**Course Code: AIPP**

**Assignment No: 4**

**Done by: 2503B05203 (M.Tech)**

**Name: B Subhash Chandra**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE** | | | | | **DEPARTMENT OF COMPUTER SCIENCE ENGINEERING** | | | | |
| **Program Name:** M.Tech. and MCA | | | | **Assignment Type: Lab** | | | **AcademicYear:**2025-2026 | | |
| **Course Coordinator Name** | | | | Venkataramana Veeramsetty | | | | | |
| **Course Code** | | |  | **Course Title** | | AI Assisted Problem Solving Using Python | | | |
| **Year/Sem** | | | I/I | **Regulation** | | R24 | | | |
| **Date and Day**  **of Assignment** | | | Week3 - Monday | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicable to**  **Batches** | | M.Tech. and MCA | | | |
| **AssignmentNumber:4.3**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***Expected Time***  ***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 to write a function that checks whether a given year is a leap year.   **Code:**    **Explanation:**  The core logic can be condensed into two primary cases that result in a leap year:   * Case 1: Standard Leap Year   + The year is divisible by 4 AND not divisible by 100. (e.g., 2004, 2096) * Case 2: Century Exception   + The year is divisible by 400. (e.g., 2000, 2400)   This is precisely what is captured in the elegant boolean expression:  (yearmod4=0∧yearmod100  =0) ∨ (yearmod400=0)  **Expected Output#1**   * AI-generated function with no examples provided     **Task Description#2**   * One-shot: Give one input-output example to guide AI in writing a function that converts centimeters to inches.   **Code:**    **Explanation:**   * Function Definition: The script defines a function named cm\_to\_inches that accepts one argument: the measurement in centimeters. * Conversion Formula: Inside the function, it calculates the equivalent measurement in inches by dividing the input centimeters by the conversion factor 2.54. * Rounding: The result is then rounded to two decimal places for a clean, practical output. * User Input: The script prompts the user to "Enter centimeters:" and converts their text input into a decimal number (float). * Function Call: The input value is passed to the cm\_to\_inches function to perform the conversion. * Output: Finally, the script prints the result, clearly showing the   **Expected Output#2**   * Function with correct conversion logic     **Task Description#3**   * Few-shot: Provide 2–3 examples to generate a function that formats full names as “Last, First”.   **Code:**    **Explanation:**   * format\_name (Simple): Splits the name by spaces and assumes "First Last". It correctly returns "Last, First" but mishandles any middle names. * format\_name\_v2 (Middle Name Ready): Uses rsplit(maxsplit=1) to ensure the last word is separated as the last name, grouping all preceding words as the first and middle names. * format\_name\_v3 (Robust): Includes input validation (checks for valid string/name length) and explicitly separates the name into First, Last, and Middle parts before reassembling them. * Goal: All functions aim to transform a name structure like "First [Middle] Last" into the professional format "Last, First [Middle]"   **Expected Output#3**   * Well-structured function respecting the examples     **Task Description#4**   * Compare zero-shot and few-shot prompts for writing a function that counts the number of vowels in a string.   Compare zero-shot vs few-shot prompts for writing a function that counts vowels.  This script contains:   * example zero-shot and few-shot prompt texts (what you'd send to an LLM) * two resulting function implementations (simulated LLM outputs) * quick tests and a short printed comparison summary   **Test and Main code:**    **Zero Short:**  **zero\_shot\_prompt:**      "Write a Python function named count\_vowels(s) that returns the number of vowels "      "in the input string s. Count only a, e, i, o, u (case-insensitive)."  **Code:**    **Few Short**  **few\_shot\_prompt:**      "# Example 1\n"      "Input: 'hello' -> Output: 2\n"      "# Example 2\n"      "Input: 'AEIOU' -> Output: 5\n"      "# Now produce the function:\n"      "Write a Python function named count\_vowels(s) that returns the number of vowels "      "in the input string s. Count only a, e, i, o, u (case-insensitive)."  **Code:**    **Expected Output#4**   * Functional output and comparative reflection      * **Summary:** Zero-shot: often produces a correct, minimal implementation from the single instruction. * Few-shot: by providing examples first, the model will more likely match expected behavior and style. * Example caveat: neither implementation handles accented vowels; add normalization if needed (NFKD + filter).   **Task Description#5**   * Use few-shot prompting to generate a function that reads a .txt file and returns the number of lines.     **Expected Output#5**   * Working file-processing function with AI-guided logic     **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) | 2.5 | | One Shot (Task#2) | 2.5 | | Few Shot (Task#3 & Task #5) | 2.5 | | Comparison (Task#4) | 2.5 | | **Total** | **10 Marks** | | | | | | | Week3 - Monday |  |