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	**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:** • 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.

Expected Output#1

- Identification of insecure logic; revised secure version with proper password hashing/encrypting 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.

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

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.

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**

TASK-1:

PROMPT: Generate a login system. Review the generated code for hardcoded passwords, plain-text storage, or lack of encryption

Code:

Task-1.py > ...

```python
import hashlib
import getpass

# Simple in-memory user database with hashed passwords
users_db = {}

def hash_password(password):
    return hashlib.sha256(password.encode()).hexdigest()

def register(username, password):
    if username in users_db:
        print("Username already exists.")
        return False
    users_db[username] = hash_password(password)
    print("Registration successful.")
    return True

def login(username, password):
    hashed = hash_password(password)
    if users_db.get(username) == hashed:
        print("Login successful.")
        return True
    else:
        print("Invalid username or password.")
        return False
```

```python
def main():
    while True:
        print("\n1. Register\n2. Login\n3. Exit")
        choice = input("Choose an option: ")
        if choice == '1':
            username = input("Enter username: ")
            password = getpass.getpass("Enter password: ")
            register(username, password)
        elif choice == '2':
            username = input("Enter username: ")
            password = getpass.getpass("Enter password: ")
            login(username, password)
        elif choice == '3':
            break
        else:
            print("Invalid choice.")

if __name__ == "__main__":
    main()
```

Output:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                                   Python

PS C:\Users\Namitha\OneDrive\Desktop\AIAC\Lab-5> & C:/Users/Namitha/AppData/Local/Microsoft/WindowsApps/python3.11.exe
 c:/Users/Namitha/OneDrive/Desktop/AIAC/Lab-5/Task-1.py

1. Register
2. Login
3. Exit
Choose an option: 1
Enter username: 2403A52255@sru.edu.in
Enter password:
Registration successful.

1. Register
2. Login
3. Exit
Choose an option: 2
Enter username: 2403A52255@sru.edu.in
Enter password:
Enter password:
Login successful.

1. Register
2. Login
3. Exit
Choose an option: 3
PS C:\Users\Namitha\OneDrive\Desktop\AIAC\Lab-5>
```

TASK-2:

- PROMPT: Use prompt variations like: “loan approval for John”, “loan approval for Priya”, etc. Evaluate whether the

generated logic exhibits bias or differing criteria based on names or genders.

```python
import random

def ai_loan_approval(applicant_name, income, credit_score):
    # Simulate AI logic (for demonstration purposes)
    # In a real scenario, logic should not depend on name or gender
    # Here, we check if the logic is biased by name
    approval = False
    if income > 50000 and credit_score > 700:
        approval = True
    # Simulate potential bias (for testing)
    if applicant_name.lower() in ['john', 'michael']:
        approval = approval or (income > 40000 and credit_score > 650)
    elif applicant_name.lower() in ['priya', 'anita']:
        approval = approval or (income > 45000 and credit_score > 680)
    # Random element to simulate unpredictability
    approval = approval or (random.random() > 0.95)
    return approval

# Test cases
applicants = [
    {"name": "John", "income": 48000, "credit_score": 660},
    {"name": "Priya", "income": 48000, "credit_score": 660},
    {"name": "Michael", "income": 52000, "credit_score": 710},
    {"name": "Anita", "income": 52000, "credit_score": 710},
]

for applicant in applicants:
    result = ai_loan_approval(applicant["name"], applicant["income"], applicant["credit_score"
    print(f"Loan approval for {applicant['name']}: {'Approved' if result else 'Denied'}")
```

Output:

```
PS C:\Users\Namitha\OneDrive\Desktop\AIAC\Lab-5> & C:/Users/Namitha/AppData/Local/Microsoft/WindowsApps/python3.11.exe
 c:/Users/Namitha/OneDrive/Desktop/AIAC/Lab-5/Task-2.py
Loan approval for John: Approved
Loan approval for Priya: Approved
Loan approval for Michael: Approved
Loan approval for Anita: Approved
PS C:\Users\Namitha\OneDrive\Desktop\AIAC\Lab-5>
```

Observation:there is no gender bias in the code .it approved based on income and credit score.

TASK-3:

PROMPT: write function calculate the nth Fibonacci number using recursion and generate comments and explain code document.

Code:

Task-3.py > fibonacci

```python
def fibonacci(n):
    """Calculate the nth Fibonacci number using recursion.

    Args:
        n (int): The position in the Fibonacci sequence (0-indexed).

    Returns:
        int: The nth Fibonacci number.

    Raises:
        ValueError: If n is negative.
    """
    # The Fibonacci sequence is defined as:
    # F(0) = 0, F(1) = 1
    # F(n) = F(n-1) + F(n-2) for n > 1

    if n < 0:
        raise ValueError("Input must be a non-negative integer.")
    if n == 0:
        return 0  # Base case: F(0) = 0
    elif n == 1:
        return 1  # Base case: F(1) = 1
    else:
        # Recursive case: sum of previous two Fibonacci numbers
        return fibonacci(n - 1) + fibonacci(n - 2)

# Example usage:
print(fibonacci(6))  # Output: 8

"""
Explanation:
------------
This function calculates the nth Fibonacci number using recursion.
It checks for base cases (n == 0 or n == 1) and returns the corresponding value.
For n > 1, it recursively calls itself to compute the sum of the previous two Fibonacci number
A ValueError is raised if the input is negative.
"""
```

Output:

```
PS C:\Users\Namitha\OneDrive\Desktop\AIAC\Lab-5> & C:/Users/Namitha/AppData/Local/Microsoft/WindowsApps/pytho
 c:/Users/Namitha/OneDrive/Desktop/AIAC/Lab-5/Task-3.py
8
PS C:\Users\Namitha\OneDrive\Desktop\AIAC\Lab-5>
```

TASK-4:

PROMPT: 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:

Task-4.py > ...

```python
def get_score(education, experience, gender, age):
    # Scoring logic
    score = 0

    # Education scoring
    edu_scores = {
        'highschool': 10,
        'bachelor': 20,
        'master': 30,
        'phd': 40
    }
    score += edu_scores.get(education.lower(), 0)

    # Experience scoring
    if experience < 1:
        score += 5
    elif experience < 3:
        score += 15
    elif experience < 5:
        score += 25
    else:
        score += 35

    # Gender scoring (should not affect score to avoid bias)
    # score += 0

    # Age scoring (avoid unfair weighting)
    if 18 <= age <= 60:
        score += 10
    else:
        score += 0

    return score
```

```python
def analyze_bias():
    print("Bias Analysis:")
    print("- Education and experience are weighted based on relevance.")
    print("- Gender does not affect score to avoid bias.")
    print("- Age is only checked for reasonable working age, not weighted unfairly.")
    print("- Ensure that all features are relevant to job performance.")

def main():
    print("Job Applicant Scoring System")
    education = input("Enter education level (HighSchool/Bachelor/Master/PhD): ")
    experience = int(input("Enter years of experience: "))
    gender = input("Enter gender: ")
    age = int(input("Enter age: "))

    score = get_score(education, experience, gender, age)
    print(f"Applicant Score: {score}")

    analyze_bias()

if __name__ == "__main__":
    main()
```

Output:

```
Enter education level (HighSchool/Bachelor/Master/PhD): Highschool
Enter years of experience: 5
Enter gender: female
Enter age: 17
Applicant Score: 45
Bias Analysis:
- Education and experience are weighted based on relevance.
- Gender does not affect score to avoid bias.
- Age is only checked for reasonable working age, not weighted unfairly.
- Ensure that all features are relevant to job performance.
- Age is only checked for reasonable working age, not weighted unfairly.
- Ensure that all features are relevant to job performance.
- Ensure that all features are relevant to job performance.
PS C:\Users\Namitha\OneDrive\Desktop\AIAC\Lab-5>
```

Task-5:

Prompt: def greet_user(name, gender):
  if gender.lower() == 'male':
    title="mr."
  else:
    title="mrs"

return f"Hello,{title} {name}! welcome
Regenerate code that includes gender-neutral also.Take input from user

Code:

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

# Take input from user
name = input("Enter your name: ")
gender = input("Enter your gender (male/female/other): ")

print(greet_user(name, gender))
```

Output:

```
Enter your name: lucky
Enter your gender (male/female/other): male
Hello, Mr. lucky! Welcome
PS C:\Users\Namitha\OneDrive\Desktop\AIAC\Lab-5> & C:/Users/Namitha/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Namitha/OneDrive/Desktop/AIAC/Lab-5/Task-5.py
Enter your name: nammu
Enter your gender (male/female/other): female
Hello, Mrs. nammu! Welcome
PS C:\Users\Namitha\OneDrive\Desktop\AIAC\Lab-5> & C:/Users/Namitha/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/Namitha/OneDrive/Desktop/AIAC/Lab-5/Task-5.py
Enter your name: benny
Enter your gender (male/female/other): other
Hello, Mx. benny! Welcome
PS C:\Users\Namitha\OneDrive\Desktop\AIAC\Lab-5>
```