Requirements for Website

- login, logout
  → redirect if authenticated
- upload, take down movies
  → store + make public, URL or file upload, denote uploader

{
- classic SQL injection
- blind SQL injection
- SSRF
- Code injection

Use Cases + Diagrams

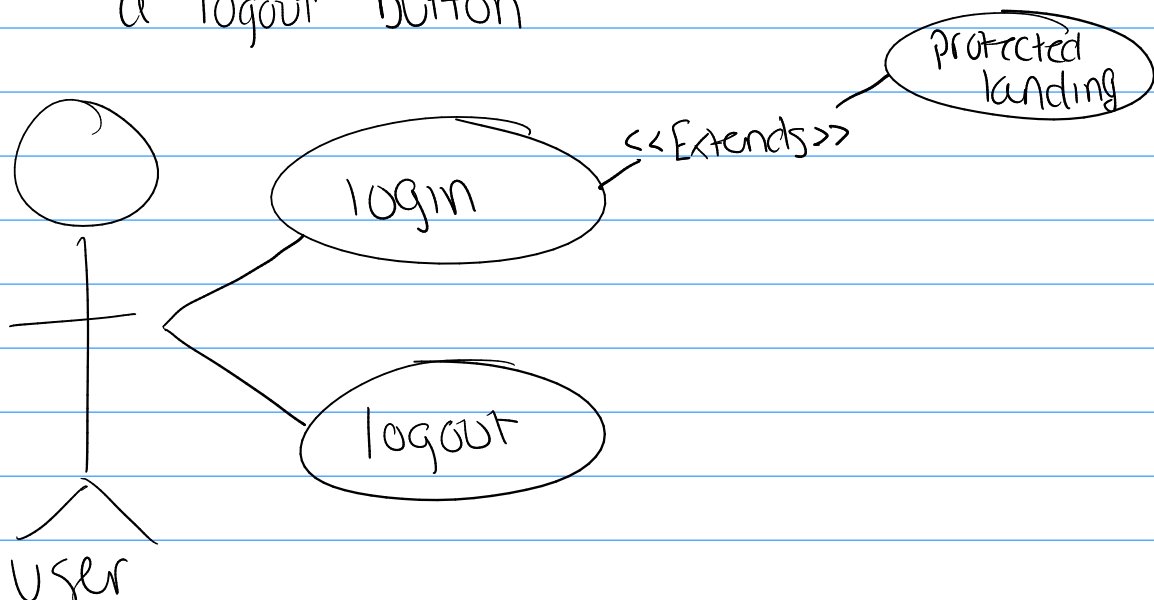A user wants to login to the application

They provide their username, password

If the credentials match an entry in the database, the user is authenticated

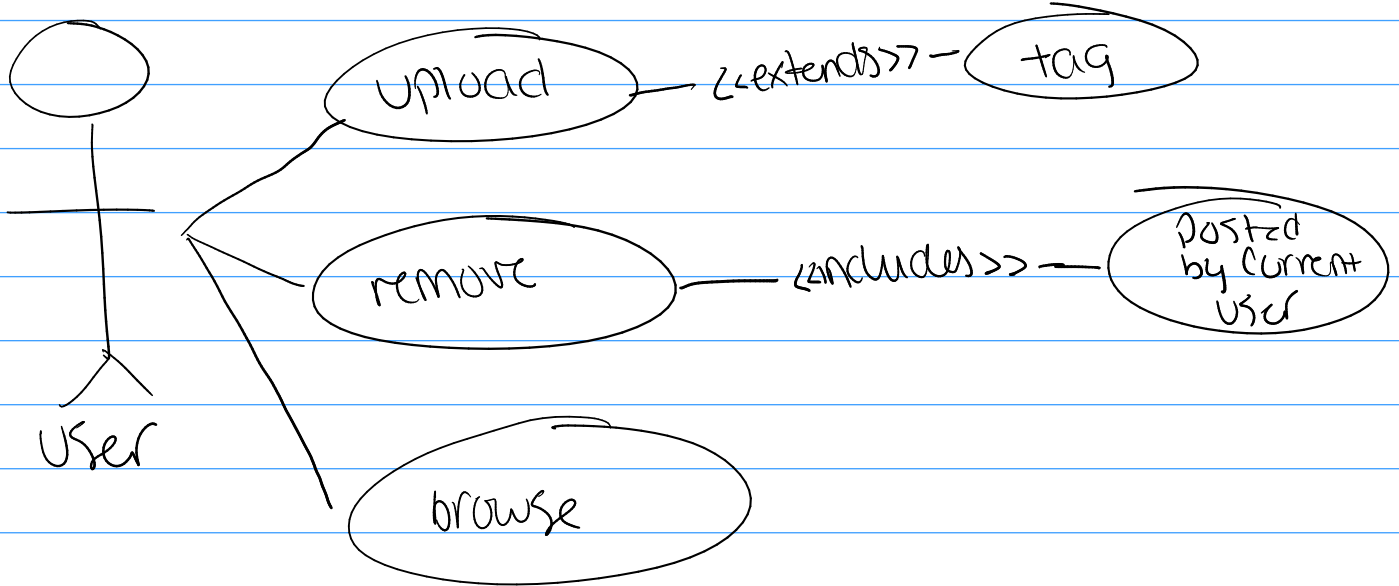If the user is not authenticated, they are prompted for new credentials

If the user is authenticated, they are shown a protected landing page

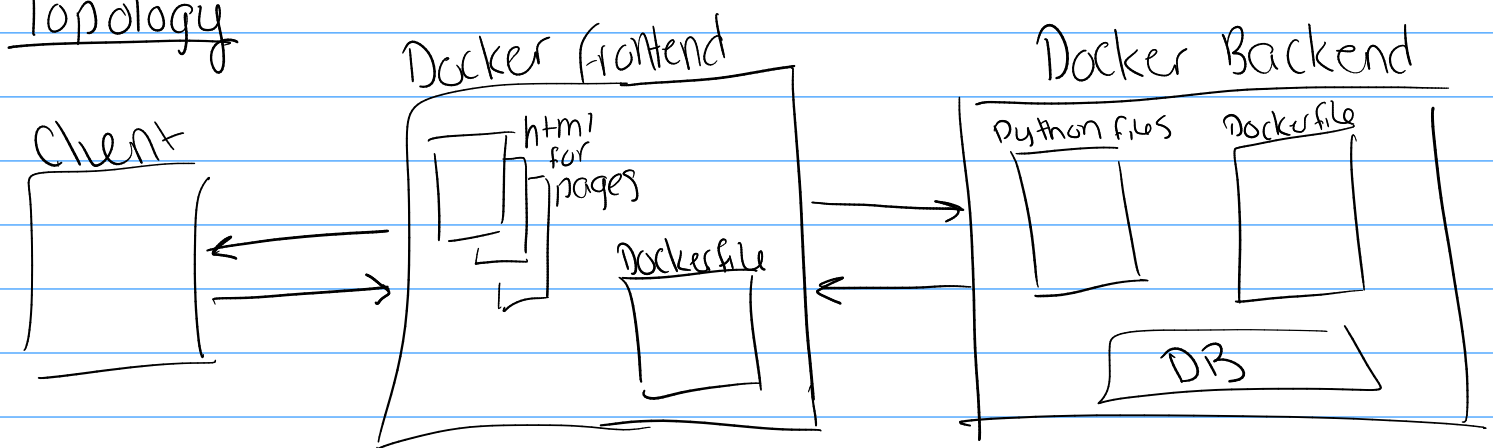Once a user is logged in, they can logout by pressing a "logout" button
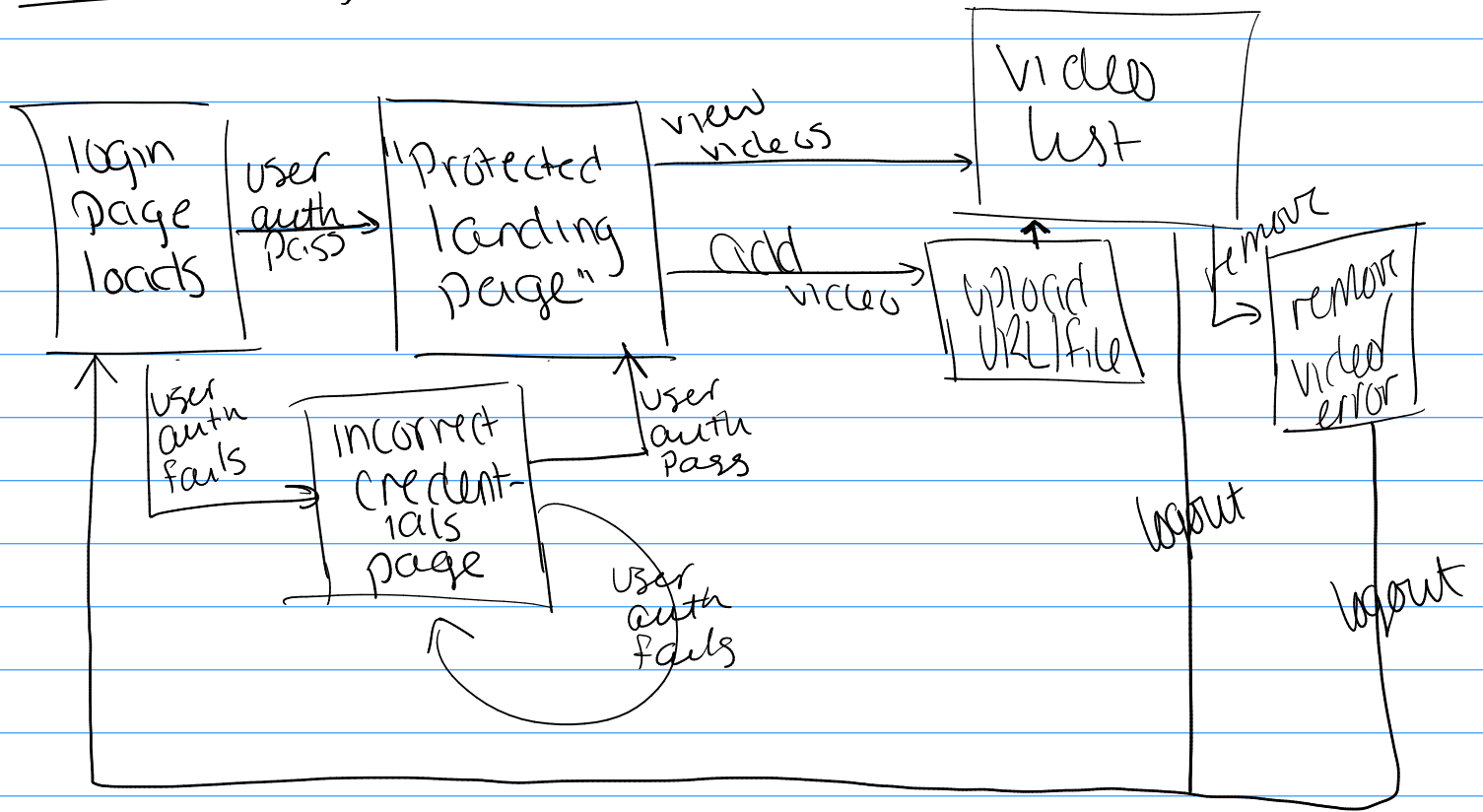
Will be tagged
with username

A user can upload videos by providing a file or URL
A user can take down videos if they are the poster
A user can browse/watch videos uploaded by other users

User

Upload —— <<extends>> — tag

remove —— <<includes>> — Posted by Current User

browse

Topology

Client

Docker Frontend
html for pages
Dockerfile

Docker Backend
Python files   Dockerfile
DB

State Diagrams

Login Page loads → user auth pass → "Protected landing page"

"Protected landing page" → view videos → Video List

"Protected landing page" → add videos → Upload URL/file

Video List / Upload URL/file → remove → remove video error

user auth fails → Incorrect credentials page

Incorrect credentials page → user auth pass → "Protected landing page"

Incorrect credentials page → user auth fails (loop)

logout → Login Page loads

Database Schema