NAME:Thokala Srivalli

ENROLLEMENT NO:2403a510b2

BATCH:04

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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** | | | Week2-Tuesday | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicableto**  **Batches** | | 24CSBTB01 To 24CSBTB39 | | | |
| **AssignmentNumber:3.2**(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**   * Ask AI to write a function to calculate compound interest, starting with only the function name. Then add a docstring, then input-output example   **PROMPT:**  #write a python code of a function to calulate  compound interest starting with the only function name then add a docstring and input output example.    **Expected Output#1**   * Comparison of AI-generated code styles     **OBSERVATION:**  The program successfully computes compound interest using the given formula. For the input values (Principal ₹1000, Rate 5%, Time 10 years, Compounded quarterly), the final amount is ₹1648.72 and the interest earned is ₹648.72. This shows how compound interest grows an investment over time by reinvesting the earned interest.  **Task Description#2**   * Do math stuff, then refine it to: # Write a function to calculate average, median, and mode of a list of numbers.   **PROMPT:**  #write a python code of  a function to calculate average, median, and mode of a list of numbers.    **Expected Output#2**   * AI-generated function evolves from unclear to accurate multi-statistical operation     .  **OBSERVATION:**  The program correctly calculates the average, median, and mode of a given list of numbers. For example, with the list [1, 2, 2, 3, 4], it outputs Average = 2.4, Median = 2, and Mode = 2, showing accurate statistical analysis.  **Task Description#3**   * Provide multiple examples of input-output to the AI for convert\_to\_binary(num) function. Observe how AI uses few-shot prompting to generalize.   **PROMPT:**  #Give some numbers to the convert\_to\_binary(num) function and check their binary outputs. From these examples, notice how the function learns the pattern and can convert any new number into binary."    **Expected Output#3**   * Enhanced AI output with clearer prompts       **OBSERVATION:**  The program correctly converts decimal numbers into their binary form. By testing with different inputs like 2, 5, 10, and 32, the function produced outputs such as 10, 101, 1010, and 100000. This shows that the function follows a consistent pattern for conversion and can handle any non-negative integer input.  **Task Description#4**   * Create an user interface for an hotel to generate bill based on customer requirements   **Expected Output#4**   * Consistent functions with shared logic   **PROMPT**: Design a Python-based user interface for a hotel management system that generates customer bills according to their requirements.  **Screenshot 2025-08-19 105149**  **Screenshot 2025-08-19 105207**  **OBSERVATION:**  The implementation provides a simple hotel billing system using Python and Gradio, making the process interactive and user-friendly. It takes inputs such as customer name, room type, number of nights, and optional services, and then generates the final bill accordingly. The interface ensures clarity in data entry while the backend function handles accurate calculation of costs. Overall, the system effectively demonstrates the use of Python for practical billing automation.  **Task Description#5**   * Analyzing Prompt Specificity: Improving Temperature Conversion Function with Clear Instructions   **Expected Output#5**   * Code quality difference analysis for various prompts   **PROMPT:**  Develop a Python function to convert temperatures between Celsius and Fahrenheit. Ensure input validation, clear variable naming, and demonstrate its usage with examples.    **OBSERVATION:**  The program successfully implements a Python function to convert temperatures between Celsius and Fahrenheit with proper input validation. It handles invalid units and temperatures below absolute zero using exceptions, ensuring robust error handling. The use of meaningful variable names improves readability, and example test cases demonstrate both valid and invalid conversions. Overall, the implementation is efficient, user-friendly, and demonstrates practical application of Python functions with error control.    **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** | | --- | --- | | Task#1 | 0.5 | | Task#2 | 0.5 | | Task #3 | 0.5 | | Task #4 | 0.5 | | Task #5 | 0.5 | | **Total** | **2.5 Marks** | | | | | | | 03.08.2025 EOD |  |