**Course Code: AIPP**

**Assignment No: 2**

**Done by: 2503B09902 (MCA)**

**Name: Sannitha Reddy**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE** | | | | | **DEPARTMENT OF COMPUTER SCIENCE ENGINEERING** | | | | |
| **Program Name:** M. Tech/MCA | | | | **Assignment Type: Lab** | | | **AcademicYear:**2025-2026 | | |
| **Course Coordinator Name** | | | | Venkataramana Veeramsetty | | | | | |
| **Course Code** | | |  | **Course Title** | | AI Assisted Problem Solving Using Python | | | |
| **Year/Sem** | | | I/I | **Regulation** | | R24 | | | |
| **Date and Day**  **of Assignment** | | | Week1 - TUESDAY | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicable to**  **Batches** | | M. Tech/MCA | | | |
| **AssignmentNumber:2.3**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***Expected Time***  ***to complete*** |  |
|  | 1 | Lab 2: Exploring Additional AI Coding Tools – 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 Description#1**   * Use Google Gemini in Colab to write a function that reads a CSV file and calculates mean, min, max.   **Prompt:**   * write a python function that reads a CSV file and calculates mean, min, max. * Modify this code so that it automatically creates a sample CSV file and then runs the function without asking for user to enter CSV file   **Code:**        **Expected Output#1**   * Functional code with output and screenshot   **Output:**  **Task Description#2**   * Compare Gemini and Copilot outputs for a palindrome check function.   **Prompt:**   * write a code to check palindrome function.   **Code:**  GitHub copilot    Gemini code    **Expected Output#2**   * Side-by-side comparison and observations   **Copilot(output)**    **Gemini(output)**    **Comparison and Observations**   * Gemini generated a short and simple palindrome code using text == text [: -1]. * It worked correctly for small words but didn’t handle capital letters or spaces. * Copilot’s code automatically converted the string to lowercase and removed spaces. * Copilot’s version gave accurate results even with mixed cases or spaces. * Both tools gave correct logic, but Copilot’s code was completer and more practical. * Overall, Copilot provided a better real-time solution compared to Gemini.   **Task Description#3**   * Ask Gemini to explain a Python function (to calculate area of various shapes) line by line.   **Prompt:**   * Write a Python function to calculate area of various shapes   **Code:**    **Expected Output#3**   * Detailed explanation with code snippet   **Output:**    **Line-by-Line Explanation**   1. import math → to use the value of π (pi) for circle area. 2. def calculate area(shape, \*\*kwargs): → defines a function that takes shape name and other values like radius, length, etc. 3. shape = shape. Lower() → converts the shape name into lowercase. 4. if shape == "circle": → checks if the shape is circle. 5. if 'radius' in kwargs: → verifies radius is given or not. 6. radius = kwargs['radius'] → takes radius value from user input. 7. return math.pi \* radius\*\*2 → uses formula πr² to find circle area. 8. Elif shape == "rectangle": → checks if the shape is rectangle. 9. length = kwargs['length'] and width = kwargs['width'] → takes length and width values. 10. return length \* width → finds rectangle area using l ×w. 11. Elif shape == "triangle": → checks if the shape is triangle. 12. base = kwargs['base'] and height = kwargs['height'] → takes base and height values. 13. return 0.5 \* base \* height → applies ½ × base × height formula. 14. else: → runs when the shape is not supported. 15. print (f"Error: Shape '{shape}' not supported.") → shows error message. 16. Example prints → display the area for each shape or an error if shape is not found.   **Task Description#4**   * Install and configure Cursor AI. Use it to generate a Python function (e.g., sum of squares).       **Prompt:**   * Write a Python function named sum\_of\_squares that takes a list of numbers * and returns the sum of their squares. Include a docstring and an example.   **Code:**    **Expected Output#4**   * Screenshots of working environments with few prompts to generate python code   **Output:**      **Screenshots few prompts to generate python code:**  **Prompt:** Add input validation and error handling.    **Prompt:** Add comments and time complexity explanation.    **Task Description#5**   * Student need to write code to calculate sum of add number and even numbers in the list   **Prompt:**   * Write a function that returns sum of even and odd numbers in a list**.** * Refactor the above code with better readability, add comments and error handling.   **Code without refactored:**    **Output:**    **Expected Output#5**   * Refactored code written by student with improved logic   **Refactored code:**    **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** | | --- | --- | | Successful Use of Gemini in Colab (Task#1 & #2) | 2.5 | | Code Explanation Accuracy (Gemini) (Task#3) | 2.5 | | Cursor AI Setup and Usage (Task#4) | 2.5 | | Refactoring and Improvement Analysis (Task#5) | 2.5 | | **Total** | **10 Marks** | | | | | | | Week1 - TuesDay |  |