|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE** | | | | | **DEPARTMENT OF COMPUTER SCIENCE ENGINEERING** | | | | |
| **Program Name:** B. Tech | | | | **Assignment Type: Lab** | | | **Academic Year:**2025-2026 | | |
| **Course Coordinator Name** | | | | 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) | | | | | | |
| **Course Code** | | | 24CS002PC215 | **Course Title** | | AI Assisted Coding | | | |
| **Year/Sem** | | | II/I | **Regulation** | | R24 | | | |
| **Date and Day**  **of Assignment** | | | Week2 - Monday | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicable to**  **Batches** | |  | | | |
| **Assignment Number:4.1**(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 #1 – Zero-Shot Prompting with Conditional Validation**  Objective  Use zero-shot prompting to instruct an AI tool to generate a function that validates an Indian mobile number.  Requirements   * The function must ensure the mobile number:   + Starts with 6, 7, 8, or 9   + Contains exactly 10 digits   Expected Output   * A valid Python function that performs all required validations without using any input-output examples in the prompt. * **PROMPT:** * **Write a python code using function a zero shot to generate an validates an Indian mobile number.** * **CURSOR AI INPUT :**   **OUTPUT:**    **VS CODE INPUT AND OUTPUT:**    **Task #2 – One-Shot Prompting with Edge Case Handling**  Objective  Use one-shot prompting to generate a Python function that calculates the factorial of a number.  Requirements   * Provide one sample input-output pair in the prompt to guide the AI. * The function should handle:   + 0! correctly   + Negative input by returning an appropriate message   Expected Output   * A Python function with correct factorial logic and edge case handling, generated from a single example. * **PROMPT:** * **CURSOR AI INPUT:** * **OUTPUT:** * **VS CODE INPUT AND OUTPUT:**   **Task #3 – Few-Shot Prompting for Nested Dictionary Extraction**  Objective  Use few-shot prompting (2–3 examples) to instruct the AI to create a function that parses a nested dictionary representing student information.  Requirements   * The function should extract and return:   + Full Name   + Branch   + SGPA   Expected Output   * A reusable Python function that correctly navigates and extracts values from nested dictionaries based on the provided examples * **PROMPT:** * **Write a python code using function that generate a nested student class and fullname,branch,sgpa and using few-shot.** * **CURSOR AI INPUT:**   **OUTPUT:**    **VS CODE INPUT AND OUTPUT:**    **Task #4 – Comparing Prompting Styles for File Analysis**  Objective  Experiment with zero-shot, one-shot, and few-shot prompting to generate functions for CSV file analysis.  Requirements   * Each generated function should:   + Read a .csv file   + Return the total number of rows   + Count the number of empty rows   + Count the number of words across the file   Expected Output   * Working Python functions for each prompting style, with a brief reflection comparing their accuracy, clarity, and efficiency. * **PROMPT1:**   **(**zero shot): Write a Python function that opens a CSV file and tells me how many rows it has, how many rows are empty, and how many words are in the file.  **CURSOR AI INPUT:**      **OUTPUT:**    **VS CODE:**    **OUTPUT:**    **PROMPT 2:**  (one shot ) : "Write a Python function that opens a CSV file and gives the total rows, empty rows, and word count; for example, if the file has 4 rows with 1 empty row and 4 words in total, the function should return those numbers."  **VS CODE INPUT:**      **OUTPUT:**    **CURSOR AI:**  **INPUT:**      **OUTPUT:**    **PROMPT 3:**  ( few shot) : Write a Python function that opens a CSV file and reports total rows, empty rows, and word count; for instance, in one case a file with 4 rows, 1 empty row, and 4 words should return those numbers, and in another case a file with 5 rows, 2 empty rows, and 5 words should return those numbers." **CURSOR AI :**  **INPUT:**    **OUTPUT:**  **VS CODE:**  **INPUT:**    **OUTPUT:**    **REPORT:**  **:** Here at the task 4 I have made three prompts the three prompts are on the conditions of given like zero shot , one shot , few shot. So I have noticed that in the first one it made a python function and made the python code which its output was total rows , empty rows , total words etc are 0. Where as in second task which is one shot same as usually similarly it made a function and develop a code this time its output was total rows was 4 , empty rows 0 , total words 7. In the third one it made a python function as usual I made A python and developed it at this point the out put same as output 2 because I have mentioned the same words in csv txt file. Whats the difference in output 3 is in this we need to enter the txt csv file name so that it calculated and gives the total rows and words and empty rows. First intial thing we need to create a txt csv file in the folder of lab 4and name it like I have named SRU.csv.txt.    **Task #5 – Few-Shot Prompting for Text Processing and Word** **Frequency**  Objective  Use few-shot prompting (with at least 3 examples) to generate a Python function that processes text and analyzes word frequency.  Requirements  The function must:   * Accept a paragraph as input * Convert all text to lowercase * Remove punctuation * Return the most frequently used word   Expected Output   * A functional Python script that performs text cleaning, tokenization, and returns the most common word using only the examples provided in the prompt * **PROMPT:** * **Write a python code using function and the processes text and analyses word frequently and the function must accept the paragraph and convert it into lowercase and remove the punctuation and return the most frequently used word in few short .** * **CURSOR AI INPUT:**   **OUTPUT:**    **VS CODE INPUT 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#4 & Task #5) | 1.5 | | **Total** | **2.5 Marks** | | | | | | | Week2 - Monday |  |