|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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.   **Expected Output#1**   * Functional code with output and screenshot   Prompt: Write a Python function to read a CSV file and calculate the mean, minimum, and maximum values for each numeric column.  **Code:**      **Output:**    **Task Description#2**   * Compare Gemini and Copilot outputs for a palindrome check function.   Prompt:  Write a Python function that checks if a word is a palindrome.  Gemini code:    Copilot code:    **Expected Output#2**   * Side-by-side comparison and observations     **Overall Observations:**   * Both programs correctly handle case insensitivity and ignore punctuation. * Differences in results mainly come from using **different test strings** (e.g., “Race Car” vs “race a car”). * Both outputs confirm that the palindrome-checking logic is working properly in each script. * The second version (PowerShell output) shows simpler formatting (no “Is … ?” structure), while the first uses a question–answer style output.   **Task Description#3**   * Ask Gemini to explain a Python function (to calculate area of various shapes) line by   line..        **Expected Output#3**   * Detailed explanation with code snippet   **Code snippet:**  **import math**  **# --- 1. Define Calculation Functions for Each Shape ---**  **def calculate\_square\_area(side):**  **"""Area of a square: side \* side"""**  **return side \* side**  **def calculate\_circle\_area(radius):**  **"""Area of a circle: pi \* radius^2"""**  **return math.pi \* radius\*\*2**  **def calculate\_rectangle\_area(length, width):**  **"""Area of a rectangle: length \* width"""**  **return length \* width**  **def calculate\_triangle\_area(base, height):**  **"""Area of a triangle: 0.5 \* base \* height"""**  **return 0.5 \* base \* height**  **# --- 2. Dictionary Dispatcher ---**  **# Maps shape names to their respective calculation function and required parameters.**  **SHAPE\_CALCULATORS = {**  **'square': {'func': calculate\_square\_area, 'params': ['side']},**  **'circle': {'func': calculate\_circle\_area, 'params': ['radius']},**  **'rectangle': {'func': calculate\_rectangle\_area, 'params': ['length', 'width']},**  **'triangle': {'func': calculate\_triangle\_area, 'params': ['base', 'height']}**  **}**  **# --- 3. Main Dispatcher Function ---**  **def calculate\_area(shape, \*\*kwargs):**  **"""**  **Calculates the area of a shape using a dictionary dispatcher.**  **Example: calculate\_area('rectangle', length=5, width=10)**  **"""**  **shape = shape.lower()**  **# Check if the requested shape is supported**  **if shape not in SHAPE\_CALCULATORS:**  **return f"Error: Shape '{shape}' not supported."**  **# Get the calculator details**  **calculator = SHAPE\_CALCULATORS[shape]**  **func = calculator['func']**  **required\_params = calculator['params']**    **# Check for required parameters**  **args = {}**  **for param in required\_params:**  **if param in kwargs:**  **args[param] = kwargs[param]**  **else:**  **return f"Error: {shape.capitalize()} requires '{param}' dimension(s)."**  **# Execute the calculation function with the collected arguments**  **return func(\*\*args)**  **# --- Example Usage ---**  **print(f"Square Area: {calculate\_area('square', side=7)}")**  **print(f"Rectangle Area: {calculate\_area('rectangle', length=8, width=5)}")**  **print(f"Triangle Area: {calculate\_area('triangle', base=10, height=4)}")**  **print(f"Circle Error: {calculate\_area('circle', radius\_incorrect=3)}")**  **Task Description#4**   * Install and configure Cursor AI. Use it to generate a Python function (e.g., sum of squares).           **Expected Output#4**   * Screenshots of working environments with few prompts to generate python code   # Sample program:      **Task Description#5**   * Student need to write code to calculate sum of add number and even numbers in the list   Prompt: Write a Python program to calculate the sum of odd numbers and even numbers in a list and print the results separately.  Code:    **Expected Output#5**   * Refactored code written by student with improved logic   **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 |  |