|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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) | | | | | | | | | |
|  | | | | | | | | | |
| **NAME : Shreemayi Katakam**  **Roll No: 2503B05124**  **Branch: M-TECH(CSE)** | | | | | | | | | |
|  | **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.   **Code:**      **Expected Output#1**   * Functional code with output and screenshot     \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  **Task Description#2**   * Compare Gemini and Copilot outputs for a palindrome check function.   **Gemini code:**    **Copilot code:**      **Expected Output#2**   * Side-by-side comparison and observations   **Gemini output:**    **Copolit output:**    **Comparsions:**  💠 Gemini   * Gemini’s palindrome program is simple and beginner-friendly. * It uses string reversal ([::-1]) to check if a cleaned version of the text matches its reverse. * The code is short, easy to read, and performs all operations inside one function. * It ignores spaces, punctuation, and case, making it accurate for most test cases. * However, it’s less efficient because it creates a reversed copy of the string (O(n) space).   💠 Copilot   * Copilot’s palindrome code is more structured and optimized. * It uses a two-pointer approach, comparing characters from both ends, which saves memory (O(1) space). * The program includes a helper function \_normalize() for text cleaning and supports optional flags like ignore\_case and ignore\_non\_alnum. * It is more configurable and professional, suitable for larger or reusable projects. * However, it’s more complex and slightly harder for beginners to follow.   **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  **Task Description#3**   * Ask Gemini to explain a Python function (to calculate area of various shapes) line by line..   **Code:**    **Expected Output#3**   * Detailed explanation with code snippet     **Explanation:**  This Python program calculates the area of different geometric shapes using separate functions: (1) The math module is imported to use math.pi for π in circle calculations. (2) The area\_circle(radius) function uses the formula πr² to compute the circle’s area. (3) The area\_rectangle(length, width) function calculates area as length × width. (4) The area\_triangle(base, height) function applies ½ × base × height to find the triangle’s area. (5) The area\_square(side) function returns side² for the square’s area. (6) The area\_trapezoid(base1, base2, height) function uses ½ × (base₁ + base₂) × height for trapezoid area. (7) Inside the if \_\_name\_\_ == "\_\_main\_\_": block, all functions are tested with sample values and results are displayed using formatted output. The output shows: Circle = 78.54, Rectangle = 24, Triangle = 6.0, Square = 25, and Trapezoid = 16.0.  **\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***  **Task Description#4**   * Install and configure Cursor AI. Use it to generate a Python function (e.g., sum of squares).   Install cursor:    **Code:**   * Prompt to generate python code * **generate sum of squares of two numbers and input should be given by user**     **Expected Output#4**   * Screenshots of working environments with few prompts to generate python code     **Task Description#5**   * Student need to write code to calculate sum of add number and even numbers in the list     **Expected Output#5**   * Refactored code written by student with improved logic     **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 |  |