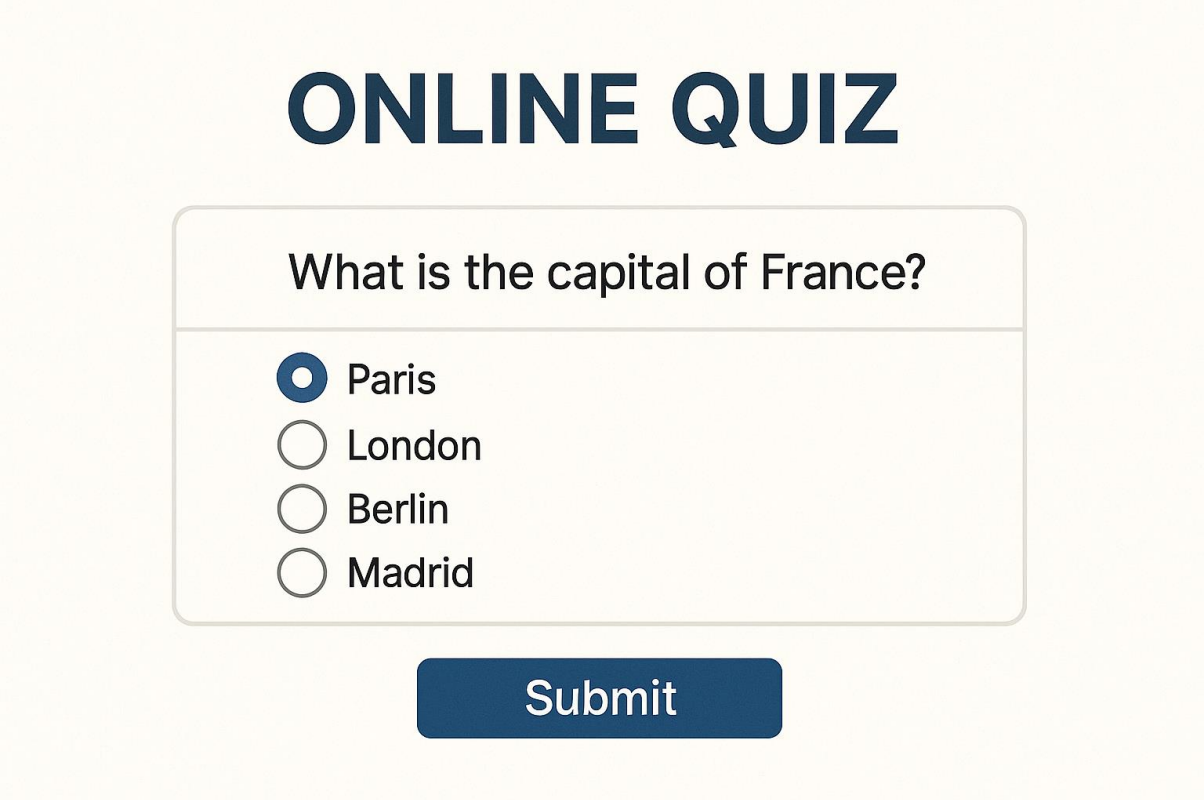
Project Report – Online Quiz Application in C

Submitted By:

Shivansh Singh
SAP ID: 590027276

Submitted To:

Ms. Tanu Singh
Faculty, School of Computer Science



2. Abstract

This project is a simple menu-based Online Quiz Application made in C.
The program lets the user add multiple-choice questions, display them, and attempt a quiz.
All questions are saved in a file, so they remain available even after restarting the program.

The project uses concepts like structures, arrays, strings, functions, and file handling.
The objective was to build a working application that covers all major topics of the course.

---

3. Problem Definition

The aim of this project is to create a system where:

- A user can add MCQ-type questions
- Each question has 4 options and 1 correct answer
- The user can view all stored questions
- A quiz can be conducted and the user gets a score
- All questions should be saved permanently in a file

Inputs: question text, options, correct answer, quiz responses
Outputs: list of questions, correct/incorrect messages, final score

---

4. System Design

4.1 Modules Used

1. **Initialization Module**
   Loads existing questions from a text file.
2. **Question Module**
   Adding a question and displaying questions.
3. **Quiz Module**
   Asking questions, checking answers, and calculating score.
4. **File Handling Module**
   Saving and loading questions from the file.

4.2 Data Structure

A structure was used to store each question:

```
struct Question {
    char questionText[100];
    char options[4][50];
    int correctOption;
};
```

4.3 Algorithms

Add Question Algorithm

1. Ask user to enter the question text.
2. Ask for 4 options.
3. Ask for correct option number (1–4).
4. Store details in the question array.
5. Increase question count.
6. Save to file.

Display Questions Algorithm

1. If no questions → show message.
2. Otherwise print each question, its options, and the correct answer.

Conduct Quiz Algorithm

1. Set score = 0.
2. For each question:
    - Show the question
    - Ask for answer
    - Compare with correct option
    - Increase score if correct
3. Show final score.

Save to File

Write question count, then write every question and option line by line.

Load from File

Read question count, then read all saved questions into the array.

5. Implementation Details

- Language: C
- Compiler: GCC
- Concepts used: structures, arrays, functions, strings, loops, file handling
- Project files are organized as required by the major project guidelines:
    - `src/` → main.c, quiz.c
    - `include/` → quiz.h
    - `docs/` → report
    - `assets/` → screenshots

The main menu includes:

1. Add Question
2. Display Questions
3. Conduct Quiz
4. Exit

---

6. Testing & Results

Test Cases Performed

1. **Add Question**
   Entered a sample question and options → question stored correctly.
2. **Display Questions**
   All added questions appeared with correct formatting.
3. **Correct Answer Test**
   Selected correct option → program showed "Correct!".
4. **Wrong Answer Test**
   Selected wrong option → program showed the right answer.
5. **File Saving Test**
   Restarted the program → previously added questions were loaded.

Screenshots

(Screenshots of menu, adding a question, and quiz output should be placed here.)

---

7. Conclusion & Future Work

Conclusion

The project meets all the requirements of the major project.
It shows how C programming can be used to build a small working application using structures, functions, and file handling. The program runs correctly and stores all data permanently.

Future Work

If extended, the project can include:

- User login
- Difficulty levels
- Timed quizzes
- Question categories
- Score history and ranking

---

8. References

- Let Us C – Yashwant Kanetkar
- Programming in ANSI C – E. Balagurusamy
- Classroom notes and slides
- GCC documentation