|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE** | | | | | **DEPARTMENT OF COMPUTER SCIENCE ENGINEERING** | | | | |
| **ProgramName:**B. Tech | | | | **Assignment Type: Lab** | | | **AcademicYear:**2025-2026 | | |
| **CourseCoordinatorName** | | | | Venkataramana Veeramsetty | | | | | |
| **Instructor(s)Name** | | | | |  | | --- | | Dr. V. Venkataramana (Co-ordinator) | | Dr. T. Sampath Kumar | | Dr. Pramoda Patro | | Dr. Brij Kishor Tiwari | | Dr.J.Ravichander | | Dr. Mohammand Ali Shaik | | Dr. Anirodh Kumar | | Mr. S.Naresh Kumar | | Dr. RAJESH VELPULA | | Mr. Kundhan Kumar | | Ms. Ch.Rajitha | | Mr. M Prakash | | Mr. B.Raju | | Intern 1 (Dharma teja) | | Intern 2 (Sai Prasad) | | Intern 3 (Sowmya) | | NS\_2 ( Mounika) | | | | | | |
| **CourseCode** | | | 24CS002PC215 | **CourseTitle** | | AI Assisted Coding | | | |
| **Year/Sem** | | | II/I | **Regulation** | | R24 | | | |
| **Date and Day**  **of Assignment** | | | Week4 - Wednesday | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicableto**  **Batches** | |  | | | |
| **AssignmentNumber:9.3**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***ExpectedTime***  ***to complete*** |  |
|  | 1 | Lab 8: Documentation Generation: Automatic documentation and code comments  **Lab Objectives:**   * To understand the importance of documentation and code comments in software development. * To explore how AI-assisted coding tools can generate meaningful documentation and inline comments. * To practice generating function-level and module-level docstrings automatically. * To evaluate the quality, accuracy, and limitations of AI-generated documentation. * To develop a small automated tool for documentation generation in Python..     **Lab Outcomes (LOs):**  After completing this lab, students will be able to:   * Apply AI-assisted coding tools to generate docstrings and inline comments for Python code. * Critically analyze AI-generated documentation for correctness, completeness, and readability. * Create structured documentation (function-level, module-level) following standard formats. * Design and implement a mini documentation generator tool to automate code commenting and docstring creation.   **Task Description#1 Basic Docstring Generation**   * Write python function to return sum of even and odd numbers in the given list. * Incorporate manual **docstring** in code with Google Style * Use an AI-assisted tool (e.g., Copilot, Cursor AI) to generate a docstring describing the function. * Compare the AI-generated docstring with your manually written one.   **Expected Outcome#1:** Students understand how AI can produce function-level documentation.    **Output:**    **Comments:**   1. **Defines a function to calculate the sum of even and odd numbers from a list.** 2. **Initializes a variable to store the sum of even numbers.** 3. **Initializes a variable to store the sum of odd numbers.** 4. **Iterates through each number in the provided list.** 5. **Checks if the current number is even.** 6. **Adds the even number to the even sum.** 7. **Adds the odd number to the odd sum if it is not even.** 8. **Returns both the even and odd sums as a tuple.** 9. **Prompts the user to enter numbers separated by spaces and processes the input.** 10. **Prints the calculated sums of even and odd numbers.**   **Task Description#2 Automatic Inline Comments**   * Write python program for **sru\_student** class with attributes like name, roll no., hostel\_status and **fee\_update** method and **display\_details** method. * Write comments manually for each line/code block * Ask an AI tool to add inline comments explaining each line/step. * Compare the AI-generated comments with your manually written one.   **Expected Output#2:** Students critically analyze AI-generated code comments.    Output:    **Comments:**   1. **Defines a class to represent an SRU student with relevant attributes.** 2. **The constructor initializes the student's name, roll number, hostel status, and fee payment status.** 3. **The fee status is set to False (not paid) by default.** 4. **Includes a method to update the fee payment status.** 5. **Includes a method to display all student details in a readable format.** 6. **Prompts the user to enter the student's name.** 7. **Prompts the user to enter the student's roll number.** 8. **Prompts the user to specify if the student stays in the hostel.** 9. **Prompts the user to specify if the fee has been paid and converts the answer to a boolean.** 10. **Creates a student object with the entered details, updates the fee status, and displays all information.**   **Task Description#3**   * Write a Python script with 3–4 functions (e.g., calculator: add, subtract, multiply, divide). * Incorporate manual **docstring** in code with NumPy Style * Use AI assistance to generate a module-level docstring + individual function docstrings. * Compare the AI-generated docstring with your manually written one.   **Expected Output#3:** Students learn structured documentation for multi-function scripts      **Output:**    **Comments:**   1. **The script defines a simple calculator with functions for addition, subtraction, multiplication, and division.** 2. **Each arithmetic operation is implemented as a separate function.** 3. **The division function includes error handling for division by zero.** 4. **The main program starts by printing a welcome message and available operations.** 5. **The user is prompted to select an operation (add, subtract, multiply, divide).** 6. **The user is then prompted to enter two numbers, which are converted to floats.** 7. **The program checks which operation was selected using if-elif-else statements.** 8. **The corresponding function is called based on the user's choice.** 9. **If the user enters an invalid operation, an error message is set as the result.** 10. **Finally, the result of the calculation is printed to the console.**   **Push documentation whole workspace as .md file in GitHub Repository**  **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** | | | | | | Week4 - Wednesday |  |