**Project: Quiz app using Tkinter**

**Problem Statement**: Create a quiz game with multiple-choice questions. Keep track of the player's score and provide feedback on their performance.

**1. Introduction:**

The "Quiz App using Tkinter" project aims to create an interactive quiz application in Python using the Tkinter library. The application presents users with multiple-choice questions, allowing them to select answers and receive immediate feedback. The primary goal is to demonstrate the use of Tkinter for creating a graphical user interface (GUI) and implement a basic quiz functionality.

**2. Project Overview:**

The project consists of a Tkinter-based GUI that displays questions with multiple-choice options. Users can select an answer, receive instant feedback, and proceed to the next question. The quiz data, including questions, choices, and correct answers, is stored in a separate module (quiz\_data.py).

**3. Code Overview:**

The code is organized into functions that handle different aspects of the quiz app. Key functions include:

* show\_question(): Displays the current question and choices.
* check\_answer(choice): Checks the user's selected answer and updates the score.
* next\_question(): Moves to the next question or ends the quiz.

**4. Tkinter GUI Design:**

The Tkinter GUI is designed with a clear and user-friendly layout. The primary components include a question label, multiple choice buttons, a feedback label, a score label, and a "Next" button. The styling is done using the ttkbootstrap library to enhance the visual appeal.

**5. Quiz Data:**

The quiz data is stored in the quiz\_data list within the quiz\_data.py module. Each entry in the list represents a question with the question itself, a list of choices, and the correct answer.

**6. Functionality:**

Upon launching the app, the first question is displayed. Users select an answer, and the app provides immediate feedback—correct or incorrect. The score is updated accordingly. After answering all questions, a messagebox displays the final score.

**7. Results and Feedback:**

The app provides instant feedback after each user response, enhancing the interactive learning experience. The final score is displayed in a messagebox at the end of the quiz.

**8. Challenges and Solutions:**

No significant challenges were encountered during the development of this simple quiz app. The project focused on fundamental Tkinter concepts and basic Python functionality.

**9. Conclusion:**

The quiz app successfully demonstrates the integration of Tkinter for GUI development in a Python application. It serves as an introductory project for those learning GUI programming with Tkinter.

**10. Future Enhancements:**

Potential enhancements include adding more diverse question types, incorporating a timer, and implementing a user login system. These improvements could make the app more engaging and feature-rich.

**11. References:**

Tkinter Documentation: <https://docs.python.org/3/library/tkinter.html>

ttkbootstrap Documentation: <https://ttkbootstrap.readthedocs.io/>

OUTPUT:



