| **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/III | **Regulation** | | R24 | | | |
| **Date and Day**  **of Assignment** | | | Week3 – Monday | **BATCHNO:06** | | **NAME:J.ABHIRAM**  **H.NO:2403A51342** | | | |
| **Duration** | | | 2 Hours | **Applicable to**  **Batches** | |  | | | |
| **Assignment Number:6.1**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***Expected Time***  ***to complete*** |  |
|  | 1 | Lab 6: AI-Based Code Completion – Classes, Loops, and Conditionals  **Lab Objectives:**   * To explore AI-powered auto-completion features for core Python constructs. * To analyze how AI suggests logic for class definitions, loops, and conditionals. * To evaluate the completeness and correctness of code generated by AI assistants.   **Lab Outcomes (LOs):**  After completing this lab, students will be able to:   * Use AI tools to generate and complete class definitions and methods. * Understand and assess AI-suggested loops for iterative tasks. * Generate conditional statements through prompt-driven suggestions. * Critically evaluate AI-assisted code for correctness and clarity.   Task Description #1 (Classes – Employee Management)   * Task: Use AI to create an Employee class with attributes (name, id, salary) and a method to calculate yearly salary. * Instructions:   + Prompt AI to generate the Employee class.   + Analyze the generated code for correctness and structure.   + Ask AI to add a method to give a bonus and recalculate salary.   **PROMPT:**  Generate a Python class named Employee with attributes name, id, and salary. Add a method to calculate the yearly salary.  **CODE:**      **OUTPUT:**    Task Description #2 (Loops – Automorphic Numbers in a Range)   * Task: Prompt AI to generate a function that displays all Automorphic numbers between 1 and 1000 using a for loop. * Instructions:   + Get AI-generated code to list Automorphic numbers using a for loop.   + Analyze the correctness and efficiency of the generated logic.   + Ask AI to regenerate using a while loop and compare both implementations.   **PROMPT:**  Generate a Python function that displays all Automorphic numbers between 1 and 1000 using a for loop.  **CODE:**      **OUTPUT:**    Task Description #3 (Conditional Statements – Online Shopping Feedback Classification)   * Task: Ask AI to write nested if-elif-else conditions to classify online shopping feedback as Positive, Neutral, or Negative based on a numerical rating (1–5). * Instructions:   + Generate initial code using nested if-elif-else.   + Analyze correctness and readability.   + Ask AI to rewrite using dictionary-based or match-case structure.   **PROMPT:**  Write Python code using nested if-elif-else statements to classify online shopping feedback as Positive, Neutral, or Negative based on a numerical rating from 1 to 5.  **CODE:**        **OUTPUT:**      Task Description #4 (Loops – Prime Numbers in a Range)   * Task: Generate a function using AI that displays all prime numbers within a user-specified range (e.g., 1 to 500). * Instructions:   + Get AI-generated code to list all primes using a for loop.   + Analyze the correctness and efficiency of the prime-checking logic.   + Ask AI to regenerate an optimized version (e.g., using the square root method).   **PROMPT:**  Write a Python function that displays all prime numbers within a user-specified range (for example, 1 to 500) using a for loop.  **CODE:**      **OUTPUT:**    Task Description #5 (Classes – Library System)   * Task: Use AI to build a Library class with methods to add\_book(), issue\_book(), and display\_books(). * Instructions:   + Generate Library class code using AI.   + Analyze if methods handle edge cases (e.g., issuing unavailable books).   + Ask AI to add comments and documentation.   **PROMPT:**  Generate a Python class named Library with methods add\_book(), issue\_book(), and display\_books()  **CODE:**        **OUTPUT:**    Note: Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots  **Evaluation Criteria:**   | **Criteria** | **Max Marks** | | --- | --- | | Class | 1.0 | | Loops | 1.0 | | Conditional Statements | 0.5 | | **Total** | **2.5 Marks** | | | | | | | Week3 - Monday |  |