Weekly Progress Report

Name: Diya Gupta
Domain: Python
Date of submission: 21-05-2025

Week Ending: 04

I. Overview

This week, I worked on developing a command-line-based Quiz Game using Python. The project focused on reading structured data from a JSON file, processing multiple-choice questions, capturing user inputs, and evaluating quiz results interactively.

II. Achievements

1. Quiz Game Development

- **Project Description: Created a console-based quiz application that dynamically loads questions from a JSON file and evaluates user performance.**
- **Key Features Implemented:**
  - **Parsed questions and options from a structured JSON file using json.load().**
  - **Displayed multiple-choice questions and captured user answers.**
  - **Implemented scoring logic and final result display.**
- **Technologies Used: Python, JSON, File I/O.**

2. File Handling Proficiency

- **Gained hands-on experience with file reading operations and working with external .json data files.**
- **Ensured error-free reading and parsing of structured content.**

3. Code Structuring and Logic Flow

- **Designed clean control flow for question rendering and answer validation.**

- **Used enumerate() to keep track of question numbers for better user experience.**

III. Challenges

1. Input Handling Robustness

- **Encountered issues with inconsistent user inputs (e.g., lowercase vs uppercase answers).**
- **Addressed this by standardizing input to uppercase and stripping whitespace.**

2. JSON Structure Consistency

- **Faced problems due to improperly formatted JSON initially.**
- **Ensured a consistent structure with fields: "question", "options", and "answer" for smooth execution.**

IV. Learning Resources

1. Python Documentation

- **Referred to Python's official documentation for json module and input handling best practices.**

2. Community Support

- **Reviewed similar quiz game examples on platforms like GitHub and Stack Overflow for inspiration and troubleshooting.**

V. Next Week's Goals

1. Feature Expansion

- **Add support for categories or difficulty levels within the quiz.**
- **Implement a timer or countdown feature for each question to increase engagement.**

2. User Interface Improvements

- **Explore ways to add a graphical interface using tkinter or PyQt for better usability.**

3. Persistent Scoreboard

- **Add functionality to store quiz scores and player names in a file or database for future reference.**

VI. Additional Comments

Working on this project was both educational and enjoyable. It helped reinforce core programming concepts like loops, conditionals, user interaction, and file processing. The interactive nature of the quiz makes it a great tool for both learning and entertainment. Looking forward to enhancing the project with more features and interactivity in the coming weeks.