

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
P.SRIDHAR 2403A53013 24BTCAICYB01		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	Applicableto Batches	
AssignmentNumber:5.2(Present assignment number)/24(Total number of assignments)			
Q.No.	Question	ExpectedTime 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 	Week3 - Wednesday	

- 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 (Privacy and Data Security)

Prompt:

 write a program in python to generate a student login system and take input user id and password

✦ Sure, I can help with that. Here is a basic Python program for a student login system that takes user ID and password as input.



- 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.

Code:

```
def student_login():
    student_credentials = {
        "student1": "pass123",
        "student2": "securepwd",
        "student3": "mysecret"
    }
    user_id = input("Enter your User ID: ")
    password = input("Enter your password: ")
    if user_id in student_credentials and student_credentials[user_id] == password:
        print(f"Welcome, {user_id}! Login successful.")
    else:
        print("Invalid User ID or password.")
student_login()
```

Explanation:

- **Gather user requirements:** Determine the necessary fields for the login system (e.g., username, password, email).
- **Design the data storage:** Choose a secure method to store user credentials, such as a database, and define the schema.
- **Implement user registration:** Create a mechanism for new users to sign up, including input validation and password hashing.
- **Implement user login:** Develop the login functionality, verifying user credentials against stored data using secure comparison methods.

	<ul style="list-style-type: none">• Address security considerations: Incorporate measures to prevent common vulnerabilities like hardcoded passwords, plain-text storage, and lack of encryption.• Implement session management (optional but recommended): Add a way to maintain user sessions after successful login.• Review and refine the code: Thoroughly review the generated code for security flaws and best practices.• Finish task: Provide the user with the complete and secure login system code, along with explanations of the security measures implemented. <p>Expected Output#1</p> <ul style="list-style-type: none">● Identification of insecure logic; revised secure version with proper password hashing and environment variable use. <pre>Enter your User ID: student1 Enter your password: pass123 Welcome, student1! Login successful.</pre> <p>Task Description#2 (Bias)</p> <p>Prompt:</p> <div><p>A □ 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.</p><p>✦ That's an important area to investigate. I can help you develop a plan to evaluate if the AI model exhibits bias in loan approval based on name or gender. Here is a plan to get started:</p><ul style="list-style-type: none">• Generate prompt variations: Create a set of prompts for loan approval requests, varying only the name and potentially implying gender.</div> <ul style="list-style-type: none">• 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. <p>Code:</p>	
--	--	--

```

import os
import pandas as pd
from openai import OpenAI
api_key = "YOUR_API_KEY"
try:
    client = OpenAI(api_key=api_key)
except Exception as e:
    print(f"Error initializing OpenAI client: {e}")
responses = []
if api_key != "YOUR_API_KEY":
    for prompt in loan_prompts:
        try:
            response = client.chat.completions.create(
                model="gpt-3.5-turbo",
                messages=[{"role": "user", "content": prompt}]
            )
            responses.append({"prompt": prompt, "response": response.choices[0].message.content})
        except Exception as e:
            print(f"Error generating response for prompt: {prompt} - {e}")
            responses.append({"prompt": prompt, "response": f"Error: {e}"})
else:
    print("Please replace 'YOUR_API_KEY' with your actual OpenAI API key to run the API calls.")
    for prompt in loan_prompts:
        responses.append({"prompt": prompt, "response": "Error: OpenAI API key not provided."})

responses_df = pd.DataFrame(responses)
display(responses_df)

```

Explanation:

Here is a Python code snippet for a basic loan approval system that considers factors like income, credit score, and loan amount, without using name or gender.


I have created a basic loan approval system based on income, credit score, and loan amount.


Expected Output#2

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

Task Description#3 (Transparency)

Prompt:

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

 Certainly. Here is a Python function that calculates the nth Fibonacci number using recursion, complete with comments and a docstring explaining the code.

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

Code:

```

def calculate_fibonacci(n):
    if n <= 1:
        return n
    else:
        return calculate_fibonacci(n - 1) + calculate_fibonacci(n - 2)

n_value = 10
fibonacci_number = calculate_fibonacci(n_value)
print(f"The {n_value}th Fibonacci number is: {fibonacci_number}")

```

Explanation:

What's the Goal : The function calculate `fibonacci_recursive(n)` is designed to find th Fibonacci number at a specific position `n` in the sequence.

Checking for Bad Input: The first thing the code does is check if the number `n` you gave it is negative. The Fibonacci sequence usually starts from position 0, so a negative position doesn't make sense. .

The Easy Answers: Fibonacci sequence starts with 0 and 1. The code knows these are the starting points.

Breaking Down the Problem : If `n` is greater than 1, the code needs to figure out the answer. The rule of the Fibonacci sequence is that each number is the sum of the two numbers before it.

Expected Output#3

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

```
➡ The 10th Fibonacci number is: 55
```

Task Description#4 (Bias)

Prompt:

A

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.

✦

Happy to help! I can generate a plan to create a job applicant scoring system based on input features like education, experience, gender, and age, and then analyze it for bias or unfair weightings.

- **Define input features and scoring criteria:** Determine the relevant input features for job applicants and define the initial scoring criteria

- 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.

Code:

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

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