|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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#1:**  1.Generate a python code to return sum of even and odd numbers in the given list by the user.  **AI-generated code:**    **Manual code:**    **Comparison Between AI-generated Docstring and Manual**  1. Manual code writes everything step by step in the main program.  2.AI-generated code separates logic into a function with docstrings.  3. Manual code is less abstract, easy to understand line by line.  4. AI-generated code introduces functions and list.  5. Both versions use try/except for invalid input.  **Code Explanation:**  1.A function is defined to calculate the sum of even and odd numbers separately.  2.It takes a list of numbers as input and returns two values: even\_sum and odd\_sum.  3. If the user enters anything that’s not a number, a ValueError occurs.  4. Program displays the sum of even numbers and the sum of odd numbers separately.  **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#2:**  Generate a python program for sru\_student class with attributes like name, roll no., hostel\_status and fee\_update method and display\_details method.  **AI-generated code:**      **Manual Code:**      **Comparison Between AI-generated comments and Manual Comments**  1. AI comments are long and descriptive, while manual comments are short and direct.  2. AI uses formal, full-sentence explanations; manual uses simple keywords or phrases.  3. AI comments are best for learning/teaching, while manual comments are best for real coding projects.  4. AI comments suit beginners and documentation; manual comments suit developers reading code quickly.  **Code Explanation:**  1. A class sru\_student is created to represent a student.  2. Initializes the student object with name, roll\_no, and hostel\_status.  3.Prints a message showing that the student’s fee has been updated with the given amount.  4. Prints the student’s name, roll number, and hostel status  **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#3:**  Generate a Python script with 3–4 functions (e.g., calculator: add, subtract, multiply, divide).with user input.  **AI-generated code:**        **Manual Code:**      **Comparison:**  **1.AI docstring** explains the function in a **formal, structured way** (with Parameters, Returns, Raises).  2. **Manual docstring** explains it in a **short and direct way** (one-liner focusing only on what the function does).  3. **AI docstrings** are best for documentation, while **Manual docstrings** are best for real projects .  4. **AI docstrings** focus on explaining how inputs/outputs work, while **Manual docstrings** focus on what the function does.  **Code Explanation:**  1.Four functions are defined → add, subtract, multiply, divide.  2.The program asks the user to enter two numbers.  3.The user chooses which operation to perform based on that choice ,the result is printed.  4.If the user enters wrong input the program shows an error message.  **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 |  |