**Project title: Quiz**

Project reference(reference\_code): https://www.geeksforgeeks.org/python-mcq-quiz-game-using-tkinter/

**-Introduction:**

In the modern digital era, educational tools and platforms play an essential role in improving learning experiences.

The purpose of this report is to outline the development and implementation of a multiple-choice quiz project that incorporates a timer,

next button, quit button, and view correct answer button. The project aims to provide an interactive and engaging quiz experience

for users while measuring their knowledge on a specific topic. The inclusion of these features enhances user engagement, provides flexibility

in quiz progression, and offers the option to review correct answers for users who got 1 or more wrong answer(s), promoting a personalized

learning experience.

**-Project ideas and value-added:**

The idea behind the quiz project is to create an educational tool that promotes learning through interactive quizzes.

By incorporating a timer, next button, quit button, and view correct answer button, the project offers several advantages.

Firstly, the timer adds a sense of urgency, encouraging participants to answer questions quickly and improving their time management skills.

Secondly, the next button empowers users to control the pace of the quiz, allowing them to review their answers before proceeding.

The quit button provides an option for users to exit the quiz at any point.

Finally, the view correct answer button allows users who got 1 or more wrong answer(s) to review the correct answers, facilitating learning from their mistakes.

**-Similar or Related Works:**

Several projects, apps, and web applications exist that incorporate multiple-choice quizzes and similar features.

Some popular examples include educational platforms like Quizlet, Kahoot, and Quizizz, which offer timed quizzes with options to proceed, quit,

and review correct answers. These platforms provide users with an interactive learning experience, allowing them to test their knowledge and track their progress.

The inclusion of timers and review options has been shown to enhance user engagement and learning outcomes.

**-SDLC Phases, Algorithms, Pseudocode, Flowchart, etc.:**

The development of the multiple-choice quiz project follows the Software Development Life Cycle (SDLC).

The phases include requirement gathering, design, implementation, testing, and deployment.

During the requirement-gathering phase, the specific features and functionality of the quiz project,

such as the timer, next button, quit button, and view correct answer button, are defined.

The design phase involves creating wireframes, user interface designs, and the overall architecture of the project.

The implementation phase includes writing the code for the quiz application. Algorithms are utilized to manage the timer, track the user's progress,

validate user answers, calculate scores, and determine eligibility to view correct answers.

Pseudocode and flowcharts are used to plan and organize the logic of the quiz application, ensuring smooth execution and user-friendly interactions.

**-System Features and Functions:**

The quiz project incorporates several key features and functions to provide an effective user experience. These include:

1. Homepage: The homepage serves as the entry point to the quiz project. It provides an overview of the quiz topic and instructions,

and there are 3 types of quiz (math with logic test, general, and history) for users to choose by clicking the button indicating what type of quiz to start the quiz.

In addition, all three types of this quiz are all implemented with the same features, however, the questions, answers, and answer options are not the same

and they are stored in a JSON file.

2. Multiple-Choice Questions: The quiz project utilizes multiple-choice questions, where users can select one option from a set of predefined choices.

3. Shuffle question: This function makes the questions pop up in a randomized order every time the users close and open the quiz application.

4. Timer: The timer displays the remaining time for each question, adding a time constraint and increasing the challenge for participants.

5. Quit Button: The quit button provides users with the option to exit the quiz at any point, allowing them to discontinue the quiz session if needed.

6. View Correct Answer Button: The view correct answer button is only accessible to users who got 1 or more wrong answer(s) and only if the result is shown(the quiz ended).

Users will see another window pop up which contains the questions and their answers. It provides them with the opportunity to review the correct answers,

facilitating learning from their mistakes.

7. Next Button: The next button allows users to proceed to the next question only if they choose their answer when the time is not up.

The user's answer is being testified when clicking this button or not clicking as long as users select their answers(When time is up,

answers will be testified as long as they did select their answer.). If it's correct we add 1 point to the user's score. Also, users will get a message pop-up saying

that the user's answer is correct if the user's selected answer option is indeed correct, otherwise, the message will be user's answer is incorrect.

If users try to click the next button while they do not select any answer and the time is not up, they will get a warning telling them to select their answer.

If users do not select their answer and the time is up, they will get a message pop-up saying that they did not select their answer and move on to the next question.

After finishing testifying user's answer and it's now the last question, users will get a message pop up displaying their results with percentage,

their number of right and wrong answers.

8. Scoring System: The project implements a scoring system that calculates and displays the user's score based on their answers.

Each correct answer contributes to the total score.

**-Before Modification vs After Modification**

+ Before Modification:

\* A result message at the end of the quiz, Next button, Quit button, questions are always in the same order even when we tried doing a second time,

no message if the answer selected is right or wrong

+ After Modification:

\* A result message at the end of the quiz, Have homepage, Modified Next button, Quit button, questions are always in a randomized order every time users try doing it again, view correct answer button, a timer for every question, a message if the answer selected is right or wrong

**-Code Implementation:**

The multiple-choice quiz project is implemented using the Python programming language.

Python provides a wide range of tools and libraries that can be utilized to enhance the functionality of the quiz application.

The code includes functions and classes to handle user interactions, manage timers, retrieve questions, validate user answers,

calculate scores, control the progression to the next question, and determine eligibility to view correct answers.

The code is structured and organized to ensure readability, maintainability, and modularity.

**-Code testing/debugging**

Thorough testing and debugging are essential to ensure the functionality and reliability of the multiple-choice quiz project.

The Python code is tested by running various scenarios, including answering questions within and not within the time limit, leaving the option not selected,

using the next button, quitting the quiz, and accessing the view correct answer button under the specified conditions.

Bugs and issues are identified and resolved through careful debugging, ensuring smooth operation and a seamless user experience.

**-System result and evaluation:**

The success of the multiple-choice quiz project can be evaluated based on several factors,

including user engagement, completion rates, user feedback, and learning outcomes.

Metrics such as average time taken to complete the quiz, average scores, and the utilization of the view correct answer button can provide insights into

user performance and the effectiveness of the features. User feedback and assessment of learning outcomes can be collected to gauge the project's impact

on knowledge acquisition and user satisfaction.

**Conclusion and Future Development:**

In conclusion, the multiple-choice quiz project implemented in Python offers an interactive and engaging quiz experience for users. Python's simplicity,

versatility and wide range of libraries and tools contribute to the flexibility and functionality of the project. The inclusion of features such as timers,

next button, quit button and view correct answer button enhance user engagement, promote time management skills, and facilitate learning from mistakes.

Thorough implementation, testing, and evaluation ensure the functionality and reliability of the project.

For future development, some possible ideas include:

1. User Authentication: Adding user authentication features to enable personalized quiz experiences and user profiles.

2. Difficulty Levels and Categories: Introducing different difficulty levels and categories for quizzes, allowing users to choose the level and topic that suits

their knowledge and learning goals.

3. Analytics and Reporting: Incorporating analytics and reporting features to provide detailed insights into user performance, progress, and areas for improvement.

4. Multimedia Support: Enhancing the quiz application to support multimedia content, such as images, videos, or audio files in questions,

providing a more engaging and diverse experience.