Hall:2303A51812 Name:Tejesh Batch-26

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Assignment Number:1.3**(Present assignment number)/**24**(Total number of assignments) | | | | |
|  | | | | |
|  | **Q.No.** | **Question** | ***Expected Time***  ***to complete*** |  |
|  | 1 | Lab 2: Exploring Additional AI Coding Tools beyond Copilot – Gemini (Colab) and Cursor AI  **Lab Objectives:**   * To explore and evaluate the functionality of Google Gemini for AI-assisted coding within Google Colab. * To understand and use Cursor AI for code generation, explanation, and refactoring. * To compare outputs and usability between Gemini, GitHub Copilot, and Cursor AI. * To perform code optimization and documentation using AI tools.   **Lab Outcomes (LOs):**  After completing this lab, students will be able to:   * Generate Python code using Google Gemini in Google Colab. * Analyze the effectiveness of code explanations and suggestions by Gemini. * Set up and use Cursor AI for AI-powered coding assistance. * Evaluate and refactor code using Cursor AI features. * Compare AI tool behavior and code quality across different platforms.   **Task 1: Word Frequency from Text File**   * **Scenario:** You are analyzing log files for keyword frequency. * **Task:** Use Gemini to generate Python code that reads a text file and counts word frequency, then explains the code. * **Expected Output:**   + Working code   + Explanation   + Screenshot   **Task 2: File Operations Using Cursor AI**   * **Scenario:** You are automating basic file operations. * **Task:** Use Cursor AI to generate a program that:   + Creates a text file   + Writes sample text   + Reads and displays the content * **Expected Output:**   + Functional code   + Cursor AI screenshots   **Task 3: CSV Data Analysis**   * **Scenario:** You are processing structured data from a CSV file. * **Task:** Use Gemini in Colab to read a CSV file and calculate mean, min, and max. * **Expected Output:**   + Correct output   + Screenshot   **Task 4: Sorting Lists – Manual vs Built-in**   * **Scenario:** You are reviewing algorithm choices for efficiency. * **Task:** Use **Gemini** to generate:   + Bubble sort   + Python’s built-in sort()   + Compare both implementations. * **Expected Output:**   + Two versions of code   + Short comparison   **Note: Report should be submitted as a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots.** | Week1 - Monday |  |