2303A51200

Batch 04

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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** | | | | Dr. Rishabh Mittal | | | | | |
| **Instructor(s) Name** | | | | |  | | --- | | Mr. S Naresh Kumar | | Ms. B. Swathi | | Dr. Sasanko Shekhar Gantayat | | Mr. Md Sallauddin | | Dr. Mathivanan | | Mr. Y Srikanth | | Ms. N Shilpa | | Dr. Rishabh Mittal (Coordinator) | | Dr. R. Prashant Kumar | | Mr. Ankushavali MD | | Mr. B Viswanath | | Ms. Sujitha Reddy | | Ms. A. Anitha | | Ms. M.Madhuri | | Ms. Katherashala Swetha | | Ms. Velpula sumalatha | | Mr. Bingi Raju | | | | | | |
| **CourseCode** | | | 23CS002PC304 | **Course Title** | | AI Assisted Coding | | | |
| **Year/Sem** | | | III/II | **Regulation** | | R23 | | | |
| **Date and Day**  **of Assignment** | | | **Week1 – Tuesday** | **Time(s)** | | 23CSBTB01 To 23CSBTB52 | | | |
| **Duration** | | | 2 Hours | **Applicable to**  **Batches** | | All batches | | | |
| **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: Cleaning Sensor Data**   * **Scenario**: * You are cleaning IoT sensor data where negative values are invalid. * **Task:**   Use Gemini in Colab to generate a function that filters out all negative numbers from a list.   * Expected Output:   + Before/after list   + Screenshot of Colab execution * OUTPUT * PROMPT: * Write a python function to remove negative numbers from the list * **Explanation:** * The function loops through each number in the input list. * It checks if the number is 0 or positive. * Only non-negative numbers are added to a new list. * Finally, it returns the filtered list without negative values.   **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Task 2: String Character Analysis**   * **Scenario:**   You are building a text-analysis feature.   * **Task:**   Use Gemini to generate a Python function that counts vowels, consonants, and digits in a string.   * **Expected Output:**   + Working function   + Sample inputs and outputs   **Output:**  **Prompt**  **Generate a python function that counts vowels constants and numbers in a string**    **Output for code**    **Eplanation:**  **The function goes through each character in the given string.**  **If the character is a letter, it checks whether it is a vowel or a consonant.**  **If the character is a digit, it increases the digit count.**  **Finally, it returns the counts of vowels, consonants, and digits.**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Task 3: Palindrome Check – Tool Comparison**   * **Scenario**:   You must decide which AI tool is clearer for string logic.   * **Task:**   Generate a palindrome-checking function using Gemini and Copilot, then compare the results.   * **Expected Output:**   + Side-by-side code comparison   + Observations on clarity and structure   Output  ***Prompt***  ***Generating palindrome code in git hub copilot , gemi***  ***ni and comparison***        **Output for code**    **Eplanation:**  **Step-by-step comparison of both palindrome codes:**  **Function Definition:**  **Both codes define a function named is\_palindrome that takes a string as input.**  **Text Cleaning:**  **In both codes, the input string is converted to lowercase and all non-alphanumeric characters are removed to ensure accurate palindrome checking.**  **Palindrome Logic:**  **Each code compares the cleaned string with its reverse using string slicing ([::-1]) to determine if it is a palindrome.**  **Return Value:**  **Both functions return True if the string is a palindrome and False otherwise.**  **Testing the Function:**  **The first code directly calls the function with print() statements, while the second code tests multiple strings inside an if \_\_name\_\_ == "\_\_main\_\_" block.**  **Code Structure:**  **The first code is simple and straightforward, whereas the second code is better structured, includes a docstring, and is more suitable for larger programs.**  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Task 4: Code Explanation Using AI**   * **Scenario:**   You are reviewing unfamiliar code written by another developer.   * **Task:**   Ask Gemini to explain a Python function (prime check OR palindrome check) line by line.   * **Expected Output:**   + Code snippet   + AI explanation   + Student comments on understanding   **Output**  *prompt*  generate a python code for prime number explain line by line    Output for code    Explanation:    **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 |  |