|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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:**    **Manual Code:**    **Prompt:**  Generate the code and identify the bug and suggest corrections.  **AI-Generated Code:**    **Comparison AI and manual code:**  1. Both use for i in range(1, n+1).  2. Manual code prints an error message; AI code raises a ValueError.  3. Manual code is beginner-friendly; AI code is more descriptive with comments.  **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 and variable names and use proper docstrings and exception handling for division by zero.  **Before Versions:**    **After AI versions:**    **Comparison between Before and After Versions:**  1.**Naming:**   * **Before:** Function name calc and parameters a, b, c → vague and non-descriptive. * **After:** Function name calculate and parameters num1, num2, operation → clear, descriptive   **2.Documentation:**   * **Before:** No docstring or comments. * **After:** Added detailed docstring → explains function purpose, parameters, return value.   3.**Error Handling:**   * **Before:** No error handling.. * **After:**Robust error handling.   + Raises ValueError for unsupported operations.   **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:**  Convert the following code to give a refactored version with proper indentation, spacing, and naming conventions.  **Code:**    **Refactored version:**    **How AI Review Saves Time in Large-Scale Projects:**  1**.Consistency**: Enforces uniform coding standards across teams.  2.**Reduced Manual Effort:** AI handles trivial fixes, humans focus on logic/design.  3.**Scalability**: Efficiently reviews hundreds of pull requests daily.  4.**KnowledgeSharing:** Helps junior developers learn best practices.  **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:**  Review the following Python function for readability, naming conventions, reusability.  **AI Modified Function:**    **Code Review:**  **1.Naming Conventions:** The function name processData should be renamed to process\_data.  **2.InputValidation**: The function should check that the input is a list of integers, and show a clear error if it is not.  **3.Type hints:** Add type hints so it’s clear what the function takes and returns.  **Short reflection:**  I would use AI mainly as a **support tool**, not as a standalone reviewer. AI is very helpful for catching common issues like naming, formatting, missing type hints, and simple bugs, which saves a lot of time in large projects. | | | | | | Week5 - Monday |  |