



Interactive Quiz

Project Report



Md. Riaz
CSE 16th Batch, 4th Semester
0322310105101024
Web Engineering (CSE 2201)

Introduction

The Interactive Quiz Application allows users to participate in an interactive quiz session with real-time feedback and score tracking. The application is designed to provide a seamless and engaging learning experience by dynamically loading questions from an external API and providing instant validation of user responses.

Related Works

Several quiz applications, such as **Kahoot**, **Quizlet**, and **Socrative**, offer interactive learning environments. However, our application focuses on real-time quiz generation with unique visual effects using libraries like **Thanos.js** for enhanced engagement. Compared to traditional quiz apps, this project emphasizes simplicity and dynamic question delivery.

Methodology

Technology Stack:

- **Frontend:** HTML, CSS, JavaScript, jQuery
- **Backend:** PHP
- **Database:** Session-based data storage
- **External API:** Open Trivia Database (for fetching questions)
- **Third-Party Libraries:** Thanos.js, html2canvas.js

Development Process:

1. **Design Phase:** Created wireframes for the user interface.
2. **Implementation Phase:** Developed the frontend and backend, integrating AJAX for dynamic interactions.
3. **Testing Phase:** Tested with sample questions and various user scenarios to ensure accuracy and stability.

Architecture Overview

The application follows a simple client-server model where:

1. **Client-Side:** Handles user interaction, dynamic UI updates, and AJAX requests.
2. **Server-Side:** Manages session data, question fetching, answer validation, and response preparation.

Flow Diagram

1. **Login:** Username entry initiates the quiz session.
2. **Question Display:** Dynamically loaded from the API.
3. **Answer Submission:** Evaluated via AJAX requests.
4. **Feedback:** Immediate display of results per question.
5. **Quiz Completion:** Final score and performance summary presented.

Outcome

- **User Authentication:** Simple session-based username handling.
- **Dynamic Question Loading:** Questions fetched from the Open Trivia Database API.
- **Real-Time Feedback:** Immediate correctness feedback on each answer.
- **Score Tracking:** Ongoing score updates displayed in real-time.
- **Visual Effects:** Integration of Thanos.js for a "disintegration" effect on incorrect answers.
- **Logout Option:** Allows users to quit and reset the quiz.

Conclusion & Future Work

The Interactive Quiz Application demonstrates how a simple yet effective quiz platform can enhance learning. For future development,

User Authentication System:

To allow users to log in and save quiz progress or historical performance.

Timer for Questions:

Adding a countdown timer for each question to increase the challenge.

Leaderboard Implementation:

Displaying top scores to encourage user competition.

Multilingual Support:

Adding support for multiple languages to expand accessibility.

This project serves as a testament to the practical skills acquired in web development and highlights the importance of combining technical proficiency with creative design.

Supervisor

Nahid Hasan

(Lecturer, CES, PUB)