|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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#1:**  **- Write a python code to create a function to calculate the compound interest starting with only function name**  **- Now, add a docstring to the created function**  **-Then add an input-output example**  **Expected Output#1**   * Comparison of AI-generated code styles     **CODE#1:**  Screenshot 2025-08-19 at 10.06.36  **OBSERVATION#1:**  **- The code correctly implements the compound interest formula**  **- the docstring is well written, it explains the arguments and return values as well**  **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#2:**  **- Write a python function to calculate average, median and mode of a list of numbers given by the user**    **Expected Output#2**   * AI-generated function evolves from unclear to accurate multi-statistical operation.   **CODE#2:**  **Screenshot 2025-08-19 at 10.30.56**  **OBSERVATION#2:**  **- The average and median are both equal to the middle number**  **- the mode includes all values since no one number is repeating**  **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#3:**  **Generate a python code which could take convert numbers to binary**  **-**  **CODE:**  **Screenshot 2025-09-02 at 05.16.34**  **Expected Output#3**   * Enhanced AI output with clearer prompts   **OBSERVATION:**  It assume the input is always a **positive integer**  **Task Description#4**   * Create an user interface for an hotel to generate bill based on customer requirements   PROMPT:  Write a python code to set up a user interface for a hotel to generate bill based on customer requirements  CODE:  Screenshot 2025-09-02 at 05.23.17  **Expected Output#4**   * Consistent functions with shared logic   **OBSERVATION:**  **We observe that after taking in the customer details, it calls down each defined function to generate the bill**  **Task Description#5**   * Analyzing Prompt Specificity: Improving Temperature Conversion Function with Clear Instructions   PROMPT:  Write a python code for temperature conversion(like celsius to kelvin and vice versa)  CODE:  Screenshot 2025-09-02 at 05.27.19  **Expected Output#5**   * Code quality difference analysis for various prompts   OBSERVATION:  We could observe from this that colab takes user input and acts accordingly whether the given input is in celsius or kelvin  **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 |  |