

# Documentation for the Quiz Game Script

## 1. Introduction

The provided Python script is an interactive quiz game developed for students aged 13-19. Designed as a fun educational tool, the game tests players' general knowledge through a series of five random questions. Players earn points for correct answers, which are stored in a cumulative score table for future reference.

---

## 2. Features

- **Player Eligibility:** Only players between the ages of 13 and 19 are allowed to participate.
  - **Randomized Questions:** A pool of diverse questions ensures variability in gameplay.
  - **Scoring System:** Each correct answer earns 20 points, with a maximum score of 100 points.
  - **Data Persistence:** Player details and scores are stored in a `GAME_SUMMARY.csv` file for record-keeping.
  - **Leaderboard:** Displays sorted results by points and announces the top scorer(s).
  - **Friendly Interface:** Clear prompts guide the player through the game process.
- 

## 3. Implementation Details

### 3.1 Game Flow

1. **Welcome Screen:** Displays rules and eligibility criteria.
2. **Player Registration:** Collects player information, including Index No, Name, and Age.
3. **Eligibility Check:** Verifies if the player is a teenager. Non-teenagers cannot proceed.
4. **Question Rounds:**
  - Five questions are chosen randomly from a predefined list.
  - The player must answer each question correctly to proceed to the next.
5. **Score Calculation:** Updates points for each correct answer.
6. **Data Storage:** Saves player data into a CSV file, appending new records.
7. **Summary Display:** Optionally shows all players' scores, sorted leaderboard, and top scorer(s).

### 3.2 Key Code Sections

- **Random Question Selection:**  
Utilizes the `random.choice` function to avoid repeated questions in a single game session.

```
ran_question = random.choice(question_list)
```

- **Data Management:**

Stores player details and scores using `pandas.DataFrame` and saves them in a CSV file.

```
data = pd.concat([data, pd.DataFrame({...})])
data.to_csv("GAME_SUMMARY.csv", index=False, mode="a",
header=(round_count == 1))
```

- **Error Handling:**

Ensures robust input validation for Index No and Age.

```
try:
    indexno = int(input("Enter Your Index No* : ").strip())
except ValueError:
    indexno = int(input("--Index number should be a number--\n\nEnter
Your Index No* : ").strip())
```

---

## 4. Usage

### Running the Script

1. Install Python and ensure `pandas` is installed (`pip install pandas`).
2. Run the script using a terminal or IDE.

```
python OG.py
```

### Interpreting Results

- Players are guided through registration and gameplay via prompts.
  - After gameplay, a CSV file (`GAME_SUMMARY.csv`) logs all players' information.
  - If requested, the leaderboard and top scorer(s) are displayed.
- 

## 5. Limitations and Notes

- **Limited Question Pool:** With repeated plays, players may encounter the same questions. Expanding the pool is recommended.
  - **No GUI:** The script runs in the console; integrating a graphical interface could enhance usability.
  - **Score Reset:** Players can replay the game, but scores from previous sessions remain in the summary file.
  - **Error Handling Improvements:** Ensure stricter validation for non-numeric inputs.
-