|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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.   **Code#1:**  **Code before correcting:**    The Expected output is:120  Generated Output is:24  **Prompt#1:**  🡪My factorial function gives 24 for factorial(5) instead of 120.Please  fix the bug and give the corrected code.  **Code after correcting:**    **Observation#1:**  🡪Based on the code in the notebook, we started with a factorial function that incorrectly calculated the factorial of 5 as 24 due to an off-by-one error in the range of the for loop. We then corrected the function by changing the range to include n, and successfully tested it to confirm that it now correctly returns 120 for factorial(5).    **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.   **Code before modifying:**    **Prompt#2:**  **🡪**Review def calc(a, b, c): ... for readability, naming, and documentation issues, then rewrite it with clear names, a docstring, and division-by-zero handling, finally comparing both versions.  **Code after modifying:**    **Observation#2:**  **🡪**This code defines two calculator functions. calc is simple but lacks error checks. calculate\_operation is more robust with error handling. The output shows a successful multiplication using the second function.  **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.   **Code before verifying:**    **Prompt#3:**  **🡪** Review def Checkprime(n): ... for correctness and PEP8 compliance, then return a refactored version with proper naming, spacing, and indentation, and explain how AI reviews save time and improve consistency in large projects.  **Code after verifying:**    **Observation#3:**  **🡪** The original Checkprime(n) function works correctly but violates PEP8 style by using a non–snake\_case name, inconsistent indentation, missing spaces around operators, and lacking a docstring; the refactored version check\_prime(number) fixes these issues, adds clear documentation, handles invalid input dynamically, and overall demonstrates how AI-assisted reviews can quickly enforce coding standards, improve readability, and save developer time in larger projects.  **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?*   **Code before Reviewing:**      **Prompt#4:**  **🡪** Review the function def processData(d): ... for readability, reusability, and edge cases. Suggest improvements including proper naming conventions, type hints, input validation (e.g., if d is not a list), and a docstring. Then provide a refactored version of the function and explain your changes.  **Code after Reviewing:**    **Obsevation#4:**  **🡪** This code efficiently takes dynamic user input, validates it, and returns a list of doubled even numbers; it is compact, readable, handles invalid inputs gracefully, and clearly separates input processing from computation. | | | | | | Week5 - Monday |  |