**Name: Akshaya Neerati**

**Enrollment No: 2403A510B8**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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.   **Prompt:** Identify the error in the code.  **Code:**    **Observation:** When the given code was executed with input factorial(5), the output was 24 instead of the expected 120. This indicates a logical error in the loop range. The loop for i in range(1, n): only multiplies numbers from 1 to n-1, excluding n. Therefore, the calculation is incomplete, which leads to an incorrect factorial result.  **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:** Rewrite the code with clear function & variable names, proper docstrings and exception handling for division by zero.  **Code:**    **Observation:** The calculator function produces correct results but suffers from poor readability and maintainability. The function and variable names (calc, a, b, c) are not descriptive, and the absence of docstrings makes it hard to understand the purpose of the function. Additionally, there is no error handling for division by zero, which may lead to runtime crashes. After AI’s improvements, the code was rewritten with clear names, proper documentation, and exception handling, making it cleaner, more professional, and easier to use.  **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 re-factored version with proper indentation, spacing, and naming conventions. * Discuss how automated AI review can save time in large-scale projects.   **Prompt:** Return the code with a re-factored version with proper indentation, spacing, and naming conventions.  **Code:**    **Observation:** The submitted code correctly identifies prime numbers, but it does not follow PEP8 style guidelines regarding function naming, indentation, and spacing. The function name Checkprime should use snake\_case (check\_prime), and proper indentation and spacing must be applied to improve readability. After refactoring, the code becomes more structured, clear, and compliant with Python’s style conventions. Automated AI review ensures faster detection of such issues, saving time in large-scale projects by maintaining consistency, reducing manual review effort, and enhancing overall code quality.  **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.   Prompt: Generate a code review comment, focusing on: Naming conventions, input validation (e.g., what if d is not a list?), adding type hints.  Code:    **Observation:** The submitted function works correctly for doubling even numbers but lacks adherence to best practices. The function name processData should follow snake\_case (process\_data) for consistency with PEP8 naming conventions. Input validation is important to handle cases where the argument is not a list of integers, and adding type hints improves code clarity. The refactored version with proper docstrings, type hints, and error handling enhances readability, reusability, and robustness. Automated AI review helps in identifying such improvements quickly, ensuring higher quality and maintainable code in collaborative projects.   * Write a short reflection: *Would you trust AI as a standalone reviewer, or only as a support tool? Why?*   **ANS:**AI is best used as a **support tool** rather than a standalone reviewer. It can catch style and syntax issues quickly, but human reviewers are needed for logic, context, and project-specific requirements. | | | | | | Week5 - Monday |  |