Project Proposal for 3. Video Streaming Using Cloud Technology

Team Number: 3

Team Members: Seth Akins 80868318, Erin Hiebert 21754635, Kiet Phan 95769543, Teresa Saller 56256902, Justin Schoenit 44680874

1. Overview

The purpose of our software is to allow users to easily send and receive videos through their web browser. Users will be able to protect their privacy in those videos by optionally blurring their faces or their background. These privacy protection features will be implemented using AWS services. Our software will be a web app that is targeted at professional settings which require video submissions, such as recruiting or education. We are hoping to make it easier for our target demographic to send and receive videos, all while maintaining privacy by giving users the ability to blur their faces in recordings before submitting. What sets our solution apart from others is the amount of control users will have over their videos and video requests.

Users will be able to either request videos or submit a video to a request. In order to request videos, they would open a “submission box,” where they can request videos from other users by entering their email addresses. These users would then receive an email notification of the submission request. To submit a video to the “submission box,” users would log into the system, where they could then record the video in the browser, or upload a pre-recorded video from their computer. The video would then be processed in our web app. For the processing, we will take advantage of different AWS APIs to enable e.g. face blurring on video recordings. Users will be able to play back their videos after processing, and once they’re satisfied, they can submit them. Once submitted, videos can still be retracted, and users can also set an expiry date on videos before submission.

All of these features aim to put as much power as possible into the user’s hands, all while making the workflow as easy as possible. Ultimately, we will help professionals easily send and receive videos while protecting their privacy through the use of cloud technology.

1.1 Envisioned Usage

Receiver:

The user is a teacher who has an account with the website. They want to have their students submit short videos in which they read a paragraph in Spanish. Since they already have an account, they navigate to the website and log in to see their dashboard. From the dashboard, they create a new submission box. They set the title of the submission box, and

invite their students to submit their videos by entering their email addresses. A few hours later, they receive a notification that a video has been submitted. They log into the website again to review the first submission. The student made a few pronunciation mistakes, so the teacher leaves targeted comments to correct their mistakes. Once the teacher is done with their review, they submit their comments, so that their student can review them.

Sender:

The user is a college student who receives an email with a link to our website inviting them to submit a video to their teacher's submission box. When they click on the link, they are prompted to log in, and since it's their first time using the website, they must create an account. After creating an account, they can access the submission request on the website's main dashboard, and click on it to start the submission process. They get the option of uploading a video to the website from their computer, recording directly in the browser, or adding a video they previously recorded on the site. They can then review the video directly on the page, decide whether to blur their face or background and add comments before submitting it. Once they've completed the submission, the video is still visible on their dashboard but shows as submitted. They also have the option to retract their submission or resubmit the video until the submission box closes, after which they can only retract the submission.

2 Major Milestones

Deadline	Deliverable
Term 1 week 9: Mini Presentation Oct 31	1. Project and Environment Setup (includes dockerization, setting up CI + linting, database setup) 2. User account creation, login & logout For this milestone we are going to have our project fully set up and dockerized. We will also be implementing account creation, login & logout. Additionally, we will take some time for UI design (mocks). On top of that, we will all need time to get used to our new tech stack, so we are planning in some time for doing tutorials
Term 1 week 13: Design submission Nov 28	1. Establish a process for sending videos, including dashboard page and setup basic page structure for video submission. 2. Uploading videos to AWS S3 and retrieving them from AWS S3 3. Previewing video in the browser: nice video playing with keyboard shortcuts, setting speed, full-screen mode etc 4. Dashboard Page for receiving videos (see all received videos)

	5. Creating submission boxes (date opened, max time video will be stored, name of submission box, date it will close, who videos are requested from, max video length) 6. Email verification for users (set up mail client) 7. Video processing features (API calls to AWS), e.g. face blurring and background blurring 8. User Interface Design 9. Design document creation 10. Video creation During this milestone, we will get the basic features of our website (video sending and receiving) working. We will also create our Design document and create a video showcasing our work.
Term 2 week 4: Peer Testing Jan 30	1. Testing (refactor, test coverage, E2E, integration testing) 2. Documentation 3. Video processing features (API calls to AWS) 4. Filter/search for videos by title, date, and person that submitted 5. General comments on video submissions 6. Invite links for submission boxes (including emails) 7. Modifying submission boxes (update/delete), and allowing multiple submissions to the same box 8. Dashboard Page for recently viewed/sent videos, and relevant submission boxes 9. Dashboard Page to see all videos the user has posted (a library of their own videos) 10. Email notifications: reminders for submission boxes (if they have been opened or close soon), submission sent/received, and comments received This milestone will expand on our base features and add some more advanced things like email notifications and the ability to leave comments on received videos.
Term 2 week 8: Peer Testing Mar 27	1. Address peer testing debugging #1 2. Address peer testing debugging #2 3. Testing (refactor, test coverage, E2E, integration testing) 4. Time-stamped comments on videos, and export comments (as text files) 5. Multiple people managing one submission box 6. Deleting recorded video (retracting submission) & recovering deleted video (with grace period) 7. Documentation 8. Tagging and advanced filtering for videos

	9. Custom batch notification to users of something by email (ex: let people who passed an interview know) 10. Deployment During this milestone, we will continue implementing more advanced features and once again we are planning in some time to address our peer testing feedback. Additionally, we will take some time to perform UI improvements.
Term 2 week 13: Final project submission	1. Address peer testing debugging 2. Testing (refactor, test coverage, E2E) 3. Customize dashboard, pin, set favourites, hide (submissions, submission boxes), background 4. UI improvements 5. Create Project Demo Video For the final milestone, we will be taking some additional time for testing and documentation. We are also hoping to have our project deployed.

3 Technology Stack

Tech	**Reasoning**
Computer (Laptop or Desktop)	- Our target demographic is average adults who have access to computers - Videos will likely be recorded at home and not on the go
Web browser	- Easily access with any operating system - Will optimize for Chrome (>60% market share worldwide)
NextJS (with Typescript)	- Can be used to create full-stack web app: No separate backend, everything is TypeScript (easier to learn) and provides type safety on top of JavaScript - Popular in industry, meaning there will be lots of documentation and StackOverflow posts to draw on and learn from - Uses React for the front end, which is the industry standard for client-side interfaces - Support for server-side rendering so pages render quicker - Automatically caches data that is requested using the JavaScript fetch() method - Has built-in routing to handle navigation between pages

	- Comes bundled with all the testing frameworks we need
Docker	- Enables easy deployment by building a container image and then publishing it on a number of platforms (such as AWS) - Multiple different operating systems in the team, using docker will streamline set-up and get rid of operating system-dependent problems
Prisma (ORM)	- Data models are very readable, it has TypeScript support, and easy migrations to update the database - Supports a range of databases - Has PrismaStudio to manipulate data in a visual format - Enables us to save time and not worry about writing SQL
Material UI (Component Library)	- Has Figma library that helps design faster, components from Figma can be imported in NextJS (ease of use) - There is a wide range of components that are already ready for use in websites, so minimal component design time is required - There are website-wide themes you can customize that adapt all components in Material UI to the theme you specify
Figma (Design)	- Interfaces nicely has collaboration features - Can be used for high-fidelity mocks with actual functionality (e.g. navigation)
Rest API	- Industry-standard for APIs
PostgreSQL (Database)	- Free & open source - Our data will be structured and therefore be easy to store within tables
SASS (UI Design)	- Support style nesting, variables, and mixin - Makes CSS more modular and reusable, but still compiles to vanilla CSS
ChatGPT, GitHub Copilot	- Accelerate writing boilerplate - Help teach unfamiliar technologies
AWS S3	- For video storage

	- Will allow storage of videos in a file system instead of a database so we do not have to deal with transforming videos between blob and mp4/mov
AWS APIs	- For video processing (e.g. blurring faces) - Project requirement
Git (version control)	- Industry-standard, required for this project
GitHub (collaboration)	- Industry-standard, required for this project
GitHub Actions (CI)	- Built into GitHub - Allows us to integrate a CI/CD pipeline into our project so testing and linting are run before merging, and our project is deployed each time a new PR is merged
WebStorm (IDE)	- All of us using the same IDE will make it easier to help each other and troubleshoot issues - Allows for easy pair programming through CodeWithMe

4 Teamwork Distribution and Anticipated Hurdles

Category	Seth	Erin	K	Teresa	Justin
Experience	Worked on Course Gamification project which used an Angular front end and Django backend and is a large web application. Also worked on some projects in class which used React and Typescript.	Previous in-class project for COSC310. Built with React.	Worked in the industry > 2 years (React, Angular, Scala, Docker). Worked on Course Gamification (Django, Python, Angular, Typescript), Teamformation (Python), Computational Puzzles (React.js, Typescript).	Have worked on Course Gamification project which used an Angular frontend, also worked in industry for two summers, so have some full stack development experience and know TypeScript.	Wrote database software in C, developed for both desktop and embedded platforms. Working on a team formation algorithm in Python. My only web experience is in PHP in COSC 360 and JSP in COSC 304.

Good At	Researching, assisting others with code issues, and management experience. I am also good at implementing logic and algorithms.	Video editing, learning and concept retention	UI mockup, Full-stack Development , Code review	Writing class reports, code reviews, figuring out problems	Backend development, algorithms, data processing.
Expect to Learn	I have some experience with React, but I expect to learn a lot more, especially about how the project is structured. I have also never worked with a full-stack framework, so I will need to get used to that.	I do not have much if any experience with project development of any kind so I expect to learn a lot of elements that make up full stack development. I also expect to learn quite a deal on new and up and coming technologies commonly used in project development.	Improve my project management by doing a project from ground-up. Also, learning a new IDE: Webstorm. Learn more about cloud technology with AWS.	I have never worked with NextJS, so I will probably need some time to get used to it, and expect to learn a lot there. I am also curious about learning how to incorporate the APIs provided by AWS.	I want to learn to use TypeScript as part of an industry standard web development tech stack. I have not had much experience with dockerization or setting up CI/CD either. Learning about the cloud technologies available on AWS is also a great opportunity.

To fill the above table, we went around in a circle during an in-person meeting and discussed our experience. Most of us have worked with each other before, either on research projects or class assignments. After this discussion, we added our experience, what we are good at, and what we are expected to learn into the table.

Category of Work/Features	Seth	Erin	K	Teresa	Justin
Project Management: GitHub Maintenance	✅	✅	✅	✅	✅

Category of Work/Features	Seth	Erin	K	Teresa	Justin
Technical Direction: Time Estimation, Making Programming Choices	✅		✅		
Technical Help: Finding Technical Solutions				✅	✅
Troubleshooting: The Go-To When Others Are Stuck			✅		
Project and Environment Setup (includes dockerization, setting up CI + linting, database setup)	✅				✅
System Architecture Design	✅				✅
User Interface Design		✅	✅		
User account creation			✅	✅	
Login/Logout		✅			
Establish a process for sending videos, including dashboard page and setup basic page structure for video submission.			✅ (create component)		✅ (implement page)
Uploading videos to AWS S3 and retrieving them from AWS S3	✅				
Previewing video in the browser: nice video playing with keyboard shortcuts, setting speed, full-screen mode etc				✅	
Dashboard Page for receiving videos (see all received videos)			✅		
Creating submission boxes (date opened, max time video will be stored, name of submission box, date it will close, who videos are requested from, max video length)		✅			
Email verification for users (set up mail client)					✅

Category of Work/Features	Seth	Erin	K	Teresa	Justin
Video processing features (API calls to AWS), e.g. face blurring and background blurring	✅				
User Interface Design		✅	✅		
Design document creation				✅	
Video creation		✅			
Testing (refactor, test coverage, E2E, integration testing)	✅	✅			
Documentation	✅	✅	✅	✅	✅
Video processing features (API calls to AWS)		✅			✅
Filter/search for videos by title, date, person that submitted	✅				
General comments on video submissions			✅		
Invite links for submission boxes (including emails)			✅		
Modifying submission boxes (update/delete), and add option for multiple submissions to the same box	✅				
Dashboard Page for recently viewed/sent videos, relevant submission boxes			✅		
Dashboard Page to see all videos the user has posted (a library of their own videos)					✅
Email notifications: reminders for submission boxes (if they have been opened or close soon), submission sent/received, comments received				✅	

Category of Work/Features	Seth	Erin	K	Teresa	Justin
Address peer testing debugging #1				✅	
Address peer testing debugging #2	✅		✅		
Testing (refactor, test coverage, E2E, integration testing)				✅	✅
Time-stamped comments on videos and ability to export comments on a video as a text file.			✅		
Multiple people managing one submission box	✅				
Deleting recorded video (retracting submission) and recovering deleted videos (with a grace period)			✅		
Documentation	✅	✅	✅	✅	✅
Tagging and advanced filtering for videos		✅			
Deployment	✅				✅
Custom batch notification to users of something by email (ex: let people who passed an interview know)		✅			
Address peer testing debugging #1	✅				
Testing (refactor, test coverage, E2E, integration testing)				✅	
Customize dashboard, pin, set favourites, hide (submissions, submission boxes), background					✅
UI improvements			✅		
Address peer testing debugging #2		✅			

Category of Work/Features	Seth	Erin	K	Teresa	Justin
Presentation Preparation	✅		✅	✅	✅
Design Video Creation		✅			
Design Video Editing		✅			
Design Report	✅		✅	✅	✅
Final Video Creation		✅			
Final Video Editing		✅			
Final Team Report	✅		✅	✅	✅
Final Individual Report	✅	✅	✅	✅	✅

In the above table, we added checkmarks next to the feature assigned to each person on our team. Most features are assigned to just one individual, except for things like testing, documentation, set-up, and UI design, which will be tackled by multiple people together. Milestones are separated by blank lines in the table. We aimed for an equal distribution of labour, and will reevaluate the assignment of tasks as we move through the project. For the final project submission, Erin will be focusing on videos, since she is the most skilled video editor of the group, while the rest of the team will work on the written reports.