**Project Proposal for Option 3**

**Team Number:** Team 06

**Team Members:** Zetian Zhao 84956093, Jiajun Huang 99007528, Yiqi Xu 53989885, Wenrui Chen 56362320, Ximin Xu 96916259

**1. Overview**

The main purpose of the software is to enable users to upload their video files and save them in the cloud using AWS servers. When a user wants to watch a video, the video streaming is preloaded while the user watches it. The benefit of this technology is that the receiver can watch the video immediately when they open it, without waiting for the entire video to be downloaded. We want to build faster and more secure transmission methods for our customers. We believe in providing a good video viewing experience while providing a secure enough transmission method.

When a user logs into the software, it automatically requests permission for the device's camera and microphone. Once the permissions are obtained, the user can record videos directly in the software. Once the sender has uploaded his video, he can decide how long the video will be valid and can choose to delete his video at any time. The web application also allows senders and receivers to comment below the video. The data from the comments is stored in a database. The software is responsive, and it is designed to be used on a variety of devices, including computers and cell phones. In order to protect users' privacy, publishers can decide who can see their videos. In more detail, publishers can decide whether their videos are public or not. If it is not public, the can decide which group of users are allowed to see the video. The user's video is encrypted by the software while it is in transit. Even if the data stream is accidentally intercepted, no one else can see the content of the video. In addition to this, we plan to blur the faces in the video using the APIs of software that can perform facial blurring.

**1.1 Envisioned Usage**

Our video streaming website offers a platform where users can watch, share, and upload videos securely. While our platform is designed to be intuitive, the following user scenarios will help in understanding the experiences of different user groups.

1. Sender: Sending video

David is a graduate student that would like to upload his homework video to the website in order for his instructor to mark.

* David navigates to our website and logs in with his personal account.
* He clicks the upload button and enters the upload page.
* Input boxes shown on screen and labeled if it is optional. David fills the boxes that he needs and chooses it as a private upload only to his instructor. Finally, he chooses a video file from local for uploading.
* After clicking the upload button, his work is successfully submitted and the file is available on the cloud server for his instructor.
* The website automatically leads David back to his main page.

1. Receiver: Receive file from other   
   Instructor receives multiple videos from students and wants to watch them one by one.

* Instructor navigates the website and logs in with his personal account(Professional).
* He clicks the inbox button and a list of different homework is shown. He picked the one he plans to mark today.
* A table shows senders’ name, date and files’ title and type. He clicks the file title and starts to watch the uploaded video from one of the students.

1. Casual User: Browning and Discovering

Jerry wants to find some interesting videos on the website on weekends.

* Jerry visits the website without a clear idea of what he wants.
* The homepage shows some trending public videos.
* He clicks one of the videos and starts to watch.
* After watching, Jerry wants to make some comments on it but the comment area is not open to users who are not logged in.
* Jerry then completes the sign up and is logged in after.
* He successfully shares his opinion towards the video.

Our MVP for this website is to ensure a smooth experience for different users like student sender, professional user(doctors , teachers etc.) and casual users. We are aiming to have more features in the future, but this is the one that shows the very basic experience of our video streaming website.

**2 Major Milestones**

Note: Normally the testing feature will include testing, debugging, and code review.

| **Deadline** | **Deliverable** |
| --- | --- |
| Term 1 week 9: Mini Presentation | In the first milestone, we will prepare the basic features of a website. The features will be tested and ready to submit:   * Login page with permission and credentials. * Signup page for creation of credentials like username, and password. |
| Term 1 week 13: Design submission | In this milestone, we will be ready to show a completed user interface with a consistent layout, color, and text font. Completed Features could be:   * The uniform user interface for the website (font, color scheme, layout) * Testing user interface interaction * Editable personal info (change password and so on) * Video submission feature which will work with AWS on the upload page. * Testing if the video can be uploaded. * Inbox windows for all people * Testing if people can receive videos from others in private. * sign out feature * Documentation and video demo * CI/CD enabled   In addition, the demo video will show how users will interact with the website like uploading the videos and so on. The frontend Test will also be completed. |
| Term 2 week 4: Peer Testing | Completed features will be:   * Functionality that people can see who send videos to their inbox * Testing and debugging if the video in the inbox has a name or not. * Find a specific video with a specific name or part of the name * Test if the filter works or not. * comment feature on a video limited to login-user * Test if the people who are not logged in can comment or not. * A homepage that contains most-view videos. * Test if the videos shown as trending. * A pop window that contains a list of videos after clicking the inbox icon. * Test if the pop window works or not.   All developed features (code) will be reviewed at the same time by all team members. The code integration will also be in progress. |
| Term 2 week 8: Peer Testing | Write tests for features:   * Test if users can sign up an account and can login with permission * Test if users can change password once the account is created * Create a logo and if it’s clicked then the webpage should be back to the top * Test if users can click the logo and go back to the top * When the user clicks on the video, increase the number of views by one and display the number of views in the bottom right corner of the video * Test if the number of views is increased by 1 and displayed properly after someone clicks the video * If the senders delete their video, the receivers should not receive the video sent by the senders * Test if senders can delete their videos successfully * Users can edit the video or make a clip of it * Test if users can do editions   All the features at this time should be tested and we should prepare to do a video to show how this application basically works. |
| Term 2 week 13: Final project submission | Conclusion:   * A documentation/reflection about what we did, how we did, what we could do better, what unexpected cases are and what the most difficult part is * A whole demo to show how our application works including the front-end and back-end * Test report should be ready because we basically test all the functions * Users can record the video or make a clip of it * Test if users can do recordings * The faces of the people in the video can be mosaicked. * Test if the faces in the video can be recognized and mosaicked |

**3. Technology Stack**

In this Project we are required to use Amazon Cloud Service (AWS). In order to accomplish our goal to create a streaming platform, we need to use the following service. The programming language we choose is based on php, python, sql, html, javascript and java.

* 1. **Amazon S3 Bucket**: This will serve as our primary storage for video content. It provides durability and scalability, making it a suitable choice for storing videos. Users can upload videos to this bucket, and we can also use it to serve as a source for Amazon Elastic Transcoder. It is currently used by NASCAR, SNAP and other technology companies.
* 2.**Amazon CloudFront**: Amazon CloudFront is a content delivery network (CDN) that will help in delivering our stored videos to users with high speed and low latency. This ensures a smooth streaming experience for our users by distributing content to edge locations around the world. It is currently used by Zalando, Honda, Supercell and other industrial and technology companies.

3.**Amazon Elastic Transcoder:** This service will be used to transcode different types of video files into a common format. This is essential for ensuring compatibility and smooth playback across various devices and platforms.It is good to cope with Amazon S3 bucket.

4.**Amazon Relational Database Service (RDS)**: We will use Amazon RDS to store user information, including video URLs. This is a suitable choice for managing user data in a structured manner and facilitating user authentication, authorization, and video retrieval through GET requests. It is currently used by Cathay Pacific, Samsung and other technology and service companies.

Technology user will be interact with:

* **Web Browser:** Users will access your streaming platform via a web browser. This is the primary interface through which users will browse, search for videos, and interact with your platform.
* **PC:** Ensuring a good user experience on PCs is crucial, as many users prefer larger screens for video streaming. We will design the web application to be responsive, so it adapts to different screen sizes, including PC screens, for a professional and user-friendly experience.

**4. Teamwork Distribution and Anticipated Hurdles**

Team Experience, Expertise, and Areas of Learning:

| **Category** | **Zetian Zhao** | **Wenrui Chen** | **Jiajun Huang** | **Ximin Xu** | **Yiqi Xu** |
| --- | --- | --- | --- | --- | --- |
| Experience | Backend experiences like database generation, link database, token generator, and RESTful API. Frontend experience would be a website of the market using JS, HTML, and CSS. | Data analytics like using R and SQL to catch data and model it in excel. Frontend: JS , HTML and CSS | Connection of backend and frontend. Familiar with AI models(might help). Good JS skill. | Using python to create a script of connecting to database | Build a website by LAMP, and connect an API from chatgpt. |
| Good at | Java, HTML, CSS, SQL, Database management. | Java, R, SQL  HTML | Python, Java, HTML, JS, SQL | Java, Python, SQL ddl, javascript, php, html, | Java, Python, mySQL, React, Nodejs |
| Expect to learn | JS (get familiar more), connection of frontend and backend, AWS. | AWS cloud service and frontend stuff like JS with CSS | How does AWS work and what AWS actually for. Cloud server. | Interact with AWS, get used to connecting my studied knowledge to practice. | how to use AWS in different scenarios and connect AWS with frontend. |

Expected Areas of Contributions:

| **Category of Work/Features** | **Zetian Zhao** | **Wenrui Chen** | **Jiajun Huang** | **XiMin Xu** | **Yiqi Xu** |
| --- | --- | --- | --- | --- | --- |
| Project Management: Trello Maintenance | ✓ | ✓ | ✓ | ✓ | ✓ |
| Technical Direction: Time Estimation, Making Programming Choices |  |  | ✓ | ✓ | ✓ |
| Troubleshooting: The Go-To When Others Are Stuck | ✓ | ✓ |  |  | ✓ |
| System Architecture Design | ✓ |  |  |  | ✓ |
| User Interface Design | ✓ |  | ✓ | ✓ | ✓ |
| CSS Development |  | ✓ |  | ✓ |  |
| 1.Login page with permission and credentials. | ✓ | ✓ |  | ✓ | ✓ |
| 2.Signup page for creation of credentials like username, and password. | ✓ | ✓ |  | ✓ | ✓ |
| 3.The uniform user interface for the website (font, color scheme, layout) |  |  | ✓ | ✓ | ✓ |
| 4.Testing user interface interaction | ✓ | ✓ |  | ✓ |  |
| 5.Editable personal info (change password and so on) |  | ✓ |  | ✓ | ✓ |
| 6.Video submission feature with AWS on the upload page |  |  |  | ✓ | ✓ |
| 7.Testing if the video can be uploaded. | ✓ |  |  | ✓ |  |
| 8.Testing if people can receive videos from others in private. | ✓ | ✓ |  |  |  |
| 9.sign out feature |  |  | ✓ | ✓ | ✓ |
| 10.Documentation and video demo | ✓ |  |  |  |  |
| 11. CI/CD enabled |  |  |  |  |  |
| 12.Functionality that people can see who send videos to their inbox |  |  | ✓ |  | ✓ |
| 13.Testing and debugging if the video in the inbox has a name or not. | ✓ |  |  |  |  |
| 14.Find a specific video with a specific name or part of the name |  |  |  | ✓ |  |
| 15.Test if the filter works or not. | ✓ |  |  | ✓ |  |
| 16. comment feature on a video limited to login-user |  | ✓ |  | ✓ | ✓ |
| 17.Test if the people who are not logged in can comment or not. | ✓ |  |  | ✓ |  |
| 18.A homepage that contains most-view videos. |  | ✓ |  | ✓ | ✓ |
| 19.Test if the videos shown as trending. | ✓ |  | ✓ |  |  |
| 20.A pop window that contains a list of videos after clicking the inbox icon. |  |  | ✓ | ✓ | ✓ |
| 21.Test if the pop window works or not. | ✓ |  |  | ✓ | ✓ |
| 22.Test if users can sign up an account and can login with permission |  | ✓ |  | ✓ |  |
| 23.Test if users can change password once the account is created | ✓ |  |  | ✓ |  |
| 24.Create a logo and if it’s clicked then the webpage should be back to the top |  |  | ✓ | ✓ |  |
| 25.Test if users can click the logo and go back to the top | ✓ |  |  |  | ✓ |
| 26.When the user clicks on the video, increase the number of views by one and display the number of views in the bottom right corner of the video |  |  | ✓ | ✓ |  |
| 27.Test if the number of views is increased by 1 and displayed properly after someone clicks the video | ✓ | ✓ |  | ✓ |  |
| 28.If the senders delete their video, the receivers should not receive the video. |  |  |  | ✓ | ✓ |
| 29.Test if senders can delete their videos successfully | ✓ |  |  | ✓ |  |
| 30.Users can edit the video or make a clip of it |  |  | ✓ | ✓ | ✓ |
| 31.Test if users can do editions | ✓ |  |  | ✓ | ✓ |
| 32.A documentation/reflection about what we did, how we did, what we could do better, what unexpected cases are and what the most difficult part is |  | ✓ | ✓ |  | ✓ |
| 33.A whole demo to show how our application works including the front-end and back-end |  |  | ✓ | ✓ |  |
| 34.Test report should be ready because we basically test all the functions | ✓ |  |  | ✓ |  |
| 35.Users can record the video or make a clip of it |  |  | ✓ | ✓ |  |
| 36.Test if users can do recordings | ✓ |  |  |  |  |
| 37.The faces of the people in the video can be mosaicked. |  |  | ✓ | ✓ | ✓ |
| 38.Test if the faces in the video can be recognized and mosaicked | ✓ |  | ✓ | ✓ |  |
| Database Setup | ✓ |  |  | ✓ | ✓ |
| Presentation Preparation | ✓ | ✓ | ✓ | ✓ | ✓ |
| Design Video Creation | ✓ | ✓ | ✓ | ✓ | ✓ |
| Design Video Editing |  |  | ✓ |  |  |
| Design Report | ✓ |  |  | ✓ |  |
| Final Video Creation |  |  | ✓ | ✓ |  |
| Final Video Editing |  |  | ✓ |  |  |
| Final Team Report | ✓ | ✓ | ✓ | ✓ | ✓ |
| Final Individual Report | ✓ | ✓ | ✓ | ✓ | ✓ |