2403A53011
THATIKONDA NIKHIL
24BTCAICYB01-2-1

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.	Week3 - Wednesday

- 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.

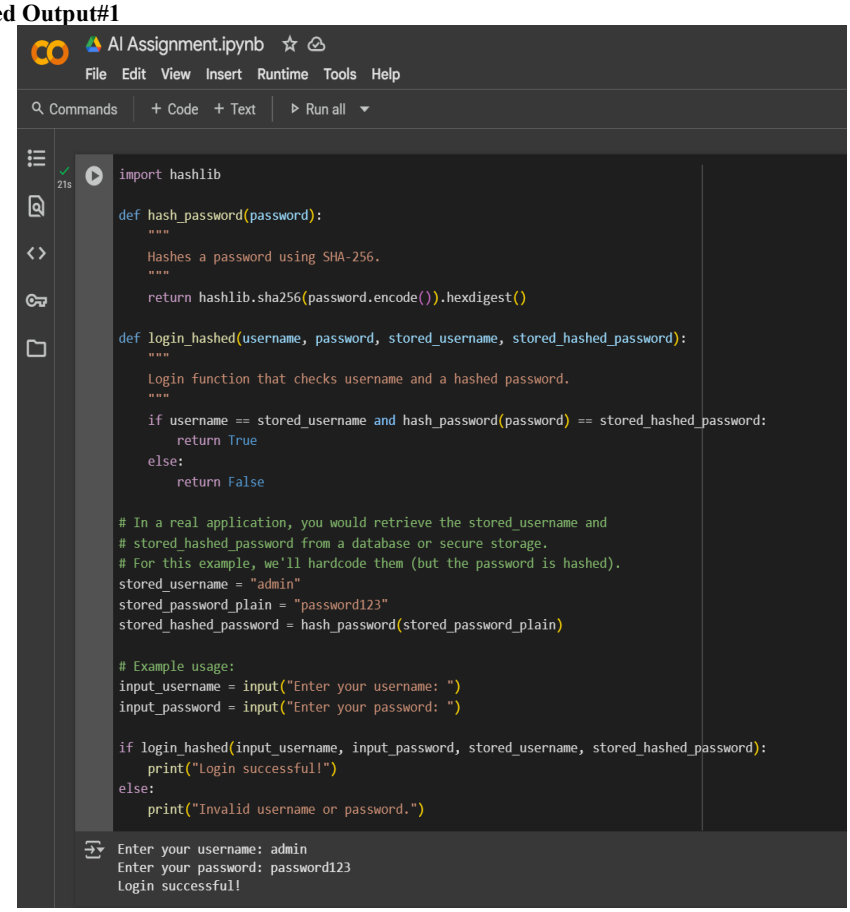
Lab Outcomes (LOs):
After completing this lab, students will be able to:

- 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 (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



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

Provide feedback

Subtask:

Output whether the loan is approved or denied and explain the reason.

Reasoning: Call the functions to get user input, evaluate the application, and then print the decision and reasons for denial.

```python
# 1. Call the get_loan_application_input() function to collect the user's loan application details and store the result in a variable.
applicant_details = get_loan_application_input()

# 2. Call the evaluate_loan_application() function, passing the collected applicant details and the loan_criteria dictionary. Store the returned decision and evaluatio
decision, evaluation_summary = evaluate_loan_application(applicant_details, loan_criteria)

# 3. Print a clear message indicating whether the loan is "APPROVED" or "DENIED" based on the decision variable.
print(f"\nLoan Application Decision: {decision.upper()}")

# 4. If the loan is denied, iterate through the evaluation_summary dictionary. For each criterion that was not met, print the criterion name and the reason provided in
if decision == "denied":
    print("\nReasons for Denial:")
    for criterion, result in evaluation_summary.items():
        if not result["met"]:
            print(f"- {criterion.replace('_', ' ').title()}: {result['reason']}")
```

```
Please provide your loan application details:
Enter your credit score (e.g., 700): 700
Enter your annual income (e.g., 50000): 50000
Enter the requested loan amount (e.g., 10000): 20000
Enter your employment status (employed, self-employed, unemployed): employed

Loan Application Decision: APPROVED
```

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

Commands + Code + Text Run all

```python
def recursive_fibonacci(n):
  """
  Calculates the nth Fibonacci number using recursion.

  The Fibonacci sequence is a series of numbers where each number is the sum
  of the two preceding ones, usually starting with 0 and 1.
  (0, 1, 1, 2, 3, 5, 8, ...)

  Args:
    n: An integer representing the position in the Fibonacci sequence
       (starting from 0).

  Returns:
    The nth Fibonacci number.

  Raises:
    ValueError: If n is a negative integer.
  """
  # Base case: If n is 0 or 1, return n directly.
  # These are the starting points of the sequence.
  if n <= 1:
    return n
  # Recursive step: For n > 1, the nth Fibonacci number is the sum of the
  # (n-1)th and (n-2)th Fibonacci numbers.
  else:
    return recursive_fibonacci(n - 1) + recursive_fibonacci(n - 2)

# Example usage:
# print(recursive_fibonacci(6)) # Output: 8
```

```python
[9] # Example usage of the recursive_fibonacci function
    n = 10
    fib_number = recursive_fibonacci(n)
    print(f"The {n}th Fibonacci number is: {fib_number}")
```

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

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
def score_applicant(applicant):
    score = 0

    # Education Score
    education_weights = {
        'High School': 10,
        'Bachelor's': 20,
        'Master's': 25,
        'PhD': 30
    }
    score += education_weights.get(applicant['education'], 0)

    # Experience Score
    score += min(applicant['years_experience'], 20) * 1.5  # Cap at 20 years

    # Certifications
    if applicant['certified']:
        score += 10

    # Skills Match
    score += min(applicant['skills_match'], 100) * 0.2  # up to 20 points

    # interview Score
    score += applicant['interview_score'] * 3  # up to 30 points

    # Age and Gender - DO NOT include in scoring
    # These are collected for reporting or compliance but should NOT impact the score
    return score
```

```python
# Example usage of the score_applicant function
example_applicant_data = {
    'education': 'Bachelor's',
    'years_experience': 5,
    'certified': True,
    'skills_match': 80,
    'interview_score': 9,
    'gender': 'female',
    'age': 30
}

applicant_score = score_applicant(example_applicant_data)
print(f"The calculated score for the example applicant is: {applicant_score:.2f}")
```

```
The calculated score for the example applicant is: 80.50
```

Task Description#5 (Inclusiveness)

- Code Snippet

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

Expected Output#5

```python
def greet_user(name, gender):
    gender = gender.lower()

    if gender == "male":
        title = "Mr."
    elif gender == "female":
        title = "Mrs."
    else:
        title = "Mx."  # Gender-neutral title

    return f"Hello, {title} {name}! Welcome."
```

```python
# Example usage of the greet_user function
greeting = greet_user("Alice", "Female")
print(greeting)

greeting_male = greet_user("Bob", "Male")
print(greeting_male)

greeting_neutral = greet_user("Charlie", "Non-binary")
print(greeting_neutral)
```

```
Hello, Mrs. Alice! Welcome.
Hello, Mr. Bob! Welcome.
Hello, Mx. Charlie! Welcome.
```

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**