|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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.  **Prompt**  Generate a python function to print the sum of even and odd numbers in the given list by the user,using docstring.  **AI-generated Code**  IMG_256  **Manual-code**  IMG_256  **Comparison Between Manual and AI-Generated Docstrings**   1. The **manual docstring** is long, detailed, and professional. 2. The **AI-generated docstring** is short and simple but less detailed. 3. Manual version includes an **example**, AI version does not. 4. Manual version is best for **projects, reports, and teamwork**. 5. AI version is good for **quick coding and drafts**.   **Code Explanation**  The program defines a function to calculate the sum of even and odd numbers in a list. It takes numbers as input from the user, separated by spaces. Inside the function, even numbers are added to even\_sum and odd numbers to odd\_sum. Finally, it prints the total sum of even numbers and the total sum of 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.  **Prompt**   * Generate a python function class with attributes like name, roll no., hostel\_status and fee\_update method and display\_details method with user input.   **AI-generated code**  IMG_256  IMG_256  **Manual-code**  IMG_256  IMG_256  **Comparison:**  **Manual comments** = detailed, explanatory, suitable for learning & documentation.  **AI-generated comments** = short, neat, but sometimes too minimal.  **Code Explanation**  1.The program creates a sru\_student class to store student details and fees. 2.It asks the user to enter name, roll number, and hostel status. 3.It shows the student’s details using the display\_details method. 4.The user can add fees using the fee\_update method. 5.Finally, it displays the updated student details with the new fees.  **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  **Prompt**   * Generate a python code with 4 functions (e.g., calculator: add, subtract, multiply, divide) with user input using docstring   **AI-generated code**  IMG_256  IMG_256  **Manual-docstring**  IMG_256  **Comparison:**   1. Manual docstrings use **NumPy style** (Parameters, Returns). 2. AI docstrings use **Args/Returns style**, simpler format. 3. Manual docstrings are **more detailed and structured**. 4. AI docstrings are **short and quick to read**.   **Code Explanation**  The program is a simple calculator with functions to add, subtract, multiply, and divide. It asks the user to choose an operation and enter two numbers. The chosen operation is performed, and the result is displayed. The divide function handles division by zero to avoid errors.  **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 |  |