NAME:CH.KruthanKiran H.NO:2303A51404 BATCH:26

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).

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).

sentiment_with_bias_handling.py > ...

```python
training_data = [
    ("I love this product", "positive"),
    ("This is terrible", "negative"),
    ("Service was good", "positive"),
    ("I hate this", "negative"),
    ("Average experience", "neutral"),

    ("People like you are useless", "negative"),
    ("That group is always bad", "negative")
]


banned_words = ["people like you", "that group", "always bad"]

def clean_dataset(data):
    clean_data = []
    for text, label in data:
        if not any(bad in text.lower() for bad in banned_words):
            clean_data.append((text, label))
    return clean_data

clean_training_data = clean_dataset(training_data)

positive_words = ["love", "good", "great", "excellent"]
negative_words = ["hate", "terrible", "bad", "awful"]

def predict_sentiment(text):
    text = text.lower()
```

PROBLEMS 76 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
Enter your name: Kruthan
Enter your age: 20
Enter your email: Kruthan@gmail.com

User data stored securely (email anonymized).
PS C:\Users\kruth\OneDrive\Desktop\java> ^C
PS C:\Users\kruth\OneDrive\Desktop\java>
PS C:\Users\kruth\OneDrive\Desktop\java>  c:; cd 'c:\Users\kruth\OneDrive\Desktop\java'; &
debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '61259' '--' 'c:\Users\kruth\One
Dataset Balance: {'positive': 2, 'negative': 2, 'neutral': 1}

Sentiment Predictions:
I love the service -> positive
This is awful -> negative
It was okay -> neutral
PS C:\Users\kruth\OneDrive\Desktop\java>
```

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.

ethical_recommendation.py > ...

```python
products = {
    "Laptop": "Electronics",
    "Headphones": "Electronics",
    "Smartphone": "Electronics",
    "Notebook": "Stationery",
    "Pen": "Stationery",
    "Backpack": "Stationery",
    "T-Shirt": "Clothing",
    "Shoes": "Clothing",
    "Jacket": "Clothing"
}
user_history = ["Laptop", "Headphones"]
def recommend_products(history, product_db):
    interested_categories = set()
    for item in history:
        if item in product_db:
            interested_categories.add(product_db[item])
    recommendations = []
    for product, category in product_db.items():
        if category in interested_categories and product not in history:
            recommendations.append(product)
    return recommendations
print("Your Purchase History:", user_history)
recommended = recommend_products(user_history, products)
print("Recommended Products for You:")
for item in recommended:
    print("-", item)
feedback = input("\nDo you like these recommendations? (yes/no): ")
if feedback.lower() == "no":
    print("Thanks for your feedback. We'll improve future suggestions.")
else:
    print("Glad you liked them!")
```

PROBLEMS 76 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
Thank you for your feedback!
PS C:\Users\kruth\OneDrive\Desktop\java> ^C
PS C:\Users\kruth\OneDrive\Desktop\java>
PS C:\Users\kruth\OneDrive\Desktop\java>  c:; cd 'c:\Users\kruth\OneDrive\Desktop\java'; &
debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '51082' '--' 'c:\Users\kruth\On
Your Purchase History: ['Laptop', 'Headphones']
Recommended Products for You:
- Smartphone

Do you like these recommendations? (yes/no): yes
Glad you liked them!
PS C:\Users\kruth\OneDrive\Desktop\java>
```

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.

```python
import logging
import re
class SensitiveDataFilter(logging.Filter):
    def filter(self, record):
        message = str(record.msg)

        email_pattern = r'[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}'
        password_pattern = r'(password|passwd|pwd|secret|token)\s*[:=]\s*["\']?.+?["\']?'

        message = re.sub(email_pattern, "[EMAIL_REDACTED]", message)
        message = re.sub(password_pattern, r"\1: [REDACTED]", message, flags=re.IGNORECASE)

        record.msg = message
        return True

logger = logging.getLogger("SecureApp")
logger.setLevel(logging.INFO)

handler = logging.FileHandler("app_activity.log")
handler.addFilter(SensitiveDataFilter())

formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')
handler.setFormatter(formatter)

logger.addHandler(handler)
def log_user_action(action, user_data):
    log_message = f"Action: {action} | Data: {user_data}"
    logger.info(log_message)
log_user_action("Login Attempt", "user: admin@example.com, password: SuperSecret123")
log_user_action("Profile Update", "Updating email to test@domain.org")
print("Logs saved securely in app_activity.log")
```

```
PROBLEMS 76  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

PS C:\Users\kruth\OneDrive\Desktop\java>  & 'c:\Users\kruth\AppData\Local\Microsoft\WindowsApps\pytho
y\launcher' '49677' '--' 'c:\Users\kruth\OneDrive\Desktop\java\secure_logging_app.py'
Logs saved securely in app_activity.log
PS C:\Users\kruth\OneDrive\Desktop\java>
```

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.

responsibly_ml.py > predict

```python
data = [
    {"age": 25, "score": 70, "label": 1},
    {"age": 30, "score": 85, "label": 1},
    {"age": 22, "score": 40, "label": 0},
    {"age": 45, "score": 90, "label": 1},
    {"age": 35, "score": 50, "label": 0},
]

def train_model(dataset):
    threshold = 60
    return threshold

def predict(model, sample):
    return 1 if sample["score"] >= model else 0
model = train_model(data)

correct = 0
for row in data:
    pred = predict(model, row)
    if pred == row["label"]:
        correct += 1

accuracy = correct / len(data)

print("Model threshold:", model)
print("Accuracy:", accuracy)

print("\nExplainability:")
print("Predictions depend only on 'score' feature.")
print("Age is ignored to avoid potential bias.")
```

PROBLEMS 76 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
ModuleNotFoundError: No module named 'sklearn'
PS C:\Users\kruth\OneDrive\Desktop\java> ^C
PS C:\Users\kruth\OneDrive\Desktop\java>
PS C:\Users\kruth\OneDrive\Desktop\java>  c:; cd 'c:\Users\kruth\OneDrive\Desktop\j
debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '50597' '--' 'c:\Users\k
Model threshold: 60
Accuracy: 1.0

Explainability:
Predictions depend only on 'score' feature.
Age is ignored to avoid potential bias.
PS C:\Users\kruth\OneDrive\Desktop\java>
```

Note: Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots