(By Tech Involvers)

Project 5: Quiz Game

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Instructions:**

* Read the problem carefully before trying to solve it.
* Do the tasks on your own. Don’t copy it.
* The output of your program must be the same as given in the sample run.

#### **Overview:**

A Quiz Game is an interactive application where users answer questions to test their knowledge on various topics. The game can present multiple-choice questions, track the player's score, and provide feedback on their performance.

#### **How to Implement**

##### **1. Game Loop**

* The game runs in a loop, presenting questions to the user and capturing their answers.
* After each question, the game checks the answer, updates the score, and provides feedback.

##### **2. Game State Management**

* The game state includes the current question, the player's score, and the total number of questions.

##### **3. Rendering**

* Use a text-based interface (console) or a graphical user interface (GUI) to display questions and capture user input.

##### **4. User Input**

* Capture user input for selecting answers to the questions.

##### **5. Question Management**

* Store questions in a data structure (e.g., a list or array) with their possible answers and the correct answer.
* Load questions from a file or database for flexibility.

##### **6. Scoring and Feedback**

* Keep track of the player's score based on correct answers.
* Provide feedback after each question and at the end of the quiz.

##### **7. End of Game**

* Display the final score and feedback on the player's performance.
* Offer an option to restart the quiz or exit the game.

#### **Steps to Implement**

1. **Design the Questions**
   * Prepare a set of questions with multiple-choice answers.
   * Store the questions in a structured format (e.g., JSON file, CSV file).
2. **Initialize the Game**
   * Load the questions from the file into the program.
   * Initialize variables for tracking the current question, score, and total number of questions.
3. **Create the Game Loop**
   * Display each question with possible answers.
   * Capture user input for the selected answer.
   * Check if the answer is correct and update the score.
   * Provide feedback on the answer.
4. **Display the Final Score**
   * After all questions are answered, display the player's final score and feedback.
   * Offer options to restart the quiz or exit the game.