|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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** | | | | 1. Dr. Mohammed Ali Shaik  2. Dr. T Sampath Kumar  3. Mr. S Naresh Kumar  4. Dr. V. Rajesh  5. Dr. Brij Kishore  6. Dr Pramoda Patro  7. Dr. Venkataramana  8. Dr. Ravi Chander  9. Dr. Jagjeeth Singh | | | | | |
| **CourseCode** | | | 24CS002PC215 | **CourseTitle** | | AI Assisted Coding | | | |
| **Year/Sem** | | | II/I | **Regulation** | | R24 | | | |
| **Date and Day**  **of Assignment** | | |  | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicableto**  **Batches** | |  | | | |
| **AssignmentNumber:3.3**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***ExpectedTime***  ***to complete*** |  |
|  | 1 | Lab 3: Prompt Engineering – Improving Prompts and Context Management  **Lab Objectives:**   * To understand how prompt structure and wording influence AI-generated code. * To explore how context (like comments and function names) helps AI generate relevant output. * To evaluate the quality and accuracy of code based on prompt clarity. * To develop effective prompting strategies for AI-assisted programming.   **Lab Outcomes (LOs):**  After completing this lab, students will be able to:   * Generate Python code using Google Gemini in Google Colab. * Analyze the effectiveness of code explanations and suggestions by Gemini. * Set up and use Cursor AI for AI-powered coding assistance. * Evaluate and refactor code using Cursor AI features. * Compare AI tool behavior and code quality across different platforms.   **Task Description#1**   * Try 3 different prompts to generate a factorial function.   **Expected Output#1**   * Comparison of AI-generated code styles   **Prompt 1: develop a python code to generate factorial function**    **Prompt 2: write a python function to give a factorial of a given number**    **Prompt 3 : *Write a simple recursive Python function to calculate the factorial of a number***    **Compare:**  implementations for calculating the factorial of a non-negative integer, but they differ in style and behavior. Both factorial and simple\_factorial use an iterative approach, making them efficient, safe from recursion limits, and suitable for large inputs. They are practically identical in logic, with only minor differences in naming and documentation. On the other hand, factorial\_recursive uses a recursive method, which is more elegant and concise, making it useful for demonstrating the concept of recursion. However, it is less efficient for large inputs and can lead to a RecursionError if the input is too high due to Python’s recursion depth limit. Overall, the iterative versions are more robust for practical use, while the recursive version is better suited for learning or use with small input values.  **Task Description#2**   * Provide a clear example input-output prompt to generate a sorting function.   **Expected Output#2**   * Functional sorting code from AI     **Prompt : develop a python program to generate a sorting function**    **Task Description#3**   * Start with the vague prompt “Generate python code to calculate power bill” and improve it step-by-step   **Expected Output#3**   * Enhanced AI output with clearer prompts   **Prompt 1 : Generate Python code to calculate power bill**    **Prompt 2 : Generate Python code that takes electricity units consumed and rate per unit to calculate total power bill**    **Task Description#4**   * Write structured comments to help AI generate two linked functions (e.g., login\_user() and register\_user()).   **Expected Output#4**   * Consistent functions with shared logic   **Prompt 1 :** write a python code to generate linked function login\_user()      **Prompt 2 : write a python code to generate linked function register\_user()**      **Task Description#5**   * Analyzing Prompt Specificity: Improving Temperature Conversion Function with Clear Instructions   **Expected Output#5**   * Code quality difference analysis for various prompts   **Prompt 1 : Write a Python function to convert temperature**      **Promp 2 : Create a Python function that converts between Celsius and Fahrenheit. The function should accept the temperature value and a string indicating the direction ('C' to 'F' or 'F' to 'C'). Include error handling for invalid input.**    **Difference :**   1. *convert\_temp() supports only Celsius to Fahrenheit and vice versa, while convert\_temperature() allows conversion between Celsius, Fahrenheit, and Kelvin in any direction.* 2. *convert\_temp() uses a single string like "C to F" to determine the direction, whereas convert\_temperature() takes separate from\_unit and to\_unit arguments, making it more structured and scalable.* 3. *convert\_temperature() follows a two-step conversion approach (via Celsius), making it easier to expand, while convert\_temp() uses direct formulas with limited flexibility.* 4. *convert\_temp() is simpler and best for basic use cases, but convert\_temperature() is more versatile and better suited for real-world applications.* 5. *convert\_temperature() has clearer separation of concerns and better unit validation, making the code more maintainable and extensible compared to convert\_temp().*   **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** | | --- | --- | | Factorial Function (Task#1) | 0.5 | | Sorting Function (Task#2) | 0.5 | | Vogue Vs. Specific Prompting (Task #3) | 0.5 | | Linked Functions (Task #4) | 0.5 | | Temperature Conversion Function (Task #5) | 0.5 | | **Total** | **2.5 Marks** | | | | | | | 03.08.2025 EOD |  |