {'score': 0.0, 'label': 'Neutral', 'confidence': 'low'}
    PS D:\AI Coding []

```

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.

The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows files like `assignment 3.py`, `AI lab5.py`, and `lab assignment5.py`.
- Code Editor:** Displays the `lab assignment5.py` file, which contains Python code for an ethical product recommender system.
- Terminal:** Shows the command `AI Coding`.
- Bottom Bar:** Includes tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS.
- Bottom Panel:** Shows recommendations for three products:
 - 1. AI Ethics Guide (ID: book_b)
Price: \$55
Ethical Rating: 0.92/1.0
Why: Similar to your interests in books with ethical rating 0.92
 - 2. Wireless Keyboard (ID: keyboard_a)
Price: \$79
Ethical Rating: 0.85/1.0
Why: Similar to your interests in electronics with ethical rating 0.85
 - 3. Budget Laptop (ID: laptop_b)

The screenshot shows the Microsoft Visual Studio Code (VS Code) interface. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a back/forward navigation bar. The left sidebar (Explorer) shows a tree view of files and folders, including 'AI CODING' and various assignment and lab files. The main editor area displays Python code for an 'EthicalProductRecommender' class. The code uses a database of products to generate recommendations based on user interests (category match) and ethical ratings (ethical score). It sorts the recommendations by score and prints them with transparency information. The bottom status bar shows 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The terminal pane at the bottom shows command-line output for three product recommendations.

```
File Edit Selection View Go Run Terminal Help < > Q AI Coding

EXPLORER Welcome assignment3.py lab assignment54.py
AI CODING assignment54.py
  - add.py
  - AI lab43.py
Assignment 1.pdf
Assignment 2.pdf
Assignment 3.pdf
Assignment4.pdf
assignment3.docx
  - factorial.py
  - jobs.py
  - jobs.py
  - jobs.py
  - lab assignment ...
  - lab assignment3.pdf
  - lab assignments...
  - lab1 HCP.pdf
  - matrixv2.py
  - Untitled0.ipynb
  - week2 HCP.pdf

TERMINAL
AI Coding

1. Wireless Keyboard (ID: keyboard_a)
Price: $59
Ethical Rating: 0.85/1.0
Why: Similar to your interests in books with ethical rating 0.82

2. Razer Laptop (ID: laptop_b)
Price: $499
Ethical Rating: 0.6/1.0
Why: Similar to your interests in electronics with ethical rating 0.85

3. Budget Laptop (ID: laptop_b)
Price: $499
Ethical Rating: 0.6/1.0
Why: Similar to your interests in electronics with ethical rating 0.6

PS D:\VAI Coding [1]
```

The screenshot shows a Python code editor interface with the following details:

- File Explorer:** Shows files like 'assignment3.py', 'lab assignment5.4.py', 'lab assignment3.4.docx', 'lab assignments5.4.pdf', 'lab1 HCP.pdf', 'matrixHCP.py', and 'Untitled20.ipynb'. The 'assignment3.py' file is open.
- Code Editor:** Displays the content of 'assignment3.py'. The code defines a class `EthicalProductRecommender` with a method `print_recommendations_with_transparency`. It also includes a main script block with sample products and user history.
- Terminal:** Shows the execution of the script and its output. The output includes recommendations for user 'user_001' based on their interests in books and electronics.

```
File Edit Selection View Go Run Terminal Help ← → Q AI Coding

EXPLORER
-- Welcome assignment3.py lab assignment5.4.py lab assignment3.4.py
  lab assignment5.4.py ...
    5 class EthicalProductRecommender:
    6     def print_recommendations_with_transparency(self, user_id: str):
    7         print(f"--- Recommendation for User {user_id} ---")
    8         print("Recommendation Criteria: 70% Category Match + 30% Ethical Rating")
    9         print("-" * 68)
   10
   11         for i, rec in enumerate(recommendations):
   12             print(f"({i+1}) {rec['name']} ({ID: {rec['product_id']}})")
   13             print(f"  Price: ${rec['price']}")
   14             print(f"  Ethical Rating: {rec['ethical_rating']}/1.0")
   15             print(f"  Why: {rec['reason']}")
   16
   17 # Example usage
   18 if __name__ == "__main__":
   19     recommender = EthicalProductRecommender()
   20
   21     # Add sample products
   22     products = [
   23         'book_a': {'name': 'Ecological Pro', 'category': 'electronics', 'price': 999, 'ethical_rating': 0.0},
   24         'laptop_b': {'name': 'Budget Laptop', 'category': 'electronics', 'price': 499, 'ethical_rating': 0.6},
   25         'keyboard_d': {'name': 'Wireless Keyboard', 'category': 'electronics', 'price': 79, 'ethical_rating': 0.85},
   26         'book_e': {'name': 'Python Programming', 'category': 'books', 'price': 45, 'ethical_rating': 0.95},
   27         'book_f': {'name': 'AI Ethics Guide', 'category': 'BOOKS', 'price': 55, 'ethical_rating': 0.92},
   28     ]
   29
   30     recommender.add_products(products)
   31
   32     # Add user history
   33     recommender.add_user_history('user_001', ['laptop_a', 'book_e'])
   34
   35     # Get and display recommendations
   36     recommender.print_recommendations_with_transparency('user_001')

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Why: Similar to your interests in books with ethical rating 0.92
2. Wireless Keyboard (ID: keyboard_d)
Price: $79
Ethical Rating: 0.85/1.0
Why: Similar to your interests in electronics with ethical rating 0.85

3. Budget Laptop (ID: laptop_b)
Price: $499
Ethical Rating: 0.6/1.0
Why: Similar to your interests in electronics with ethical rating 0.6
PS D:\AI Coding
```

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

The screenshot shows a Python development environment with the following details:

- File Explorer:** Shows files like `ass 4.4.pdf`, `ass 4.4.py`, `check_lead_year.py`, `collect_user_data_minmax.py`, `collect_user_data.py`, `lab 4.3.py`, `lab 4.3.pdf`, `lab 4.3x.py`, `lab assignment 3.4.pdf`, `lab assignment 3.4x.py`, `lab assignment 1.4.pdf`, `lab assignment 1.4x.py`, `lab assignment 2.4.pdf`, `lab assignment 2.4x.py`, `log.py`, and `README.md`.
- Code Editor:** The main editor window displays `ass 4.4.py` with the following content:

```
import logging
import os
from logging import handlers
from functools import wraps

# Configuring secure logging
class SensitiveDataFilter(logging.Filter):
    """Filter to remove sensitive information from logs."""
    def filter(self, record):
        """Filter log records to remove sensitive data."""
        message = record.getMessage()
        for pattern_name, pattern in self.PATTERNS.items():
            message = re.sub(pattern, r'\1[REDACTED]\2', message)
        record.msg = message
        record.args = ()
        return True

def setup_logger(name, log_file='app.log', level=logging.INFO):
    """
    Set up a secure logger with rotating file handler.

    Args:
        name: Logger name
        log_file: Path to log file
        level: Logging level

    Returns:
        Configured logger instance
    """
    logger = logging.getLogger(name)
    logger.setLevel(level)
    # Add sensitive data filter
    logger.addFilter(SensitiveDataFilter())
    if not os.path.exists(os.path.dirname(log_file)):
        os.makedirs(os.path.dirname(log_file))
    handler = RotatingFileHandler(log_file, maxBytes=1000000, backupCount=5)
    logger.addHandler(handler)

if __name__ == '__main__':
    setup_logger('test', 'test.log')
    logger = logging.getLogger('test')
    logger.info('This is a test message')
    logger.error('This is an error message')
```
- Terminal:** Shows the command `python app.py` being run.
- Output:** Shows the application's output:

```
root@DESKTOP-CJL109:~/Desktop/Ass4# python app.py
2023-08-17 16:28:44,444 INFO test This is a test message
2023-08-17 16:28:44,444 ERROR test This is an error message
```
- Problems:** Shows one error: `raise s.error('Invalid group reference %s' % index, pos)`.
- Right Panel:** Includes sections for "AI Assistant", "Logs", "File History", "Generated code", "Generated files", and "Working...".

```

    # Add sensitive data filter
    sensitive_filter = SensitiveDataFilter()

    # Create rotating file handler
    handler = RotatingFileHandler(
        log_file,
        maxBytes=10485760, # 10MB
        backupCount=5
    )

    # Create console handler
    console_handler = Logging.StreamHandler()

    # Create formatter
    formatter = logging.Formatter(
        '[%(asctime)s] %(name)s - %(levelname)s',
        datefmt='%Y-%m-%d %H:%M:%S'
    )

    handler.setFormatter(formatter)
    console_handler.setFormatter(formatter)

    # Add filter to both handlers
    handler.addFilter(sensitive_filter)
    console_handler.addFilter(sensitive_filter)

    logger.addHandler(handler)
    logger.addHandler(console_handler)

    return logger

def log_request(logger):
    """Decorator to log HTTP requests securely."""
    def decorator(func):
        @wraps(func)
        def wrapper(*args, **kwargs):
            # Log request details (sensitive data will be redacted)
            logger.info("Request to (%s)", func.__name__)
            try:
                result = func(*args, **kwargs)
                logger.info("%s(%s) executed successfully", func.__name__, result)
                return result
            except Exception as e:
                logger.error("%s(%s) - %s", func.__name__, result, str(e))
                raise
            return wrapper
        return wrapper
    return decorator

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

False source("invalid group reference %d % index, pos")
re.PatternError: invalid group reference 1 at position 1
PS C:\Users\parva\Desktop\AI Assisted

22:24 PM 1/29/2026

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).

```

    from sklearn.datasets import load_iris
    from sklearn.model_selection import train_test_split
    from sklearn.neighbors import KNeighborsClassifier
    from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
    import numpy as np

    class IrisClassifier:
        """
        A simple k-Nearest Neighbors classifier for the Iris dataset.

        This class encapsulates the training and prediction logic for classifying
        iris flowers into one of three species based on their measurements.
        """

        def __init__(self, n_neighbors=3):
            """
            Initialize the classifier.

            Args:
                n_neighbors (int): Number of neighbors to consider for prediction. Default is 3.
            """
            self.n_neighbors = n_neighbors
            self.model = KNeighborsClassifier(n_neighbors=n_neighbors)
            self.X_train = None
            self.X_test = None
            self.y_train = None
            self.y_test = None
            self.iris_data = None
            self.target_names = None

        def load_data(self, test_size=0.2, random_state=42):
            """
            Load and split the Iris dataset.

            Args:
                test_size (float): Proportion of data to use for testing. Default is 0.2 (20%).
                random_state (int): Seed for reproducibility. Default is 42.
            """
            self.iris_data = load_iris()
            X, y, test_size=test_size, random_state=random_state
            X_train, self.X_test, self.y_train, self.y_test = train_test_split(X, y, test_size=test_size, random_state=random_state)
            print(f"Data loaded: {len(self.X_train)} training samples, {len(self.X_test)} test samples")

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

raise s.error("invalid group reference %d % index, pos")
re.PatternError: invalid group reference 1 at position 1
PS C:\Users\parva\Desktop\AI Assisted

22:24 PM 1/29/2026

The screenshot shows the VS Code interface with the following details:

- Explorer:** Shows files like `ass 4.kpdf`, `ass 4.pdf`, `check_lear_year.py`, `collect_user_data_minimal.py`, `logger.py`, and `README.md`.
- Editor:** Displays the `ass 4.kpdf` file content, which is a Python script for a k-Nearest Neighbors classifier. It includes imports for `sklearn.datasets`, `sklearn.model_selection`, `sklearn.neighbors`, and `sklearn.metrics`. The class `IrisClassifier` has methods for training, testing, and predicting.
- Terminal:** Shows the command `ps c:users\parwa\onedrive\desktop\ai assisted` and a log message:

```
File: C:\Users\Parwa\OneDrive\Desktop\AI Assisted\ass 4.kpdf
Line: 16
Column: 1
Error: [Errno 2] invalid group reference '$' % index, pos
re.PatternError: invalid group reference '$' at position 1
PS C:\Users\Parwa\OneDrive\Desktop\AI Assisted
```
- Bottom:** Shows the system tray with a weather icon (29°C, Sunny) and the taskbar with various application icons.

The screenshot shows the VS Code interface with the following details:

- Explorer:** Shows files like `ass 4.kpdf`, `ass 4.pdf`, `check_lear_year.py`, `collect_user_data_minimal.py`, `logger.py`, and `README.md`.
- Editor:** Displays the `ass 4.kpdf` file content, which is a Python script for a k-Nearest Neighbors classifier. It includes imports for `sklearn.datasets`, `sklearn.model_selection`, `sklearn.neighbors`, and `sklearn.metrics`. The class `IrisClassifier` has methods for training, testing, and predicting.
- Terminal:** Shows the command `ps c:users\parwa\onedrive\desktop\ai assisted` and a log message:

```
File: C:\Users\Parwa\OneDrive\Desktop\AI Assisted\ass 4.kpdf
Line: 16
Column: 1
Error: [Errno 2] invalid group reference '$' % index, pos
re.PatternError: invalid group reference '$' at position 1
PS C:\Users\Parwa\OneDrive\Desktop\AI Assisted
```
- Bottom:** Shows the system tray with a weather icon (29°C, Sunny) and the taskbar with various application icons.