

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
ProgramName: B. Tech		Assignment Type: Lab	AcademicYear:2025-2026
CourseCoordinatorName		Venkataramana Veeramsetty	
Instructor(s)Name		Dr. V. Venkataramana (Co-ordinator)	
		Dr. T. Sampath Kumar	
		Dr. Pramoda Patro	
		Dr. Brij Kishor Tiwari	
		Dr.J.Ravichander	
		Dr. Mohammand Ali Shaik	
		Dr. Anirodh Kumar	
		Mr. S.Naresh Kumar	
		Dr. RAJESH VELPULA	
		Mr. Kundhan Kumar	
		Ms. Ch.Rajitha	
		Mr. M Prakash	
		Mr. B.Raju	
		Intern 1 (Dharma teja)	
		Intern 2 (Sai Prasad)	
		Intern 3 (Sowmya)	
		NS_2 (Mounika)	
CourseCode	24CS002PC215	CourseTitle	AI Assisted Coding
Year/Sem	II/I	Regulation	R24
Date and Day of Assignment	Week3 - Thursday	Time(s)	
Duration	2 Hours	Applicable to Batches	
AssignmentNumber: 5.4(Present assignment number)/24(Total number of assignments)			
Q.No.	Question	Expected Time to complete	
1	<p>Lab 5: Ethical Foundations – Responsible AI Coding Practices</p> <p>Lab Objectives:</p> <ul style="list-style-type: none"> 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. 	Week3 - Thursday	

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).

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.

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.

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.

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

Evaluation Criteria:

Criteria	Max Marks
How to anonymize or protect this data	0.5

	Identify and handle potential biases in the data.	0.5		
	Follow ethical guidelines like transparency and fairness.	0.5		
	logs do not record sensitive information.	0.5		
	How to use the model responsibly	0.5		
	Total	2.5 Marks		

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 that collects user data (e.g., name, age,email). Then add comments on how to anonymize or protect this data

CODE 1:

```
lab-5.1.py > ...
1  # Python script to collect user data
2
3  def collect_user_data():
4      user_data = {}
5      user_data['name'] = input("Enter your name: ")
6      user_data['age'] = input("Enter your age: ")
7      user_data['email'] = input("Enter your email: ")
8      return user_data
9
10 def main():
11     data = collect_user_data()
12     print("Collected Data:", data)
13
14     # To anonymize or protect this data:
15     # - Do not store raw data in plain text files.
16     # - Hash or encrypt sensitive fields (e.g., use hashlib or cryptography libraries).
17     # - Remove or mask direct identifiers (e.g., replace name/email with pseudonyms or hashes).
18     # - Store data in secure, access-controlled locations.
19     # - Only collect and retain data that is strictly necessary.
20
21 if __name__ == "__main__":
22     main()
23
```

OUTPUT1:

```
PS C:\Users\solle\OneDrive\Desktop\AIAC\lab-05> & C:/Users/solle/AppData/Local/Programs/Python/Python313/python.exe c:/Users/solle/OneDrive/Desktop/AIAC/lab-05/lab-5.1.py
Enter your name: swadha
Enter your age: 18
Enter your email: solletiswadhareddy@gmail.com
Collected Data: {'name': 'swadha', 'age': '18', 'email': 'solletiswadhareddy@gmail.com'}
PS C:\Users\solle\OneDrive\Desktop\AIAC\lab-05>
```

PROMPT2: Generate a Python script that collects user data (e.g., name, age, email). Ensure that the email is stored in a protected form, such as masking it with stars (e.g., j*@gmail.com). Add comments in the code explaining how to anonymize or protect personal data (like removing identifiers, masking, or encrypting sensitive fields)."

CODE2:

```
lab-5.1.1.py > ...
1  def mask_email(email):
2      """
3      Masks the email address for privacy.
4      Example: 'john.doe@gmail.com' -> 'j*****@gmail.com'
5      """
6      try:
7          local, domain = email.split('@')
8          if len(local) > 1:
9              masked_local = local[0] + '*' * (len(local) - 1)
10             else:
11                 masked_local = '*' # If local part is only 1 character
12             return masked_local + '@' + domain
13         except Exception:
14             return '*' * len(email)
15
16 def collect_user_data():
17     user_data = {}
18     user_data['name'] = input("Enter your name: ")
19     user_data['age'] = input("Enter your age: ")
20     email = input("Enter your email: ")
21     user_data['email'] = mask_email(email) # Store masked email only
22     return user_data
23
24 def main():
25     data = collect_user_data()
26     print("Collected Data (with protected email):", data)
27
28     # How to further anonymize or protect personal data:
29     # - Do not store raw data in plain text files.
30     # - Remove or mask direct identifiers (e.g., replace name/email with pseudonyms or hashes).
31     # - Encrypt sensitive fields using secure libraries (e.g., cryptography).
32     # - Store data in secure, access-controlled locations.
33     # - Only collect and retain data that is strictly necessary.
34
35 if __name__ == "__main__":
36     main()
37
```

OUTPUT 2:

```
PS C:\Users\solle\OneDrive\Desktop\AIAC\lab-05> & C:/Users/solle/AppData/Local/Programs/Python/Python313/python.exe c:/Users/solle/OneDrive/Desktop/AIAC/lab-05/lab-5.1.1.py
Enter your name: Swadha reddy
Enter your age: 18
Enter your email: solletiswadhareddy@gmail.com
Enter your age: 18
Enter your email: solletiswadhareddy@gmail.com
Enter your email: solletiswadhareddy@gmail.com
Collected Data (with protected email): {'name': 'Swadha reddy', 'age': '18', 'email': 's*****
*****@gmail.com'}
*****@gmail.com'}
PS C:\Users\solle\OneDrive\Desktop\AIAC\lab-05>
```

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 code for sentiment analysis. Then identify and handle potential biases in the data ,provide an option for input

```
lab-5.2.py > clean_text
1  import re
2
3  # Simple sentiment lexicon
4  positive_words = {"good", "happy", "joy", "excellent", "fortunate", "correct", "superior", "love", "like",
5  negative_words = {"bad", "sad", "pain", "terrible", "unfortunate", "wrong", "inferior", "hate", "dislike",
6
7  # Example of potentially biased words (for demonstration)
8  biased_words = {"always", "never", "everyone", "nobody"}
9
10 def clean_text(text):
11     # Lowercase and remove non-alphabetic characters
12     return re.sub(r'^a-zA-Z\s]', '', text.lower())
13
14 def detect_bias(text):
15     found_biased = [word for word in biased_words if word in text.split()]
16     return found_biased
17
18 def sentiment_analysis(text):
19     text = clean_text(text)
20     words = text.split()
21     pos_count = sum(1 for word in words if word in positive_words)
22     neg_count = sum(1 for word in words if word in negative_words)
23     bias_found = detect_bias(text)
24
25     if bias_found:
26         print(f"Warning: Potentially biased words detected: {'', '.join(bias_found)}")
27         print("Consider revising your input to reduce bias for more accurate sentiment analysis.")
28
29     if pos_count > neg_count:
30         return "Positive"
31     elif neg_count > pos_count:
32         return "Negative"
33     else:
34         return "Neutral"
35
36 def main():
37     print("Sentiment Analysis Tool")
38     user_input = input("Enter a sentence for sentiment analysis: ")
39     result = sentiment_analysis(user_input)
40     print(f"Sentiment: {result}")
41
42 if __name__ == "__main__":
43     main()
44
```

```

PS C:\Users\solle\OneDrive\Desktop\AIAC\lab-05> & C:/Users/solle/AppData/Local/Programs/Python/Python313/python.exe c:/Users/solle/OneDrive/Desktop/AIAC/lab-05/lab-5.2.py
Sentiment Analysis Tool
Enter a sentence for sentiment analysis: i am feeling sad
Sentiment: Negative
PS C:\Users\solle\OneDrive\Desktop\AIAC\lab-05> & C:/Users/solle/AppData/Local/Programs/Python/Python313/python.exe c:/Users/solle/OneDrive/Desktop/AIAC/lab-05/lab-5.2.py
Sentiment Analysis Tool
Enter a sentence for sentiment analysis: i got first rank,hurray!
Sentiment: Neutral
PS C:\Users\solle\OneDrive\Desktop\AIAC\lab-05> & C:/Users/solle/AppData/Local/Programs/Python/Python313/python.exe c:/Users/solle/OneDrive/Desktop/AIAC/lab-05/lab-5.2.py
Sentiment Analysis Tool
Enter a sentence for sentiment analysis: i got a new dress for my birthday
Sentiment: Neutral
PS C:\Users\solle\OneDrive\Desktop\AIAC\lab-05> & C:/Users/solle/AppData/Local/Programs/Python/Python313/python.exe c:/Users/solle/OneDrive/Desktop/AIAC/lab-05/lab-5.2.py
Sentiment Analysis Tool
Enter a sentence for sentiment analysis: i got a new dress for my birthday
Enter a sentence for sentiment analysis: i got a new dress for my birthday
Sentiment: Neutral

```

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

```

1  # Product Recommendation System with Ethical Guidelines
2
3  def collect_user_history():
4      print("To recommend products, we need to collect your product usage or purchase history.")
5      print("We value your privacy and will use your data only for providing recommendations.")
6      history = []
7      while True:
8          product = input("Enter a product you have used or purchased (or type 'done' to finish): ")
9          if product.lower() == 'done':
10             break
11             history.append(product.strip().lower())
12     return history
13
14     def recommend_products(user_history, product_catalog):
15         # Simple recommendation: recommend products from the same category as user's history
16         recommendations = set()
17         for product in user_history:
18             category = product_catalog.get(product)
19             if category:
20                 for prod, cat in product_catalog.items():
21                     if cat == category and prod != product:
22                         recommendations.add(prod)
23         # Remove already used products
24         recommendations = recommendations - set(user_history)
25         return list(recommendations)
26
27     def main():
28         print("Welcome to the Ethical Product Recommendation System!")
29         print("Transparency: We will explain how recommendations are made.")

```

```

30     print("Fairness: Recommendations are based on your input, not on any personal attributes.")
31     print("You can opt out at any time by typing 'done'.\n")
32
33     # Example product catalog: product -> category
34     product_catalog = {
35         "laptop": "electronics",
36         "smartphone": "electronics",
37         "headphones": "electronics",
38         "novel": "books",
39         "cookbook": "books",
40         "running shoes": "sportswear",
41         "yoga mat": "sportswear",
42         "basketball": "sportswear",
43         "blender": "kitchen",
44         "toaster": "kitchen"
45     }
46
47     user_history = collect_user_history()
48     if not user_history:
49         print("No history provided. Unable to make recommendations.")
50         return
51
52     recommendations = recommend_products(user_history, product_catalog)
53     print("\nHow recommendations are made: We look for products in the same categories as those you've used")
54     if recommendations:
55         print("Recommended products for you (based on your history):")
56         for rec in recommendations:
57             print(f"- {rec.title()}")
58     else:
59         print("No new recommendations found based on your history.")
60
61     print("\nThank you for using our system. Your data was used only for this session and not stored.")
62
63 if __name__ == "__main__":
64     main()
65

```

We value your privacy and will use your data only for providing recommendations.

Enter a product you have used or purchased (or type 'done' to finish): laptop

Enter a product you have used or purchased (or type 'done' to finish): novel

Enter a product you have used or purchased (or type 'done' to finish): done

How recommendations are made: We look for products in the same categories as those you've used or purchased.

Recommended products for you (based on your history):

- Headphones
- Cookbook
- Smartphone

Thank you for using our system. Your data was used only for this session and not stored.

PS C:\Users\solle\OneDrive\Desktop\AIAC\lab-05>