|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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:10.1**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***Expected Time***  ***to complete*** |  |
|  | 1 | **Lab 10 – Code Review and Quality: Using AI to Improve Code Quality and Readability**  **Lab Objectives**   * Use AI for automated code review and quality enhancement. * Identify and fix syntax, logical, performance, and security issues in Python code. * Improve readability and maintainability through structured refactoring and comments. * Apply prompt engineering for targeted improvements. * Evaluate AI-generated suggestions against PEP 8 standards and software engineering best practices   **Task Description #1 – Syntax and Logic Errors**  Task: Use AI to identify and fix syntax and logic errors in a faulty Python script.  Sample Input Code:  # Calculate average score of a student  def calc\_average(marks):  total = 0  for m in marks:  total += m  average = total / len(marks)  return avrage # Typo here  marks = [85, 90, 78, 92]  print("Average Score is ", calc\_average(marks)  Expected Output:  Corrected and runnable Python code with explanations of the fixes. PROMPT:CORRECT THE GIVEN CODE  EXPLAINATION: This code calculates and prints the average score of a student based on a list of marks. The function [calc\_average](vscode-file://vscode-app/c:/Users/ashwi/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html" \o ") takes a list called [marks](vscode-file://vscode-app/c:/Users/ashwi/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) as its parameter. Inside the function, it initializes a variable [total](vscode-file://vscode-app/c:/Users/ashwi/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) to zero, then iterates through each mark in the list, adding each value to [total](vscode-file://vscode-app/c:/Users/ashwi/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html). After the loop, it computes the average by  dividing the total sum by the number of marks, which is obtained using the built-in [len()](vscode-file://vscode-app/c:/Users/ashwi/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html" \o ") function. The function then returns the calculated average.  **Task Description #2 – PEP 8 Compliance**  Task: Use AI to refactor Python code to follow PEP 8 style guidelines.  Sample Input Code:  def area\_of\_rect(L,B):return L\*B  print(area\_of\_rect(10,20))  Expected Output:  Well-formatted PEP 8-compliant Python code. PROMPT: REFACTOR THE PYHTON CODE USING GOOGLE STYLE DOC STRING   OUTPUT:  EXPLANATION: This code defines a Python function named [area\_of\_rect](vscode-file://vscode-app/c:/Users/ashwi/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html" \o ") that calculates the area of a rectangle. The function takes two parameters: [L](vscode-file://vscode-app/c:/Users/ashwi/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html), representing the length of the rectangle, and [B](vscode-file://vscode-app/c:/Users/ashwi/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html), representing the breadth (or width). Both parameters are expected to be floating-point numbers, as indicated in the function's docstring.  Inside the function, the area is computed by multiplying the length ([L](vscode-file://vscode-app/c:/Users/ashwi/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)) by the breadth ([B](vscode-file://vscode-app/c:/Users/ashwi/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)). The result of this multiplication is then returned as the output of the function. The docstring provides a clear description of the function's purpose, its arguments, and its return value, which helps other programmers understand how to use the function correctly  **Task Description #3 – Readability Enhancement**  Task: Use AI to make code more readable without changing its logic**.**  Sample Input Code:  def c(x,y):  return x\*y/100  a=200  b=15  print(c(a,b))  Expected Output:   * Python code with descriptive variable names, inline comments, and clear formatting.   PROMPT:GENERATE CODE WITH DESCRIPTIVE ANALYSIS  INLINE COMMENTS AND OUTPUT    EXPLANATION: The code defines a function named [calculate\_percentage](vscode-file://vscode-app/c:/Users/ashwi/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html" \o ") that takes two arguments: [amount](vscode-file://vscode-app/c:/Users/ashwi/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) and [percentage](vscode-file://vscode-app/c:/Users/ashwi/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html). This function calculates what a given percentage of an amount is by multiplying the amount by the percentage and then dividing by 100. For example, to find 15% of 200, the function multiplies 200 by 15 and divides the result by 100, yielding 30.0.  **Task Description #4 – Refactoring for Maintainability**  Task: Use AI to break repetitive or long code into reusable functions.  Sample Input Code:  students = ["Alice", "Bob", "Charlie"]  print("Welcome", students[0])  print("Welcome", students[1])  print("Welcome", students[2])  Expected Output:   * Modular code with reusable functions.   PROMPT:GENERATE PYTHON CODE WITH REUSABLE FUNCTION INLINE COMMENTS AND THE OUTPUT    **Task Description #5 – Performance Optimization**  Task: Use AI to make the code run faster.  Sample Input Code:  # Find squares of numbers  nums = [i for i in range(1,1000000)]  squares = []  for n in nums:  squares.append(n\*\*2)  print(len(squares))  Expected Output:   * Optimized code using list comprehensions or vectorized operations.   PROMPT:REFACTOR THE CODE BELOW WITH LIST COMPRHENSION OF FOR LOOP  OUTPUT: EXPLANATION: This code measures how long it takes to create a list of squares for all integers from 1 up to, but not including, 1,000,000. It begins by importing the [time](vscode-file://vscode-app/c:/Users/ashwi/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) module, which provides functions for tracking time. The variable [start\_time](vscode-file://vscode-app/c:/Users/ashwi/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html" \o ") records the current time at the beginning of the program  **Task Description #6 – Complexity Reduction**  Task: Use AI to simplify overly complex logic.  Sample Input Code:  def grade(score):  if score >= 90:  return "A"  else:  if score >= 80:  return "B"  else:  if score >= 70:  return "C"  else:  if score >= 60:  return "D"  else:  return "F"  Expected Output:   * Cleaner logic using elif or dictionary mapping   PROMPT:CORRECT THE CODE FOR GETTING THE OUPUT    EXPLANATION:  This Python code defines a function called [grade](vscode-file://vscode-app/c:/Users/ashwi/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html) that takes a numerical score as input and returns a corresponding letter grade based on standard grading thresholds. The function uses a series of if, elif, and else statements to check the value of the input score. If the score is 90 or above, it returns "A". If the score is between 80 and 89, it returns "B". Scores between 70 and 79 receive a "C", and those between 60 and 69 get a "D". Any score below 60 results in an "F". | | | | | | Week5 - Monday |  |