|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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: Write a python function to generate a simple factorial function by entering input from user.**    **Expected output:**    **PROMPT 2:** Develop a python function and calculate factorial of a given number from the user.      **Expected output:**    **Prompt 3:** Write python function to calculate factorial.      **Expected output:**    **REPORT:**  **Task 1.py - Simple Approach**  **Key Characteristics:**   * **Error Handling**: Returns error message as string instead of raising exception * **Single Function**: Only has factorial() function * **Basic Structure**: Simple, straightforward implementation * **Return Type**: Can return either int or str (error message)   **Task 1\_2.py - Robust User Input**  **Key Characteristics:**   * **Exception Handling: Uses raise ValueError() for errors** * **Separate Input Function: Has dedicated get\_user\_input() function with validation loop** * **Better UX: Continuously prompts until valid input is received** * **Mathematical Notation: Shows both plain text and mathematical notation (n!)**   **Task 1\_3.py - Comprehensive Testing**  **Key Characteristics:**   * **Multiple Approaches: Includes both iterative AND recursive factorial functions** * **Extensive Testing: Tests multiple numbers automatically** * **Comparison Display: Shows results from both approaches side-by-side** * **Educational: Demonstrates different algorithmic approaches**   **Task Description#2**   * Provide a clear example input-output prompt to generate a sorting function.   **Expected Output#2**   * Functional sorting code from AI   **PROMPT:** Create a python sorting function which give clear input and output.        **Expected output:**    **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(basic):** write a python code to calculate powerbill.    **Expected output:**    **PROMPT (Modified):** Develop a python to calculate power bill for different industries based on units.          **Expected output:**    **Report:**  In basic prompt if we give units it directly calculates the powerbill.  As in modified prompt it specifies in such a way given below.,  **I have updated prompt in such a way to generate the power bill for multiple usages like residence , schools , industires etc..**  **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:**  **Write two Python functions, register\_user() and login\_user(), that use a dictionary users\_db to store usernames and passwords.**          **Expected output:**    **Task Description#5**   * Analyzing Prompt Specificity: Improving Temperature Conversion Function with Clear Instructions   **Expected Output#5**   * Code quality difference analysis for various prompts   Prompt(basic):  **Expected output:**    **PROMPT (modified):** Develop a python function converting temperature conversion from different units.        **Expected output:**    **Report:**  **-From basic prompt it directly takes input by itself and convert temperatures from one form to another form.**  **-Where as in modified prompt it specifies that we need to select from which state to change from and it gives resultant output.**  **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 |  |