|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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.   **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.   **Promt:**  write a program in python to generate a student login system and take input user id and password  **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. * **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.   **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.   **Promt:**  write a python code to generate a loan approval system without using name or gender as factors  **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)**   * Write prompt to write function calculate the nth Fibonacci number using recursion and generate comments and explain code document   **Promt:**  Write a function that calculates the nth Fibonacci number using recursion and generate comments and explain the code in a understandable way  **Comments:**  The **comments** (lines starting with #) are like little notes in the code to explain what's happening in that specific part. The **docstring** (the text in triple quotes """...""") is a more detailed explanation at the beginning of the function, telling you what the function does, what kind of input it expects (Args), what it gives back (Returns), and what kind of errors might happen (Raises).  **Explanation:**   1. **What's the Goal?**   The function calculate\_fibonacci\_recursive(n) is designed to find the Fibonacci number at a specific position n in the sequence.   1. **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. If it's negative, the code stops and tells you there's an error (raise ValueError).   1. **The Easy Answers (Base Cases):**   Fibonacci sequence starts with 0 and 1. The code knows these are the starting points.   * + If you ask for the number at position 0 (n == 0), it just says "0".   + If you ask for the number at position 1 (n == 1), it just says "1". These are called "base cases" because they are the simplest scenarios that the recursion can directly answer without calling itself again.  1. **Breaking Down the Problem (Recursive Step):** 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. So, to find the number at position n, the code asks:    * "What's the Fibonacci number at position n-1?"    * "What's the Fibonacci number at position n-2?"   It does this by calling itself (calculate\_fibonacci\_recursive) with n-1 and n-2. This is the "recursive" part – the function is calling itself to solve smaller versions of the same problem.   1. **Putting It Together:**   Once it gets the answers for n-1 and n-2, it adds them together (+). This sum is the Fibonacci number at position n.  **Expected Output#3**      **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.     **Promt:**  write a python function to score job applications based on input education, years of experience, gender  **Explanation:**  **Expected Output#4**      **Task Description#5 (Inclusiveness)**   * Code Snippet     **Promt:**  generate a code for gender with gender-neutral also  **Expected Output#5**    **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** | | | | | | | Week3 - Wednesday |  |