-- video intro --

Twitch is a multi-billion dollar streaming platform developed by a team of professionals and currently owned and operated by trillion dollar corporate giant Amazon. Every day, 15 million people log into Twitch to tune into their favorite streams, and about 9 million people host streams every month. It took four years for the project to finally enter a public beta. I'm going to make it in like a week.

-- end video intro --

Chapters:

- Nginx and rtmp with the linux server (with breif explination into rtmp)

- Convert to HLS

- set up the viewing page

- chat

- account creation

- settings

-- Nginx and rtmp in Linux server (Day 1) --

The first step to any service like this is the backend. More specifically, the server that the streams go to. This is done thorough a thing known as RTMP or Real-time Messageing Protocol.

-- Explaination of RTMP --

TLDR; Made by a company I've never heard of, bought by Adobe, streams audio and video over the wifis, It's exactly what I need, I'm using it.

// Old:

// RTMP is a protocol made by a company called Macromedia which was later // bought by Adobe. It was made to stream audio, video, and other data over // the internet between Adobe Flash Player and a server.

-- end Explaination of RTMP --

Unluckily for us, It was made for Adobe Flash and Adobe shut down Flash Player last year so we can't just go streight from rtmp to browser. We need to convert the stream.

-- End of Nginx and rtmp in Linux server --

-- Convert to HLS --

It's one line.

-- end Convert to HLS --

-- Set up viewing page (Day 3) --

Now for the fun part. The website itself. Some styling, some code, bip bam boop.

-- end Set up viewing page --

-- Chat (Day 4) --

The chat part is easy enough. With firebase as a database just keep the chat in a database and use the onSnapshot function. Every time something in the database changes, like a message being posted, the funciton runs and lets us update the chat. Moderation is cringe, so cope, seithe and mald. CHAT'S DONE.

-- end chat --

-- Accounts (Day 5) --

Now we need user accounts for what I hope are obvious reasons. Boop //Firebase auth. A basic login form and some spaghetti code later and we have a working account system.

-- end accounts --

-- Account settings (Day 6) --

With accounts, comes personalization. With some CSS we can make a dropdown from the profile picture with options like "stream manager" which we will make later, "settings" which we are making now, and "logout". For the settings page lets go with a div with a black border and rounded corners. Profile picture, display name, and bio. New section, privacy. Change password and that's that.

-- end Account settings --

-- Stream Manager (Day 7) --

Time for the stream manager. Just use the ViewPage and make a few modifications.

-- end Stream manager --

-- Discovery home --

Finally we need something for the home page. YouTube has the recommended, Twitch has explore, we're calling it discovery. It's... not hard. OUTRO!

-- end Discovery home --

-- Video Outro --

That's twitch in a week. This was harder than I thought. Thank you for watching and if you liked it press the red button for more.

-- end Video Outro --