Final Documents

B11102015 Bingxiu Yu / B11102213 Guojun Zeng/

B11102112 Chiajui Lee

I. System Architecture :

A. HTML: The standard markup language for creating web pages. HTML is a foundational technology used alongside CSS and JavaScript to design web pages and user interfaces.

B. CSS: Enhances the layout and appearance of web pages, providing visually appealing designs.

C. Javascript: It can be used to manipulate the content and structure of HTML documents, dynamically updating the page without requiring a reload. It responds to user interactions, such as button clicks or text input, thereby altering the appearance and behavior of the page.

D. Node.js : Node.js is a JavaScript-based backend runtime environment that runs code using the Chrome V8 JavaScript engine. It focuses on providing a non-blocking and event-driven architecture, making it suitable for building high-performance, scalable network applications. Node.js is one of the key technologies in modern web development, particularly widely used in real-time applications and API development.

E. Express : Express.js is a fast, flexible, and minimalist web application framework built on Node.js, primarily used for constructing web applications and APIs. It provides a powerful set of tools to handle routing, requests, responses, and middleware.

F. MySQL: MySQL is an open-source (free) database management system widely used in small to medium-sized websites. It works alongside web programming languages such as PHP, ASP, or ASP.NET to store large amounts of data. If a website has a backend management system (admin panel), it typically requires database functionality. A database serves as a warehouse for storing large amounts of data and files, SQL acts as the communication protocol with the warehouse, and MySQL is the system used to manage the warehouse.

G. WebRTC : WebRTC is an open-source project designed to enable real-time audio, and video communication, and data transfer between web browsers and mobile applications through simple JavaScript APIs.

H. Azure: Users can quickly deploy and manage website environments, supporting various development frameworks and tools, suitable for testing, deployment, and scaling web applications. By integrating Azure services, developers can easily achieve high performance, security, and scalability in website solutions.

I. Socket: Sockets provide the foundation for real-time communication in web development, allowing persistent connections between the server and client. It supports bidirectional data transmission, making it ideal for developing chat applications, real-time notifications, and multiplayer interactive applications. Through efficient, low-latency communication, sockets offer powerful support for implementing real-time features in modern web applications.

J. TURN Server： The TURN server is a crucial component of WebRTC technology, designed to handle situations where direct peer-to-peer (P2P) connections cannot be established due to firewalls or NAT configurations. The primary function of the TURN server is to act as a relay server, allowing data to pass through the server to facilitate communication between devices.

II. System Service and Feature :

A. Streaming platform main page

B. streamer dashboard

C. streamer chat box

D. EMOJI

E. streaming screen

III. Professional Competencies required for the course :

A. Ability to query data

B. Ability to find solutions and resolve issues

C. Leadership skills and teamwork collaboration

D. Strong communication skills

E. Basic web design skills (HTML, CSS)

F. Programming skills in JavaScript

IV. Design Concept :

Most of our team members enjoy watching live streams from Twitch streamers and are familiar with most of Twitch's interface and features. We want to use Twitch as a model, simplifying and modifying it to create a live streaming platform with our own unique features.

V. Result and analysis :

In this project, we successfully built a web-based live streaming platform and implemented core features such as the homepage, streamer dashboard, chatroom, and real-time streaming display. Throughout the development process, we overcame several technical challenges and accumulated valuable experience.

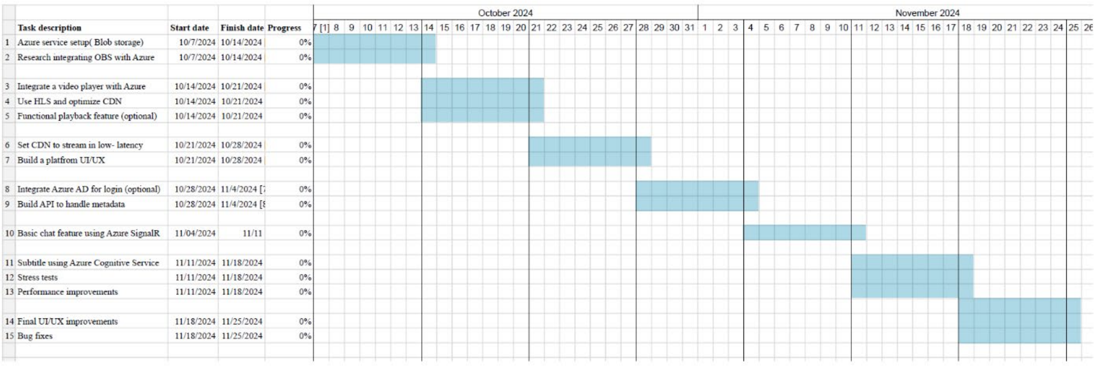
For the server-side, we used Node.js combined with the Express framework for development. This architecture not only improved the flexibility and stability of the system but also accelerated the development process. In terms of security, to protect user passwords and data, we used bcrypt for password encryption, ensuring that even if the database were compromised, passwords could not be reverse-engineered. Additionally, we implemented a session-based user authentication mechanism, managing user login states through a unique session ID, further enhancing the system's security.

For live streaming, we originally planned to use the RTMP protocol for real-time streaming, but due to its inherent delay of at least 10 seconds, we ultimately chose to switch to WebRTC technology. WebRTC is a peer-to-peer real-time communication technology that effectively reduces latency and enhances user experience. At the same time, we mastered key components such as ICE, STUN, and TURN, successfully implementing stable real-time communication features.

The team had clear divisions of labor, with each member responsible for designing and developing different modules. We not only learned and mastered front-end web technologies like HTML, CSS, and JavaScript, but also became familiar with the use of MySQL databases for efficient data querying and management in backend development. For cloud deployment, we used Azure services, learned how to set up virtual machines, and configured the application environment to complete the online deployment of the project.

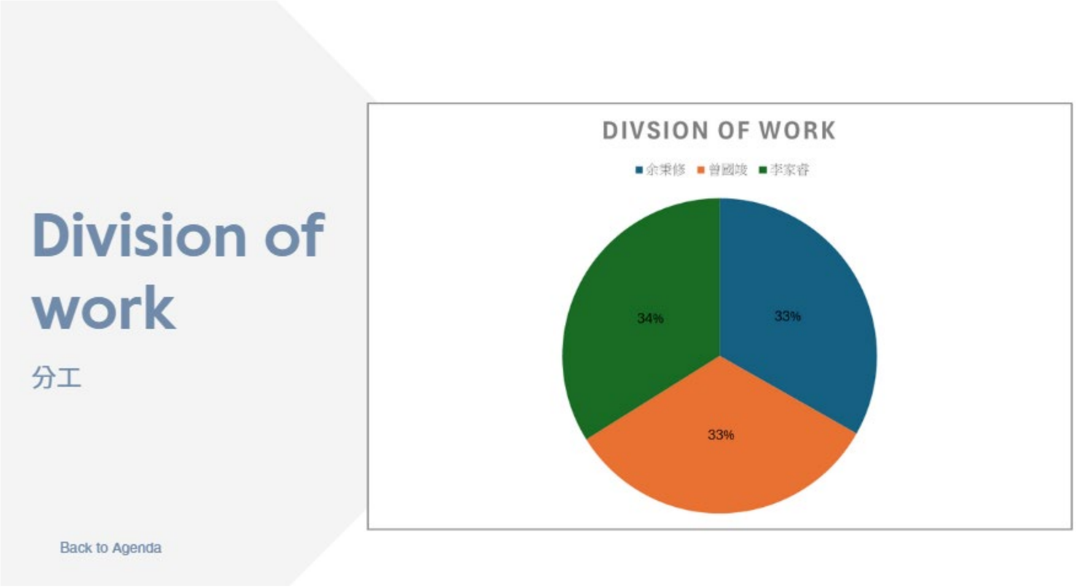
Through this project, we not only successfully built a live streaming platform with basic functionality but also learned various modern development technologies, such as cloud services, real-time streaming protocols, and database design. The application of these technologies not only solved real-world development challenges but also provided us with valuable experience for participating in more complex projects in the future.

VI. Schedule :



	Task description	Start date	Finish date	Progress
1	Azure service setup(Blob storage)	10/7/2024	10/14/2024	0%
2	Research integrating OBS with Azure	10/7/2024	10/14/2024	0%
3	Integrate a video player with Azure	10/14/2024	10/21/2024	0%
4	Use HLS and optimize CDN	10/14/2024	10/21/2024	0%
5	Functional playback feature (optional)	10/14/2024	10/21/2024	0%
6	Set CDN to stream in low- latency	10/21/2024	10/28/2024	0%
7	Build a platfrom UI/UX	10/21/2024	10/28/2024	0%
8	Integrate Azure AD for login (optional)	10/28/2024	11/4/2024 [7	0%
9	Build API to handle metadata	10/28/2024	11/4/2024 [8	0%
10	Basic chat feature using Azure SignalR	11/04/2024	11/11	0%
11	Subtitle using Azure Cognitive Service	11/11/2024	11/18/2024	0%
12	Stress tests	11/11/2024	11/18/2024	0%
13	Performance improvements	11/11/2024	11/18/2024	0%
14	Final UI/UX improvements	11/18/2024	11/25/2024	0%
15	Bug fixes	11/18/2024	11/25/2024	0%

VII. Job partition :



Division of work

分工

Back to Agenda

VIII. Encountered problems and how to solve it :

A. How to create a server?

Solution: JavaScript has a built-in http module, and the Express framework is built on top of this http module, making it more convenient to use with enhanced functionality.

B. How to ensure password security by encrypting it?

Solution: Use bcrypt.hash to encrypt passwords. Once encrypted, passwords cannot be reverse-engineered.

C. How to authenticate users?

Solution: Use sessions for user authentication. When the user logs in, the server generates a unique session ID and sends it to the client.

D. RTMP causes at least a 10-second delay.

Solution: Use WebRTC, as taught in later classes, for live streaming to minimize latency.

IIX. Techniques we learned :

A. Cloud :

1. Provides on-demand virtual servers, storage, and networking services.
2. Offers tools for developing and deploying applications without the need to manage infrastructure.
3. Delivers software-as-a-service (SaaS) through the internet.

B. Create a virtual machine :

1. Enter Azure services and click "Create a resource."
2. Select "Virtual Machine," then configure the name, region, and specifications.
3. Review the settings, click "Review + Create," and then click "Create."

C. .HTML,CSS,JS :

HTML (Hypertext Markup Language) is used to structure web pages, organizing content in a hierarchical manner. CSS (Cascading Style Sheets) is responsible for styling the web page, making the design more attractive and consistent. JavaScript is a powerful programming language that enhances web interactivity, enabling dynamic features and functionalities.

D. DB :

A database is a system tool used to store and manage data, efficiently handling large-scale datasets. We are familiar with SQL (Structured Query Language), which allows us to manage data through querying, inserting, updating, and deleting operations. We understand database structure design, including concepts such as tables, data columns, primary keys, and foreign keys. Additionally, we are familiar with both relational databases (such as MySQL and PostgreSQL) and non-relational databases (such as MongoDB).

E. RTMP :

We have learned and mastered the basic principles and applications of RTMP (Real-Time Messaging Protocol). RTMP is a protocol designed for real-time streaming media transmission, commonly used in live streaming and video-on-demand scenarios. We understand the data transmission mechanism of RTMP, including how to use it for transmitting audio, video, and metadata.

F. WebRTC :

WebRTC is an open standard technology that enables peer-to-peer audio, video, and data transmission between browsers or mobile devices without the need for additional plugins. We understand the core components of WebRTC, including ICE (Interactive Connectivity Establishment), STUN (Session Traversal Utilities for NAT), and TURN (Traversal Using Relays around NAT), and how they help facilitate connection traversal through NAT (Network Address Translation).