|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE** | | | | | **DEPARTMENT OF COMPUTER SCIENCE ENGINEERING** | | | | |
| **ProgramName:**B. Tech | | | | **Assignment Type: Lab** | | | **AcademicYear:**2025-2026 | | |
| **CourseCoordinatorName** | | | | 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) | | | | | | |
| **CourseCode** | | | 24CS002PC215 | **CourseTitle** | | AI Assisted Coding | | | |
| **Year/Sem** | | | II/I | **Regulation** | | R24 | | | |
| **Date and Day**  **of Assignment** | | | Week4 - Wednesday | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicableto**  **Batches** | |  | | | |
| **AssignmentNumber:7.3**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***ExpectedTime***  ***to complete*** |  |
|  | 1 | Lab 7: AI-Error Debugging with AI: Systematic approaches to finding and fixing bugs  **Lab Objectives:**   * To identify and correct syntax, logic, and runtime errors in Python programs using AI tools. * To understand common programming bugs and AI-assisted debugging suggestions. * To evaluate how AI explains, detects, and fixes different types of coding errors. * To build confidence in using AI to perform structured debugging practices.   **Lab Outcomes (LOs):**  After completing this lab, students will be able to:   * Use AI tools to detect and correct syntax, logic, and runtime errors. * Interpret AI-suggested bug fixes and explanations. * Apply systematic debugging strategies supported by AI-generated insights. * Refactor buggy code using responsible and reliable programming patterns.   **Task Description#1**   * Paste a function with a missing colon (add(a, b)), and let AI fix the syntax error.      * Prompt: Paste a function with a missing colon (add(a, b)), and let AI fix the syntax error.     **Expected Output#1**   * Corrected function with syntax fix   **Through this task, we understood how minor syntax mistakes, such as a missing colon (:), can prevent a Python program from executing. AI tools efficiently identified the missing symbol and corrected it, enhancing our ability to read and interpret syntax-related error messages more accurately.**  **Syntax Error – Missing Colon**  **Buggy Code:**  **def add(a, b) # No colon → SyntaxError**  **AI Fix:**  **def add(a, b): # Colon added**  **Output:**  **5 + 3 = 8**  **Observation:**  **AI quickly fixed a common syntax error. This helps beginners avoid frustration.**  **Task Description#2 (Loops)**   * Identify and fix a logic error in a loop that causes infinite iteration.       **With manual coding**    **Expected Output#2**   * AI fixes increment/decrement error     Logic Error in Loops  This task demonstrated the significance of logical flow in programming. An infinite loop caused by a missing increment or decrement statement was detected and corrected using AI assistance. We learned how to design proper loop conditions and maintain control flow to avoid unintended infinite iterations.  **Task Description#3**   * Debug a runtime error caused by division by zero. Let AI insert try-except.     **Expected Output#3**  **Traceback (most recent call last):**  **File "lab7\_3.py", line XX, in <module>**  **print(10 / 0)**  **Program CRASHES Terminal stops with RED ERROR No output after that**  **~~^~~**  **ZeroDivisionError: division by zero**   * Corrected function with safe error handling       **This exercise highlighted the importance of handling runtime exceptions gracefully. The AI tool identified the division-by-zero issue and suggested implementing a try-except block for error handling. This approach taught us how to prevent program crashes and write more stable and user-friendly code.**  **Task Description#4**   * Provide a faulty class definition (missing self in parameters). Let AI fix it     **Expected eroor**  **TypeError: Student.\_\_init\_\_() takes 2 positional arguments but 3 were given**  **Expected Output#4**   * Correct \_\_init\_\_() method and explanation         **Reflection / Conclusion:**  **In Task 4, we learned how a small mistake like missing self in a class can cause big errors in Python. This task helped us understand the importance of self in connecting variables and methods to a particular object. By using AI, we were able to find and fix the error easily and also understand why it happened. This improved our confidence in debugging object-oriented programs and showed how AI can make coding and learning faster and smar**  **Task Description#5**  **Prompt: "Fix index error: lst = [1,2,3] print(lst[5])**   * Access an invalid list index and use AI to resolve the Index Error.     **Expected Output#5**  **Expected error : IndexError: list index out of range**     * AI suggests checking length or using safe access logic     **Out put :**    **This task involved accessing an invalid list index, resulting in an IndexError. AI provided a robust solution by suggesting list length checks or try-except blocks for safe access. We learned how to write defensive code that anticipates and prevents such runtime errors.**  **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** | | --- | --- | | Identification of bugs | 0.5 | | Application of AI-suggested fixes | 0.5 | | Explanation and understanding of errors | 0.5 | | Corrected code functionality | 0.5 | | Report structure and reflection | 0.5 | | **Total** | **2.5 Marks** | | | | | | | Week4 - Wednesday |  |