Online quiz system

Navya Sree-24KB1A05FK

Deekshitha-24KB1A05FP

Sruthi-24KB1A05FX

Nasreen-24KB1A05JA

CSE – Department

Date: 01-05-2025

Acknowledgement:

I would like to express my gratitude to my instructor for their guidance and support throughout this project. I also thank my peers for their valuable feedback and my family for their constant encouragement.

Abstract:

This project is a simple multiple-choice quiz game written in C. It allows users to answer questions, calculates their score, and saves it to a file. A timer feature can be enabled to end the quiz after a set duration. The program also displays a list of high scores from previous players.

Introduction:

The idea behind this project is to create an interactive quiz game to help users test their knowledge. It offers multiple-choice

questions, a timer, and a score tracker. The project provides practice with file handling, user input, and basic programming logic. The motivation was

to create an engaging program for learning C programming concepts, focusing on file handling and time management.

Objectives:

- Create an interactive multiple-choice quiz game.
- Implement a timer feature to limit quiz duration.
- Track and save high scores in a file.
- Provide an engaging and competitive quiz experience.

System Requirements:

Software:

- C programming language
- Compiler: GCC, Code::Blocks, or Turbo C++

Hardware:

- Processor: Intel Core i3 or equivalent
- RAM: 2 GB minimum
- Storage: 100 MB free space
- Display: 1024x768 resolution.

Methodology

1. **Requirement Gathering:** Identified core functionalities: questions, scoring, timer, and file handling.
2. **Design:** Planned the structure of the program, including user interaction and file management.
3. **Implementation:** Developed functions for loading questions, calculating scores, handling user input, and saving high scores.

4. **Testing:** Ensured the program works as expected, especially the timer and file handling features.
5. **Optimization:** Improved efficiency and input validation.

Project Description

Problem Statement: Interactive quiz games are an excellent tool for learning. The challenge was to create a quiz game that includes scoring, file storage, and time limits.

Proposed Solution: A C program that presents multiple-choice questions, calculates the score, saves the score to a file, and allows users to view high scores. A timer can be enabled to limit quiz duration.

Key Features:

- Multiple-choice questions with four options.
- Timer option for time-limited quizzes.
- Score calculation and storage in a high score file.
- Display of high scores after each quiz.

Flow chart:

Start Program

|

Enter Player Name

|

Enable Timer (Yes/No)?

|

Start Timer (if enabled)

|

Display Question 1

|

Get User Answer

|

Check Answer (Correct/Incorrect)

|

Next Question?

|

Timer Expired?

|

Calculate Score

|

Save High Score

|

Display High Scores

|

End Program

Program code:

https://onlinegdb.com/Gkw6cEKJS

Testing and validation:

Testing Process:

Functional Testing:

- Tested user input for name and timer choice.
- Verified question display and correct answer validation.
- Checked timer functionality (if enabled) and ensured quiz stops after 60 seconds.
- Ensured high scores are saved and displayed correctly.

Edge Case Testing:

- Ensured invalid answers (non-A/B/C/D) prompt re-entry.
- Handled missing high score file gracefully.

Validation:

- The program produces expected results, handles errors well, and performs efficiently.

Limitations (Disadvantages):

1. **Fixed Number of Questions**: The program only supports 3 questions, limiting scalability.
2. **No Difficulty Levels**: All questions are of equal difficulty, with no options for varying difficulty.
3. **Limited Timer Functionality**: The timer applies only to the whole quiz, not individual questions.
4. **Basic Input Validation**: The program doesn't handle special characters in names.
5. **No High Score Management**: High scores are appended without sorting or limiting entries.

Future Enhancements:

1. **Dynamic Question Set**: Allow loading questions from external sources (e.g., file or database).
2. **Multiple Difficulty Levels**: Add easy, medium, and hard difficulty settings for varied user experience.
3. **Timer per Question**: Implement a separate timer for each question.
4. **Improved User Feedback**: Provide feedback immediately after each question (e.g., "Correct!" or "Incorrect!").
5. **High Score Sorting**: Sort high scores by highest to lowest and manage the number of entries.
6. **Better Input Validation**: Validate player names to avoid invalid characters or spaces.

Conclusion:

The program successfully implements a multiple-choice quiz with timer functionality and high score management. It demonstrates basic C concepts such as file handling, structs, arrays, and time management.

Learning Outcomes:

- Gained experience in file handling, using structs and arrays.
- Learned to implement time-based logic in C.
- Understood the importance of input validation for user inputs.

References:

- **C Programming Language**: Brian W. Kernighan, Dennis M. Ritchie, *The C Programming Language*, 2nd Edition.
- **C Standard Library**: GNU C Library Documentation.
- **Timer in C**: GeeksforGeeks - time() function.