|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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:**  **BASED ON GIVEN CODE,FINISH THE CODE.**    **GENERATE A CODE BY CORRECTING THE PREVIOUS CODE.**    **OBSERVATION:**  Both your fix and AI's suggestion align perfectly. The correction is simple but critical for accurate results. This small fix demonstrates how AI-assisted code review can catch subtle bugs, improve reliability, and reinforce good coding habits.  **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:**  **WITH HELP OF THE GIVEN PART OF CODE, FINISH THE REST OF THE CODE.**    **GENERATE A CODE FOR THE SIMPLE CALCULATOR.**    **OBSERVATION:**  **The original code for performing arithmetic operations lacked clarity, documentation, and error handling. After AI-assisted review, the function was rewritten with meaningful names, a detailed docstring, type hints, and safeguards like division-by-zero checks. This transformation not only improves readability and reliability but also aligns the code with professional standards, making it easier to maintain and scale.**  **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:**  **TAKE THE CODE AND LET DEBUG THE ERROR AND FINISH THE CODE.**    **GENERATE THE CODE AND GIVE PROPER OUTPUT**.    **OBSERVATION:**  **The original function was functional but lacked clarity and safeguards. After AI review, it was rewritten with meaningful names, clear documentation, and proper error handling—especially for division by zero. These improvements make the code easier to understand, safer to use, and more aligned with professional coding standards.**  **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?*   ***PROMPT:***  ***LOOK THE CODE AND FILL THE REST OF THE CODE.***    GENERATE THE CODE AND FINISH THE CODE USING PROPER PRESENATION.    OBSERVATION:  Sure! Here's another version of the summary using different wording:  The initial code performed basic arithmetic but lacked clarity and safeguards. After AI refinement, it was transformed with meaningful names, clear instructions through docstrings, and built-in error checks like division by zero. These enhancements make the function more readable, reliable, and suitable for use in professional or collaborative coding environments.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | | | | | | Week5 - Monday |  |