|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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.  **PROMPT:**  Write a python function that validates Indian mobile number . The python function ensures that whether the mobile number starts with 6,7,8,9 and it contains exact 10 digits. We need to give the input and the function ensures.  **CURSOR AI CODE SCREENSHOT:**    **GITHUB COPILOT CODE:**    Requirements   * The function must ensure the mobile number:   + Starts with 6, 7, 8, or 9   + Contains exactly 10 digits   **CURSOR AI CODE OUPUT:**      **GITHUB COPILOT OUTPUT:**    Expected Output   * A valid Python function that performs all required validations without using any input-output examples in the prompt.   **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.  **PROMPT:** Write a python function to calculate factorial of a given number from the user . If the given number is negative then give a appropriate message. Example:2!=2  Requirements.  **CURSOR AI CODE SCREENSHOT:**     * 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   **GITHUB COPILOT CODE:**    Expected Output   * A Python function with correct factorial logic and edge case handling, generated from a single example.   **CURSOR AI CODE OUPUT:**      **GITHUB COPILOT 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.  **PROMPT:** Write a python function that represents nested dictionary representing student information . The function should extract and return. Example 1: Full Name : Siddu , Branch : CSE , SGPA : 9.0  Example 2: Full Name : harsha , Branch : CSM , SGPA : 9.2  **CURSOR AI CODE SCREENSHOT:**      **GITHUB COPILOT CODE:**    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.   **CURSOR AI CODE OUPUT:**    **GITHUB COPILOT 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   **PROMPT:**  Zero shot prompting:  Expected Output   * Working Python functions for each prompting style, with a brief reflection comparing their accuracy, clarity, and efficiency.   **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   **PROMPT:** Write a python function that process text and analyse word frequency . We need to give input of paragraph and convert all text into lowercase and remove punctuation and return the most frequently used word.  Example1 :  SR University, located in Warangal, Telangana, is a leading institution known for its innovation-driven education. It offers a wide range of programs in engineering, management, and sciences, fostering research and entrepreneurship. With modern infrastructure and skilled faculty, SR University provides students with opportunities to excel globally.  Example 2:  Telangana, located in southern India, is known for its rich history, diverse culture, and rapid development. It is home to iconic landmarks like the Charminar, Golconda Fort, and the serene Hussain Sagar Lake. With thriving industries and a strong IT hub in Hyderabad, Telangana is a blend of tradition and modernity.  Example 3:  India, the world’s largest democracy, is a land of diverse cultures, languages, and traditions. It is known for its rich history, iconic monuments like the Taj Mahal, and spiritual heritage. With rapid economic growth and technological advancements, India continues to shine on the global stage.  **CURSOR AI CODE SCREENSHOT:**      **GITHUB COPILOT CODE:**    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   **CURSOR AI CODE OUPUT:**    **GITHUB COPILOT 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 |  |