|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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: : generate a python function must ensure the Indian mobile number that starts with 6,7,8 or 9 and contains exactly 10 numbers.  GITHUB COPILOT:      CURSOR AI :      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: generate a python function that calculate a factorial of the number that handles 0! Correctly and Negative input by returning an appropriate message if input is -5 the output should be not defined  GITHUB COPILOT:      CURSOR AI:      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:  GITHUB COPILOT:      CURSOR AI:      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.  USING ZERO SHOT:  PROMPT: Write a Python function that reads a .csv file and returns:  - Total number of rows  - Number of empty rows  - Total number of words across the file  GITHUB COPILOT:      CURSOR AI:      USING ONE SHOT:  Prompt: Write a Python function that reads a .csv file and returns:  - Total number of rows  - Number of empty rows  - Total number of words across the file  Example:  If the CSV file contains:  Name, Age  Amit, 21  (empty row)  Riya, 22  Output: Total Rows = 4, Empty Rows = 1, Word Count = 6  GITHUB COPILOT:      CURSOR AI:      USING FEW SHOT:  Prompt:  Write a Python function that reads a .csv file and returns:  - Total number of rows  - Number of empty rows  - Total number of words across the file  Example 1:  CSV:  Name, Age  Amit, 21  (empty row)  Riya, 22  Output: Total Rows = 4, Empty Rows = 1, Word Count = 6  Example 2:  CSV:  Title, Marks  Maths, 95  Science, 88  History, 76  Output: Total Rows = 4, Empty Rows = 0, Word Count = 7  GITHUB COPILOT:      CURSOR AI:      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: # Example 1  # Input: "Hello, hello! How are you?"  # Output: "hello"  # Example 2  # Input: "The quick brown fox jumps over the lazy dog. The dog was not amused."  # Output: "the"  # Example 3  # Input: "Apples and oranges, apples are great. Oranges are fine too!"  # Output: "apples"  # Now write a Python function that follows the same logic:  # - Takes a paragraph as input  # - Converts text to lowercase  # - Removes punctuation  # - Splits the text into words  # - Returns the most frequent word  def most\_frequent\_word(text):  import string  from collections import Counter  # Convert to lowercase  text = text.lower()  # Remove punctuation  text = text.translate(str.maketrans('', '', string.punctuation))  # Tokenize (split into words)  words = text.split()  # Count word frequency  word\_counts = Counter(words)  # Return the most common word  return word\_counts.most\_common(1)[0][0]  GITHUB COPILOT:      CURSOR AI:      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 |  |