Technical Report: Interactive Video Quiz Program

Prepared by: Vedanth Adityan & Bhaavesh Satish
Date: 29/10/2025

Page 1: Introduction

The interactive video quiz program was designed to provide an engaging, multimedia-based learning experience for users. The project integrates short educational or illustrative videos with multiple-choice questions, allowing users to interact with the content while learning or testing knowledge. The program aims to combine the visual and cognitive aspects of learning by presenting video material and immediately testing comprehension through interactive questions.

One of the primary objectives of this project was to create a **user-friendly and visually appealing interface**. Users can watch videos, answer questions, receive instant feedback, and see their overall performance at the end of the quiz. The integration of videos into the program introduces an additional layer of complexity because the program must handle media content alongside dynamically changing questions and options.

Another objective was to ensure the **technical robustness and accessibility** of the application. Videos needed to be embedded in a way that allowed playback across different browsers without requiring users to manually upload files. To achieve this, the team explored embedding techniques using streaming platforms, allowing the program to load videos dynamically from the internet.

This project also had educational and technical aims. Educationally, it encourages users to actively engage with content by combining video observation with critical thinking to answer questions. Technically, it allowed the team to practice **front-end programming skills**, including **HTML for structure**, **CSS for styling and layout**, and **JavaScript for dynamic content, interactivity, and logic handling**.

Finally, the project reflects a focus on **interactivity and feedback mechanisms**. When users select answers, the program highlights correct and incorrect choices immediately. This feature enhances the learning experience by providing instant reinforcement, a principle commonly used in educational software to improve retention and engagement.

Through this project, the team aimed to explore the intersection of multimedia learning, user interface design, and interactive programming logic, producing a tool that is both functional and educational. The subsequent sections of this report will discuss the system design, program logic, technical challenges, lessons learned, and conclusions, providing a comprehensive view of the project from conception to completion.

Page 2: System Design

The design of the interactive video quiz program was based on a **single-page web application architecture**, utilizing only client-side technologies. The choice of a single-page layout was deliberate to ensure seamless transitions between video questions and to avoid page reloads, which could disrupt the user experience.

2.1 Components

The program is divided into three primary components:

1. **Video Display Section** – This section embeds the video associated with each question. Initially, a standard HTML <video> element was considered, but this required local video files or user uploads, which was not user-friendly. To overcome this, the program uses **iframe embedding** from the Streamable platform, allowing videos to be streamed directly from the web. The iframe dynamically changes for each question, ensuring that the correct video is displayed at the appropriate time.
2. **Question Section** – Positioned adjacent to the video, this section displays the question and multiple-choice options. A **flexbox layout** ensures that the video is aligned on the left and the question/options on the right, providing a clean and intuitive interface. The question container also includes buttons for each option, which are styled for easy readability and clickability.
3. **Control & Result Section** – This section includes the "Next" button to move through questions and the final results page. After all questions are answered, the program calculates the score and displays it along with a thank-you message. This separation of concerns simplifies both the design and the logic implementation.

2.2 Data Structure

The quiz content is stored in a **quizData array**, where each element is an object containing the video URL, question text, multiple-choice options, and the index of the correct answer. This structure allows the program to dynamically load any number of questions and corresponding videos without hardcoding individual elements into the HTML.

2.3 Layout and Responsiveness

The design emphasizes **responsive design principles**. By using flexbox and relative widths, the program adjusts the layout for various screen sizes. On larger screens, the video and questions are displayed side by side, while on smaller screens, they stack vertically for better readability. This ensures accessibility across desktops, laptops, and tablets.

The system design demonstrates a careful balance between **user experience, technical feasibility, and modular programming**, setting the stage for effective implementation and testing.

Page 3: Program Logic

The program logic for the interactive video quiz revolves around **dynamic content loading, user interaction handling, and score calculation**. A well-defined logic flow ensures that videos, questions, and answers are synchronized seamlessly.

3.1 Dynamic Video and Question Loading

At the heart of the program is the quizData array, which contains objects with video URLs, question text, options, and the index of the correct answer. The program uses JavaScript to **dynamically inject the video iframe** and question elements into the DOM for each quiz step. This eliminates the need to hardcode multiple HTML pages, allowing for easy expansion of the quiz.

```
videoContainer.innerHTML = `<iframe src="${q.video}" ...></iframe>`;
```

This dynamic approach allows each question to load with its associated video without reloading the page.

3.2 User Interaction Handling

Each multiple-choice option is a button element. When the user clicks a button:

1. All buttons are disabled to prevent multiple selections.
2. Correct answers are highlighted in green, and incorrect selections are highlighted in red.
3. The “Next” button becomes visible, allowing the user to proceed to the next question.

This logic provides **immediate feedback**, reinforcing learning through real-time validation of user answers.

3.3 Navigation and Score Calculation

The program maintains a currentQuestion index to track progress. After each answer selection, the “Next” button increments the index. Once all questions are answered, the program calculates the total score by counting correct selections and displays it on a **results page**.

```
scoreEl.textContent = `You got ${score} out of ${quizData.length} correct.`;
```

3.4 Flexibility and Scalability

The logic is designed for scalability. New videos and questions can be added simply by appending objects to quizData. The dynamic loading mechanism automatically handles these additions without requiring additional code modifications.

By separating **data, layout, and logic**, the program achieves modularity, making it easier to maintain and extend. The system ensures a **consistent, interactive experience** across all questions and video content.

Page 4: Technical Challenges

During the development of the interactive video quiz program, the team encountered several technical challenges that required creative problem-solving. One of the most significant obstacles was the **integration of video content**. Initially, the team considered using local MP4 files embedded in the HTML. However, this approach proved impractical because it required users to upload videos manually or download large files, creating a cumbersome experience. Moreover, different browsers had varying levels of support for video formats, which further complicated local file handling.

After exploring multiple solutions, the team attempted to use **free video hosting platforms** such as Vimeo, YouTube, and Streamable. While YouTube offered high reliability, it introduced challenges with autoplay, ads, and branding overlays that disrupted the quiz experience. Vimeo's free tier had strict upload limits and required additional account management for each video, which made it less suitable for seamless integration. Streamable offered a simpler embed interface but also imposed limitations on the number of videos and playback options.

Finally, the team discovered **MindVideo.ai**, a platform that allowed free hosting, embedding, and smooth playback with minimal restrictions. Using MindVideo.ai, the program could dynamically load video content via iframe embeds without requiring the user to upload anything. This discovery was pivotal, as it resolved most of the video integration problems and allowed the program to maintain a **clean, uninterrupted user experience**.

Other challenges included **layout and responsiveness**. The team struggled to arrange the video and question containers side by side while ensuring readability on smaller screens. Flexbox was chosen as the solution, but careful testing was required to handle edge cases on different screen sizes and resolutions.

Finally, implementing **immediate feedback** for user answers required managing CSS classes dynamically. The team had to ensure that correct and incorrect answers were highlighted appropriately while disabling other buttons to prevent multiple selections.

Overall, these challenges emphasized the importance of **platform selection, user experience considerations, and modular programming**, which were critical in delivering a robust, functional program.

Page 5: Lessons Learned

Developing the interactive video quiz program was a significant learning experience. The team gained both technical knowledge and insights into **project planning, problem-solving, and user-centered design**.

One of the most valuable lessons was the importance of **choosing the right platform for hosting video content**. Initially, multiple options were considered, including local hosting and popular streaming services like YouTube and Vimeo. Each presented trade-offs related to accessibility, restrictions, branding, and playback reliability. After extensive testing, MindVideo.ai was chosen because it offered seamless embedding, smooth playback, and a user-friendly interface. This experience highlighted the need to evaluate **technical feasibility alongside user experience** when selecting third-party services.

Another key lesson was the significance of **dynamic, modular programming**. By structuring the quiz around a quizData array and dynamically injecting HTML elements, the program could handle multiple questions and videos without repeating code. This approach reinforced best practices in **separation of concerns**, making it easier to update the program, add new questions, or change layout and styling without extensive rewrites.

The team also improved their **front-end development skills**, including flexbox layout, responsive design, and interactive JavaScript. Building the side-by-side video and question layout, while maintaining usability across screen sizes, required iterative testing and adjustment of relative widths and padding. Additionally, implementing **real-time feedback** for answers taught the team how to manipulate CSS classes dynamically and handle event listeners efficiently.

Finally, the project reinforced **soft skills**, including collaborative problem-solving, time management, and presentation skills. During the final presentation, the team discussed the challenges faced, decisions made, and solutions implemented. Explaining the technical process to peers improved communication and reinforced understanding of the program’s design and logic.

Overall, the project provided a comprehensive learning experience, combining technical skill development, user-centered design, and reflective problem-solving.

Page 6: Conclusion

The interactive video quiz program successfully achieved its goals of providing a **functional, educational, and user-friendly tool** for learning through multimedia content. By integrating videos with interactive multiple-choice questions, the program offers a dynamic and engaging experience that reinforces knowledge through immediate feedback.

The project demonstrated the **importance of thoughtful system design**. Using a single-page application structure, the team ensured that videos and questions could be displayed seamlessly without page reloads, providing smooth transitions between quiz steps. The use of a quizData array and dynamic DOM injection allowed scalability, enabling the addition of more videos and questions with minimal changes to the codebase.

The program also highlighted the value of **responsive and flexible layouts**. Through the use of flexbox, the video is displayed on the left and the question/options on the right, creating an intuitive interface for desktop users. The design remains adaptable for smaller screens, ensuring accessibility and usability across multiple devices.

Technical challenges, such as selecting a suitable video hosting platform, embedding videos, and providing real-time feedback, were successfully overcome. MindVideo.ai proved to be an effective solution for hosting and embedding videos, while CSS and JavaScript provided robust mechanisms for interaction and visual feedback. These solutions strengthened the team's **problem-solving skills** and understanding of front-end development best practices.

In conclusion, the project was a **successful combination of technical implementation and educational design**. The team gained valuable experience in programming, multimedia integration, and interface design. Lessons learned from this project can be applied to future interactive learning tools, demonstrating the importance of **modularity, user experience, and careful selection of third-party services**.

The program can be expanded further with additional features such as more questions, auto-play between videos, progress tracking, and enhanced mobile optimization, making it a **versatile template for interactive multimedia quizzes**. Overall, the project reflects both the technical and educational growth of the team and serves as a practical example of combining video content with interactive learning methods.