ONLINE QUIZ SYSTEM WITH TIMER

Group :- 25

JULY 9, 2025

Hemant Kumar

Submitted To :- Dr. Shudha Shankar Prashad / Dr. Amrindar Kaur

Online Quiz System with Timer

1. Introduction

The "Online Quiz System with Timer" is an interactive command-line application written in C++, featuring:

- A multiple-choice quiz interface.
- A countdown timer for each question.
- Real-time score tracking.
- Immediate feedback on answers.
- End-of-quiz score summary.

2. Objectives

- Implement real-time user interaction.
- Reinforce C++ fundamentals: structures, vectors, loops, I/O, and thread management.
- Introduce timer functionality to simulate time-bound quizzes.
- Modular design for possible future extensions.

3. Functional Requirements

- Load questions: Hardcoded list of quiz questions, options, and answers.
- Display timer: 15 seconds allotted for each question.
- Capture user input: Detect key press without waiting (non-blocking).
- Validate answers: Compare user selection with correct option.
- Provide feedback: Show correct/incorrect immediately after each answer.
- Show results: Display total score at the end.

4. System Design

4.1 Data Structures

```
struct Question
{
   string questionText;
   vector<string> options;
   char correctOption;
};
```

4.2 Key Functions

askQuestionWithTimer(const Question& q, int timeLimitSeconds)

- Displays question + options.

- Waits up to timeLimitSeconds for user input using _kbhit() and _getch().
- Implements delay using std::this_thread::sleep_for.
- Returns user’s choice or blank if timed out.

4.3 Main Flow

1. Load quiz data into vector<Question>.
2. Initialize score = 0.
3. Loop through each Question:
   - Present question via askQuestionWithTimer.
   - Validate and update score.
4. After all questions, output a summary.

5. Implementation

Full code is as follows:

```
#include <iostream>
#include <string>
#include <vector>
#include <thread>
#include <chrono>
#include <conio.h>

using namespace std;

struct Question {
    string questionText;
    vector<string> options;
    char correctOption;
};

char askQuestionWithTimer(const Question& q, int timeLimitSeconds) {
    cout << q.questionText << "\n";
    for (int i = 0; i < q.options.size(); i++)
        cout << char('A' + i) << ". " << q.options[i] << "\n";

    cout << "You have " << timeLimitSeconds << " seconds to answer...\n";
    char answer = ' ';
    bool answered = false;

    for (int i = 0; i < timeLimitSeconds * 10; i++) {
        if (_kbhit()) {
            answer = toupper(_getch());
            answered = true;
            break;
        }
        std::this_thread::sleep_for(std::chrono::milliseconds(100));
    }

    if (!answered)
        cout << "Time's up! Moving to next question.\n";
    return answer;
}

int main() {
    vector<Question> quiz = {
        {"What is the capital of France?",
         {"Berlin", "Madrid", "Paris", "Rome"},
         'C'},
        {"Which planet is known as the Red Planet?",
         {"Earth", "Mars", "Jupiter", "Venus"},
         'B'},
        {"Who wrote 'Romeo and Juliet'?",
         {"Shakespeare", "Dickens", "Wordsworth", "Poe"},
         'A'},
        {"What is 9 x 9?", {"81", "72", "99", "108"}, 'A'}
    };

    int score = 0;
    int total = quiz.size();
    int timePerQuestion = 15;

    cout << "==== Welcome to Online Quiz System ====\n";

    for (int i = 0; i < total; i++) {
        cout << "\nQuestion " << i + 1 << ":\n";
        char userAnswer = askQuestionWithTimer(quiz[i], timePerQuestion);
        if (userAnswer == quiz[i].correctOption) {
            cout << "Correct!\n";
            score++;
        } else if (userAnswer != ' ') {
            cout << "Wrong! Correct: " << quiz[i].correctOption << "\n";
        }
    }

    cout << "\nQuiz Over! Your Score: " << score << " out of " << total << "\n";
    return 0;
}
```

6. Compilation & Execution

On Windows (with <conio.h> support):

```
g++ -std=c++11 quiz.cpp -o quiz
quiz.exe
```

On Linux/macOS: Replace _kbhit()/_getch() logic with termios, or remove timer for compatibility.

7. Testing

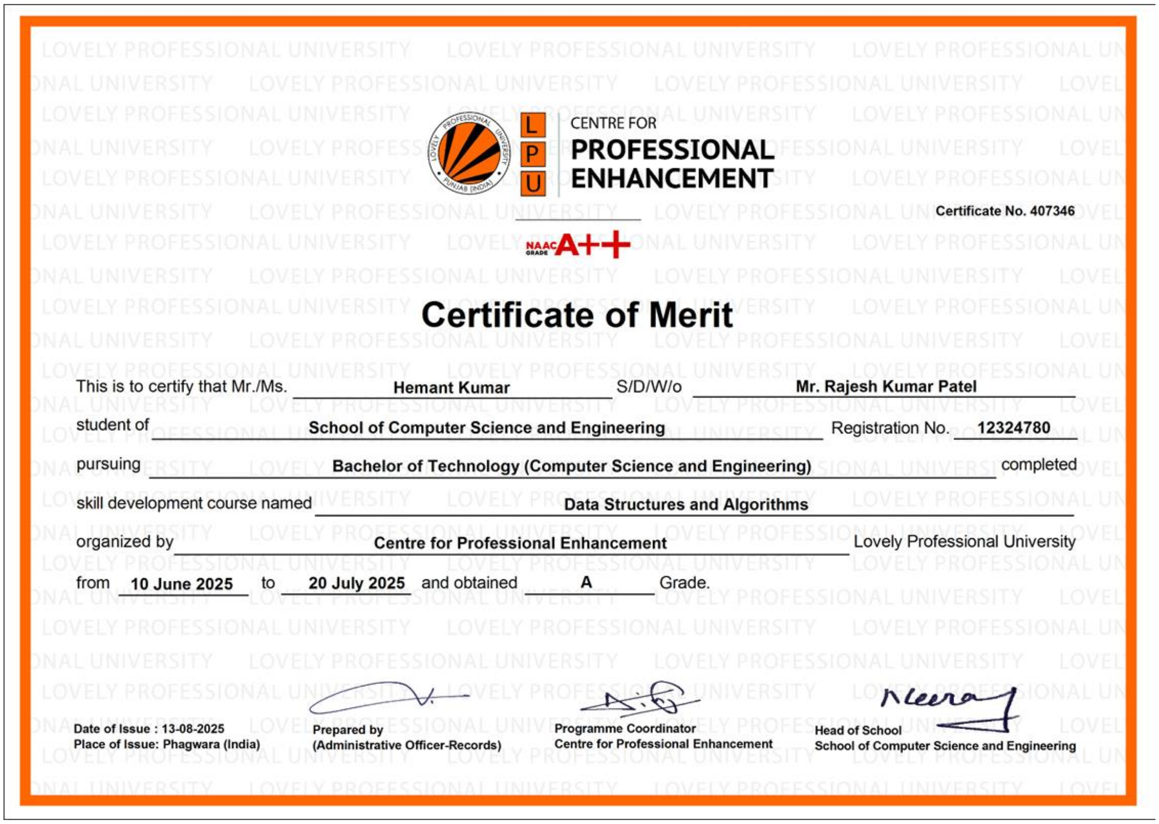
- Scenario 1: Answer within time → verify correct scoring and output.
- Scenario 2: No response → ensure timeout behavior triggers.
- Scenario 3: Incorrect answer → confirm feedback with correct option.
- Scenario 4: Mixed answers → final score reflects correct totals.

8. Future Enhancements

- Load questions dynamically from a file.
- Store user scores in an external file or database.
- Add user authentication/login.
- Improve UI by using GUI libraries.
- Allow different quiz modes and timer settings.

9. Conclusion

This project demonstrates core C++ skills: data management, loops, user input, and time control. The timer-based quiz enhances interactivity and encourages timely responses. The modular design supports easy extensions and future improvements.



CENTRE FOR
PROFESSIONAL
ENHANCEMENT

NAAC GRADE A++

Certificate No. 407346

Certificate of Merit

This is to certify that Mr./Ms. **Hemant Kumar** S/D/W/o **Mr. Rajesh Kumar Patel**
student of **School of Computer Science and Engineering** Registration No. **12324780**
pursuing **Bachelor of Technology (Computer Science and Engineering)** completed
skill development course named **Data Structures and Algorithms**
organized by **Centre for Professional Enhancement** Lovely Professional University
from **10 June 2025** to **20 July 2025** and obtained **A** Grade.

Date of Issue : 13-08-2025
Place of Issue: Phagwara (India)

Prepared by
(Administrative Officer-Records)

Programme Coordinator
Centre for Professional Enhancement

Neera
Head of School
School of Computer Science and Engineering