|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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** | | | | 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) | | | | | | |
| **Course Code** | | | 24CS002PC215 | **Course Title** | | AI Assisted Coding | | | |
| **Year/Sem** | | | II/I | **Regulation** | | R24 | | | |
| **Date and Day**  **of Assignment** | | | Week5 - Monday | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicable to**  **Batches** | |  | | | |
| **AssignmentNumber: 9.1**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***Expected Time***  ***to complete*** |  |
|  | 1 | **Lab 9 – Code Review and Quality: Using AI to improve code quality and readability**  **Lab Objectives**   * Inline comments * Docstrings * Auto-documentation tools * AI-assisted summarization   **Task Description #1** (AI-Assisted Bug Detection)  **Scenario:** A junior developer wrote the following Python function to calculate factorials:  def factorial(n):  result = 1  for i in range(1, n):  result = result \* i  return result   * Run the code and test it with factorial(5)   (expected output = 120).   * Use AI (prompting) to review this code and identify the bug. * Ask AI to suggest corrections and rewrite the code. * Compare AI’s corrected code with your own fix.   **Manual code:**  IMG_256  **AI’s corrected code**  IMG_256  **Comparison:**   1. Both AI’s corrected version and fixed are **functionally identical**. 2. The only tiny difference is that AI used the shorthand result \*= i, while you used result = result \* i. Both are correct. 3. Both fixes solve the **off-by-one bug** in the loop.   **Task Description #2** (Improving Readability & Documentation)  **Scenario:** The following code works but is poorly written:  def calc(a,b,c):  if c=="add":  return a+b  elif c=="sub":  return a-b  elif c=="mul":  return a\*b  elif c=="div":  return a/b   * Use AI to review this code for readability, naming, and documentation issues. * Prompt AI to rewrite the code with: * Clear function & variable names. * Proper docstrings. * Exception handling for division by zero. * Compare the before-and-after versions to evaluate AI’s contribution.   **Prompt**  Correct the following code with functions and handling for division for zero by using proper docstring  **Before AI’s Contribution**  IMG_256  **After AI’s Contribution**  IMG_256  **Comparison:**  **Naming:**   1. Before: calc(a,b,c) → unclear. 2. After: calculate(a, b, operation) → descriptive.   **Documentation**   1. Before: No docstring. 2. After: Clear NumPy-style docstring with parameters, return type, and errors.   **Error Handling**   1. Before: No check for division by zero. 2. After: Raises ValueError if b == 0.   **Task Description #3** (Enforcing Coding Standards)  **Scenario:** A team project requires following PEP8 style guide. One developer submits:  def Checkprime(n):  for i in range(2,n):  if n%i==0:  return False  return True   * Run this code and verify correctness. * Use AI to perform a code quality review for PEP8 compliance. * Prompt AI to return a refactored version with proper indentation, spacing, and naming conventions. * Discuss how automated AI review can save time in large-scale projects.   **Prompt**  Generate the following code to print a refactored version with proper indentation,spacing and naming conventions  **Verification of code**  IMG_256  **AI refactored version**  IMG_256 ****Why AI Review Helps in Large Projects:**** **Consistency**: Ensures all team members follow the same coding style (PEP8).  **Time-Saving**: Automated reviews catch formatting, naming, and documentation issues instantly.  **Scalability**: In large projects with hundreds of files, AI tools prevent human reviewers from spending hours on style nitpicks.  **Quality Assurance**: Focus shifts from style to logic, improving productivity. **AI Code Quality Review :** :**Function naming**: Checkprime should be check\_prime (snake\_case).  **Indentation**: Missing 4-space indentation inside the function.  **Spacing**: Needs spaces around operators (%, ==, ,).  **Docstring**: Missing explanation of parameters, return value, and purpose.  **Edge case handling**: n <= 1 should return False.  **Task Description #4** (AI as a Code Reviewer in Real Projects)  **Scenario:** You are part of a GitHub project. A teammate submits this pull request:  def processData(d):  return [x\*2 for x in d if x%2==0]   * Review this function manually for readability, reusability, and edge cases. * Use AI to generate a code review comment, focusing on: * Naming conventions. * Input validation (e.g., what if d is not a list?). * Adding type hints. * Modify the function based on AI’s suggestions. * Write a short reflection: *Would you trust AI as a standalone reviewer, or only as a support tool? Why?*   **Modifying the function**  IMG_256  **Code Review:**  **Naming conventions**: Please use snake\_case (process\_data) to follow PEP8.  **Input validation**: Consider adding checks to ensure d is an iterable of numbers, otherwise raise a helpful error.  **Type hints**: Adding type hints (e.g., List[int] -> List[int]) improves clarity.  **Documentation**: A short docstring would make the function easier to understand for future contributors."  **Reflection**  No, not fully.   1. AI is excellent at catching **style issues, naming, documentation, and common pitfalls**. 2. But AI might miss **business logic errors**, **performance concerns**, or **project-specific requirements**. 3. Human reviewers bring **domain knowledge** and **context** that AI doesn’t always have | | | | | | Week5 - Monday |  |