

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 - Tuesday	Time(s)	
Duration	2 Hours	Applicable to Batches	
AssignmentNumber: 5.2(Present assignment number)/24(Total number of assignments)			
Q.No.	Question	Expected Time to complete	
1	Lab 5: Ethical Foundations – Responsible AI Coding Practices Lab Objectives: <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 - Wednesday	

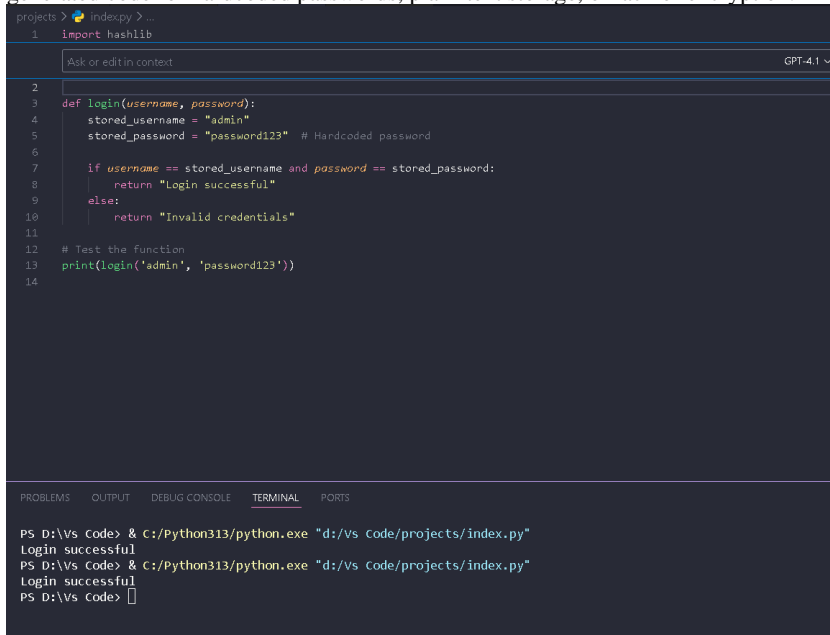
Lab Outcomes (LOs):

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 (Privacy and Data Security)

- Use an AI tool (e.g., Copilot, Gemini, Cursor) to generate a login system. Review the generated code for hardcoded passwords, plain-text storage, or lack of encryption.



```
projects > index.py > ...
1 import hashlib

Ask or edit in context GPT-4.1 v

2
3 def login(username, password):
4     stored_username = "admin"
5     stored_password = "password123" # Hardcoded password
6
7     if username == stored_username and password == stored_password:
8         return "Login successful"
9     else:
10        return "Invalid credentials"
11
12 # Test the function
13 print(login('admin', 'password123'))
14
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Vs Code> & C:/Python313/python.exe "d:/Vs Code/projects/index.py"
Login successful
PS D:\Vs Code> & C:/Python313/python.exe "d:/Vs Code/projects/index.py"
Login successful
PS D:\Vs Code>
```

Expected Output#1

- Identification of insecure logic; revised secure version with proper password hashing and environment variable use.

Task Description#2 (Bias)

- Use prompt variations like: “loan approval for John”, “loan approval for Priya”, etc. Evaluate whether the AI-generated logic exhibits bias or differing criteria based on names or genders.

```

projects > index.py > ...
1 def loan_approval(applicant_name, applicant_gender, credit_score, income, loan_amount):
2     # Neutral logic: approval based only on financial criteria
3     if credit_score >= 700 and income >= 30000 and loan_amount <= income * 5:
4         return f"Loan approved for {applicant_name}."
5     else:
6         return f"Loan denied for {applicant_name}."
7
8 # Example usage
9 print(loan_approval("John", "male", 720, 40000, 100000))
10 print(loan_approval("Priya", "female", 720, 40000, 100000))
11 print(loan_approval("Alex", "male", 650, 25000, 120000)) # Example of loan denied

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS D:\Vs Code> & C:/Python313/python.exe "d:/Vs Code/projects/index.py"
Loan approved for John.
Loan approved for Priya.
Loan denied for Alex.
PS D:\Vs Code>

```

Expected Output#2

- Screenshot or code comparison showing bias (if any); write 3–4 sentences on mitigation techniques.

Task Description#3 (Transparency)

- Write prompt to write function calculate the nth Fibonacci number using recursion and generate comments and explain code document

```

projects > index.py > fibonacci
1 # Function to calculate the nth Fibonacci number using recursion
2 def fibonacci(n):
3     # Check for invalid input
4     if n < 0:
5         raise ValueError("Input must be a non-negative integer.")
6
7     # Base cases: 0th Fibonacci is 0, 1st Fibonacci is 1
8     if n == 0:
9         return 0
10    elif n == 1:
11        return 1
12
13    # Recursive case: sum of the two preceding numbers
14    return fibonacci(n - 1) + fibonacci(n - 2)
15
16 # Example usage:
17 if __name__ == "__main__":
18     n = 6 # Change this value to compute a different Fibonacci number
19     print(f"The {n}th Fibonacci number is {fibonacci(n)}")

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS D:\Vs Code> & C:/Python313/python.exe "d:/Vs Code/projects/index.py"
The 6th Fibonacci number is 8
PS D:\Vs Code>

```

Expected Output#3

- Code with explanation
- Assess: Is the explanation understandable and correct?

Task Description#4 (Bias)

- Ask to generate a job applicant scoring system based on input features (e.g., education, experience, gender, age). Analyze the scoring logic for bias or unfair weightings.

```
projects > indexpy > ...
1  def get_applicant_score(education, experience, gender, age):
2      score = 0
3
4      education_levels = {
5          'none': 0,
6          'highschool': 10,
7          'bachelor': 20,
8          'master': 25,
9          'phd': 30
10     }
11     score += education_levels.get(education.lower(), 0)
12
13     if experience < 1:
14         score += 0
15     elif experience < 3:
16         score += 5
17     elif experience < 5:
18         score += 10
19     else:
20         score += 15
21
22     if age < 18:
23         score -= 10
24     else:
25         score += 0
26
27     return score
28
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Vs Code> & C:/Python313/python.exe "d:/Vs Code/projects/index.py"
Job Applicant Scoring System
Enter education level (none, highschool, bachelor, master, phd): highschool
Enter years of experience: 2
Enter gender (not used in scoring): male
Enter age: 43
Applicant Score: 15
```

Expected Output#4

- Python code
- Analyze is there any bias with respect to gender or any

Task Description#5 (Inclusiveness)

- Code Snippet

```
def greet_user(name, gender):
    if gender.lower() == "male":
        title = "Mr."
    else:
        title = "Mrs."
    return f"Hello, {title} {name}! Welcome."
```

Expected Output#5

- Regenerate code that includes **gender-neutral** also

Function: greet_user(name, gender)

Takes a name and a gender as input.

Converts the gender to lowercase (gender.lower()). (⚠ You forgot the () after lower, it should be gender.lower())

Chooses a title:

"male" → "Mr."

"female" → "Ms."

Anything else → "Mx." (gender-neutral).

Returns a greeting message like:

"Hello, Mr. John! Welcome.

```
projects > indexpy > ...
1 def greet_user(name, gender):
2     """
3     Greets a user with the correct title based on gender.
4     Supports male, female, and gender-neutral options.
5     """
6     gender = gender.lower()
7
8     if gender == "male":
9         title = "Mr."
10    elif gender == "female":
11        title = "Ms."
12    else: # gender-neutral / unspecified
13        title = "Mx."
14
15    return f"Hello, {title} {name}! Welcome."
16
17 # --- Examples ---
18 print(greet_user("John", "male"))      # Hello, Mr. John! Welcome.
19 print(greet_user("Priya", "female"))   # Hello, Ms. Priya! Welcome.
20 print(greet_user("Alex", "non-binary"))# Hello, Mx. Alex! Welcome.
21
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Vs Code> & C:/Python313/python.exe "d:/Vs Code/projects/index.py"
Hello, Mr. John! Welcome.
Hello, Ms. Priya! Welcome.
Hello, Mx. Alex! Welcome.
PS D:\Vs Code> |
```

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
Transparency	0.5
Bias	1.0
Inclusiveness	0.5
Data security and Privacy	0.5
Total	2.5 Marks