

Project Report: Video Uploader Web Application

1. Introduction

This report presents the development of the "Video Uploader" web application, a frontend project built with React and Vite. The aim was to design a lightweight video management tool that supports uploading and previewing videos in the browser.

2. Background

As modern web platforms increasingly rely on multimedia content, understanding how to process and preview media on the frontend is crucial. This project provided the opportunity to explore file handling, component reuse, and third-party media tools within the React ecosystem.

3. Learning Objectives

- Learn to set up and configure Vite with React - Implement video upload and playback using HTML5 APIs - Integrate external packages like react-player - Enhance UI/UX using basic CSS styling and layout techniques - Develop functional, responsive components and manage state effectively

4. Activities and Tasks

- Initialized the project using Vite's React template - Built a file input component to handle video uploads - Integrated the `react-player` package for in-browser video playback - Created a layout with CSS for responsive UI - Conducted browser testing and UI refinement

5. Skills and Competencies

Technical Skills: - React fundamentals including hooks, props, and state - Vite setup and build tools - File handling in JavaScript - Media playback integration - Basic CSS styling
Soft Skills: - Time management and self-discipline - Critical thinking in resolving frontend challenges - Improving code readability and modularization

6. Feedback and Evidence

The application was demonstrated and reviewed by peers. It received positive feedback on UI

Project Report: Video Uploader Web Application

simplicity and responsive behavior. Screenshots of functional components and successful test runs were documented. Suggestions for improvement included adding file format validation and upload progress indicators.

7. Challenges and Solutions

One challenge was managing memory when previewing large video files. This was resolved by using object URLs and placing limits on file size. Another issue was with customizing react-player's behavior, which was addressed by closely following the library's documentation and community examples.

8. Outcomes and Impact

This project improved the understanding of real-world component design and file-based input handling in React. It also helped reinforce best practices for development efficiency using Vite, laying a strong foundation for future multimedia and UI-heavy web applications.

9. Conclusion

The Video Uploader project offered practical exposure to modern web development techniques. It enhanced both technical and analytical skills while demonstrating how React applications can effectively handle interactive tasks like video upload and playback. The lessons from this project will inform future frontend development endeavors.