|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 program to generate an Indian mobile number which starts with 6,7,8,or 9 and contains exactly 10 digits.  **Code & Output:**  **1.** **Cursor**  **2.Visual Studio Code**    **Explaination:**  This Python script generates a random valid Indian mobile number. Here’s how it works:   * It imports the [random](vscode-file://vscode-app/c:/Users/KRUTHIKAPC/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) module to help generate random numbers. * The function [generate\_indian\_mobile\_number()](vscode-file://vscode-app/c:/Users/KRUTHIKAPC/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html):   + Picks the first digit randomly from 6, 7, 8, or 9 (since Indian mobile numbers start with these digits).   + Generates the remaining 9 digits randomly (each between 0 and 9).   + Combines the first digit and the remaining digits to form a 10-digit mobile number. * When you run the script, it calls the function and prints the generated mobile number.   **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:** Write a python function to calculate the factorial of the given number.  **Ex:** Input=5  Output:  The factorial of 5=120  **Code & Output:**  **1.Cursor**    **2.VS Code**    **Explaination:**  This Python script calculates the factorial of a number entered by the user:   * It asks the user to enter a number and converts the input to an integer. * It initializes a variable [factorial](vscode-file://vscode-app/c:/Users/KRUTHIKAPC/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) to 1. * It uses a for loop to multiply [factorial](vscode-file://vscode-app/c:/Users/KRUTHIKAPC/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) by every number from 1 up to the entered number. * Finally, it prints the result in the format: "The factorial of [number]=[factorial]".   **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 program to create a function that parses a nested dictionary representing student information with user as input.  **Ex:**   * Input: * First Name: Tej * Last Name: Deva * Branch: CSE * SGPA: 9.0   Output:  Student Details:   * Full Name: Tej Deva * Branch: CSE * SGPA:9 * Input: * First Name: God * Last Name: Zilla * Branch: EEE * SGPA: 4.0   Output:  Student Details:   * Full Name: God Zilla * Branch: EEE * SGPA:4   **Code & Output:**  **1.Cursor**      **2.VS Code**    **Output**  **Explaination:**  **This script collects and displays student information:**   * **The**[**get\_student\_input()**](vscode-file://vscode-app/c:/Users/KRUTHIKAPC/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)**function asks the user to enter their first name, last name, branch, and SGPA, then stores this data in a nested dictionary.** * **The**[**parse\_student\_info(student\_dict)**](vscode-file://vscode-app/c:/Users/KRUTHIKAPC/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)**function takes this dictionary, combines the first and last names, gets the branch, and converts the SGPA to an integer (after converting it to float first). It then prints the student's details in a formatted way.** * **When you run the script, it collects the student's info and displays it using the above functions.**   **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.   **Prompt\_1**: Write a python program to generate functions of CSV file Analysis(default input), which should read a file ,return the total number of rows,count the number of empty rows,count the number of words across the file.  **Code &Output:**   1. **Cursor**            1. **VS Code**       **Prompt\_2**: Write a python program to generate functions of CSV file Analysis like below  Ex:-  Input: Name,Age,City,Occupation  John Doe,25,New York,Software Engineer  Jane Smith,30,Los Angeles,Data Analyst  Bob Johnson,35,Chicago,Project Manager  Output:  Final Analysis is   * Total number of rows: 4 * Number of empty rows: 0 * Total number of words: 21 * Non-empty rows: 4   **Code & Output:**  **1.Cursor**        **2.VS Code**      **Prompt\_3**: Write a python program to generate functions of CSV file Analysis like below  Ex:-  Input: Name,Age,City,Occupation  John Doe,25,New York,Software Engineer  Jane Smith,30,Los Angeles,Data Analyst  Bob Johnson,35,Chicago,Project Manager  Output:  Final Analysis is   * Total number of rows: 4 * Number of empty rows: 0 * Total number of words: 21 * Non-empty rows: 4   **Ex:**  **Input:** **,**,  David Lee,29,San Francisco,Developer  Emma Davis,27,Austin,Designer  ,,  Frank Miller,31,Denver,Engineer  Grace Taylor,26,Portland,Analyst  Output:  Final Analysis is   * Total number of rows: 6 * Number of empty rows: 2 * Total number of words: 17 * Non-empty rows: 4   **Code & Output:**  **1.Cursor**  **Explaination:**  In all the above prompts we have taken zero short, one short, few short prompting’s.  In most of all the code generated by cursor ai was default input compared to visual studio code.  In Vs code it was asking the user to input the file name and then giving output.  **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   **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 |  |