|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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** | | | Week2 - Tuesday | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicableto**  **Batches** | |  | | | |
| **AssignmentNumber:4.2**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***ExpectedTime***  ***to complete*** |  |
|  | 1 | Lab 4: Advanced Prompt Engineering – Zero-shot, One-shot, and Few-shot Techniques  **Lab Objectives:**   * To explore and apply different levels of prompt examples in AI-assisted code generation. * To understand how zero-shot, one-shot, and few-shot prompting affect AI output quality. * To evaluate the impact of context richness and example quantity on AI performance. * To build awareness of prompt strategy effectiveness for different problem types.   **Lab Outcomes (LOs):**  After completing this lab, students will be able to:   * Use zero-shot prompting to instruct AI with minimal context. * Use one-shot prompting with a single example to guide AI code generation. * Apply few-shot prompting using multiple examples to improve AI responses. * Compare AI outputs across the three prompting strategies.   **Task Description#1**   * **Zero-shot:** Prompt AI with only the instruction — Write a Python function to generate the Fibonacci sequence up to n terms   **Expected Output#1**   * A working function without using any sample inputs/outputs.   Prompt:write a python function to generate the Fibonacci sequence up to n terms.  Code:    Test and output:    **Task Description#2**   * One-shot: Provide one example: Input: 100, Output: 37.78 to help AI generate a function that converts Fahrenheit to Celsius.   **Expected Output#2**   * A correct conversion function guided by the single example.   Prompt:provide one example to help ai generator a function that converts Fahrenheit.  Code and output:    **Task Description#3**   * **Few-shot:** Give 2–3 examples to create a function that extracts the domain name from an email address.   **Expected Output#3**   * Accurate function that returns only the domain portion of an email (e.g., @gmail.com).   Prompt:create a function that extracts the domain name from an email address.  Code :    **Output:**    **Task Description#4**   * Compare zero-shot vs few-shot prompting for generating a function that checks whether a word is a palindrome, ignoring punctuation and case.   **Expected Output#4**   * Output comparison + student explanation on how examples helped the model.   Prompt:compare zero short a function that checks whether a word is a palindrome,ignore punctuation and case.  Code and ouput:  Zero shot:    **Prompt:compare few shot a function** that checks whether a word is a palindrome, ignoring punctuation and case.  **Code :**  **Few short:**    **Output:**    **Comprasion:**  **Zero-shot: The model generates a generic solution based only on the prompt. It may miss edge cases or ignore some requirements.**   * **Few-shot: By providing examples, the model better understands the requirements (e.g., ignoring punctuation and case) and produces a more accurate, robust function.**   **Explanation:**   * **Student Explanation: Examples help the model learn the expected input/output behavior, clarify edge cases, and ensure the function meets all requirements. With few-shot prompting, the model is less likely to overlook details and more likely to produce code that works for real-world cases.**   **Task Description#5**   * Use few-shot prompting with 3 sample inputs to generate a function that determines the maximum of three numbers without using the built-in max() function.   **Expected Output#5**   * A function that handles all cases with correct logic based on example patterns.   Prompt: function that determines the maximum of three numbers without using the built-in max() function.  Code and 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** | | --- | --- | | Zero Shot (Task #1) | 0.5 | | One Shot (Task#2) | 0.5 | | Few Shot (Task#3 & Task #5) | 1.0 | | Comparison (Task#4) | 0.5 | | **Total** | **2.5 Marks** | | | | | | | Week2 - Wednesday |  |