

**Done By**

Othman Maher

**Instructors**

Dr.Motasem Aldiab

Dr.Fahed Jubair

Containerization Project

Contents

Introduction…………………………………………………………………………………………………………3

Run The Project……………………………………………………………………………………………………4

Services……..…………………………………………………………………………………………………………5

Streaming Service……………………….……………………………………………………………5

Upload Frontend………………………………………………………………………………………6

Authentication Service………………..……………………………………………………………8

File System Service……………………..……………………………………………………………8

Upload Service…………………………………………………………………………………………9

MySql Database………………………….…………………………………………………………10

Introduction

-In this assignment, we’ve got a choice to choose between build a containerized microservices about Analytics System or Video Streaming System, and I chose a Video Streaming System but I made my own changes on the system, here is my own implementation.

Upload Service

Streaming

Service

Authentication Service

MySql DB

File system service

Upload

Frontend

-Streaming Service **and** Upload Frontend are images built using Vue.js (front-end framework).

Streaming Service: to stream and show the uploaded videos.

Upload Frontend: to upload the videos based on the users in the front-end.

-Authentication Service, File System Service **and** Upload Service are images built using Spring Boot (back-end framework).

Authentication Service: to create users and generate JWT after successful logging after talking to the database.

File System Service: to upload files to the storage and return the links, I used S3 Buckets.

Upload Service: to save the video information in the database.

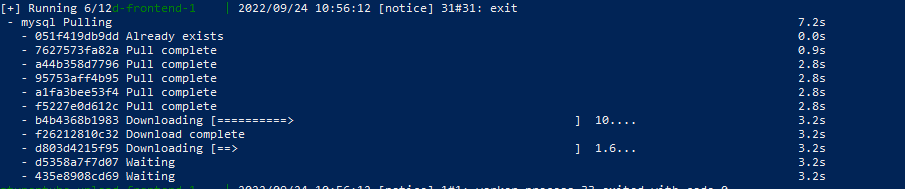
-MySql DB: database image that has the users and videos information.

Run the Project

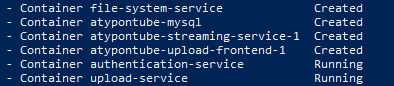
-First of all, you need to download Docker Desktop from their website (<https://docs.docker.com/get-docker/>), then run it in your device.

-After running Docker open the Microservice Project folder in vs code or open the terminal and change directory to the project folder and run (**docker-compose up**).

-It will first create the images if they’re not exist



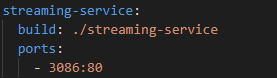
Then it will create containers from the images then run it



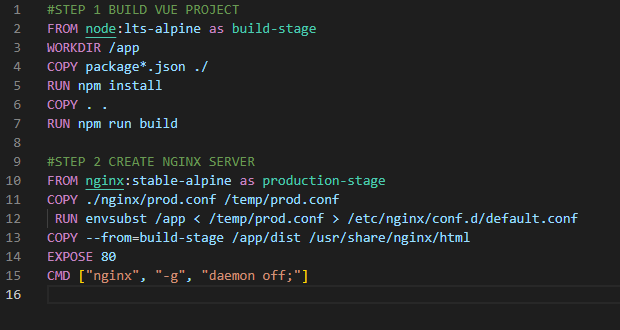
-After the run finishes, go to one of your browsers and type this link to see the Streaming image (<http://localhost:3086/>) to see users videos or the Upload Frontend image (<http://localhost:3087/>) to create an account if you don’t have one and logged in the system to upload your videos.

Streaming Service

* The Implementation for this image in the Docker Compose file



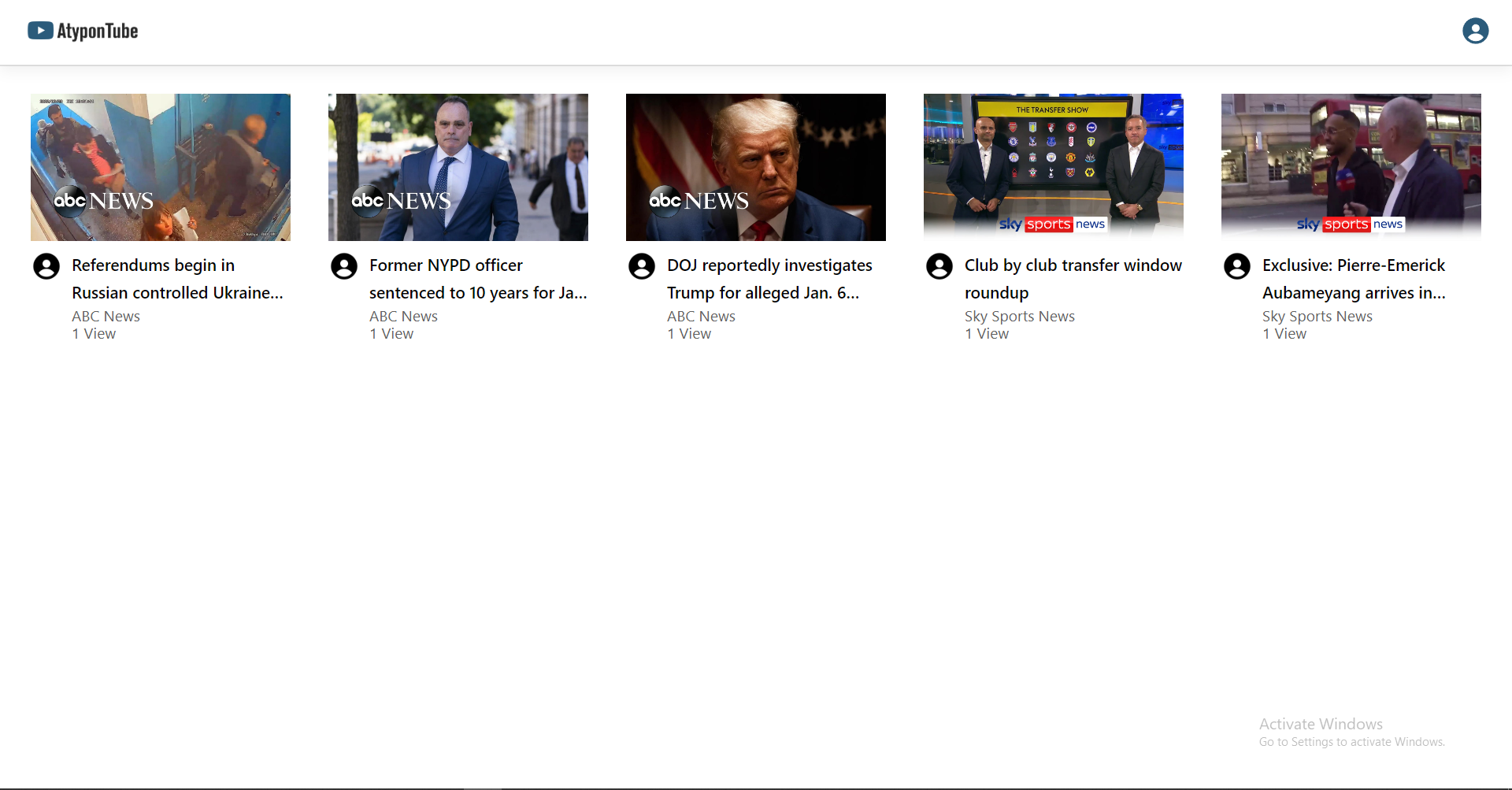
* The Docker file



**Description**

This image is built using Vue.js (front-end framework) and it has two pages.

-Home Page that has all the videos in the System that got it from the Upload Service.

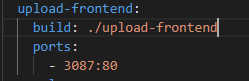


-Video Page that stream the video and get the video details from Upload Service.



Upload Frontend

* The Implementation for this image in the Docker Compose file



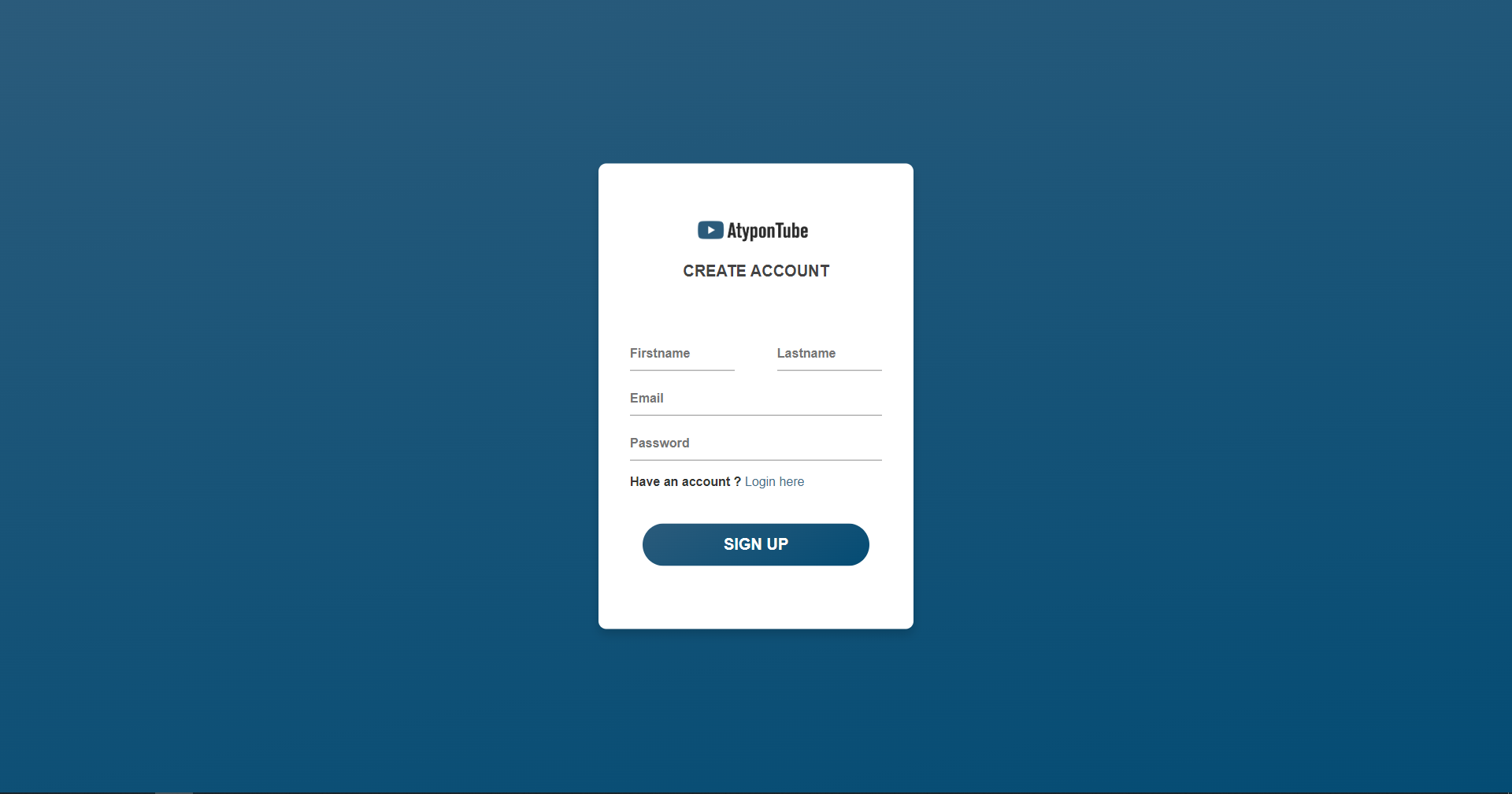
* The Docker file



**Description**

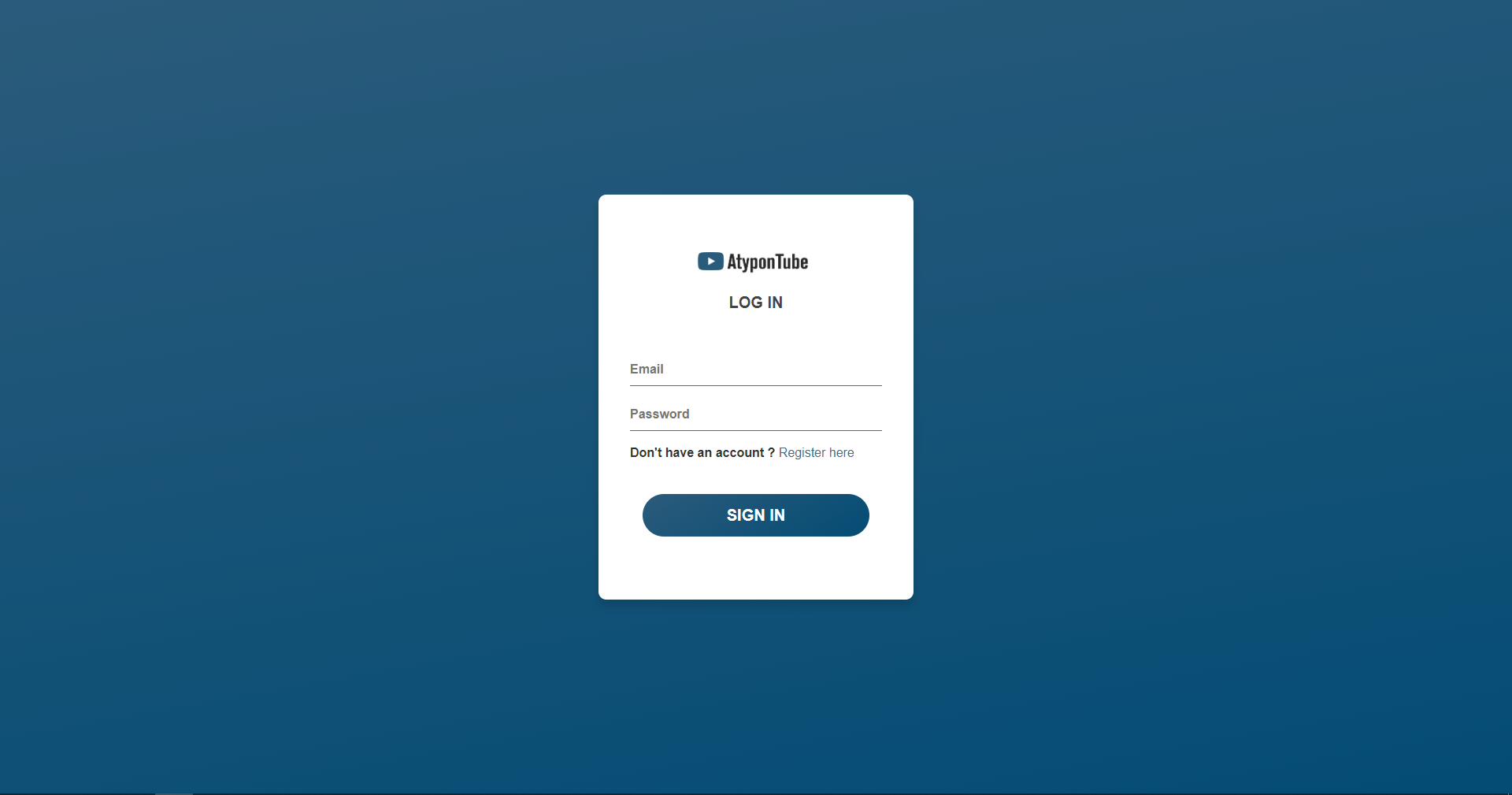
This image is built using Vue.js (front-end framework) and it has three pages.

-Register Page that contact with the Authentication Service to create users.



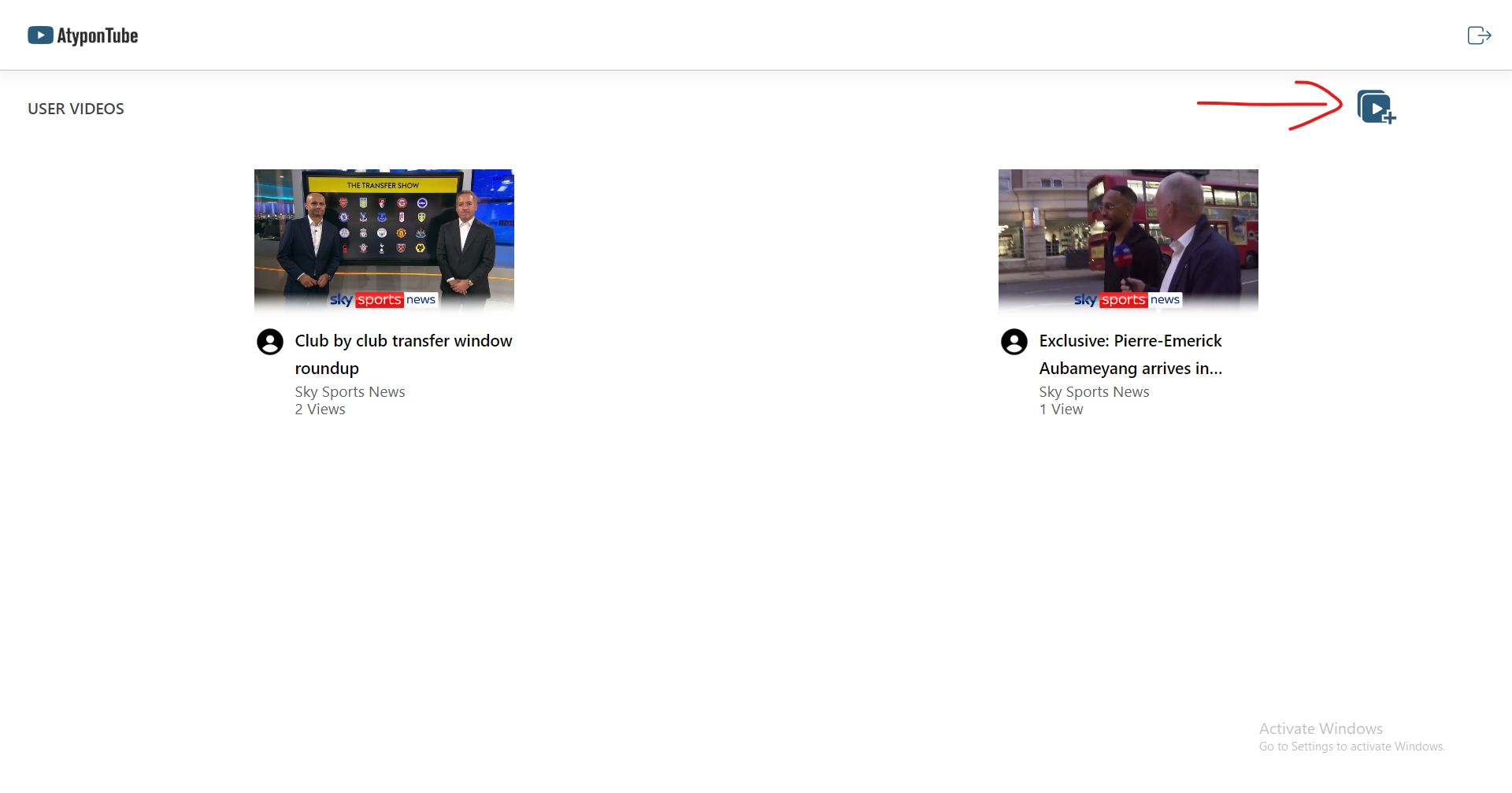
-Login Page that contact with the Authentication Service to generate JWT and logged

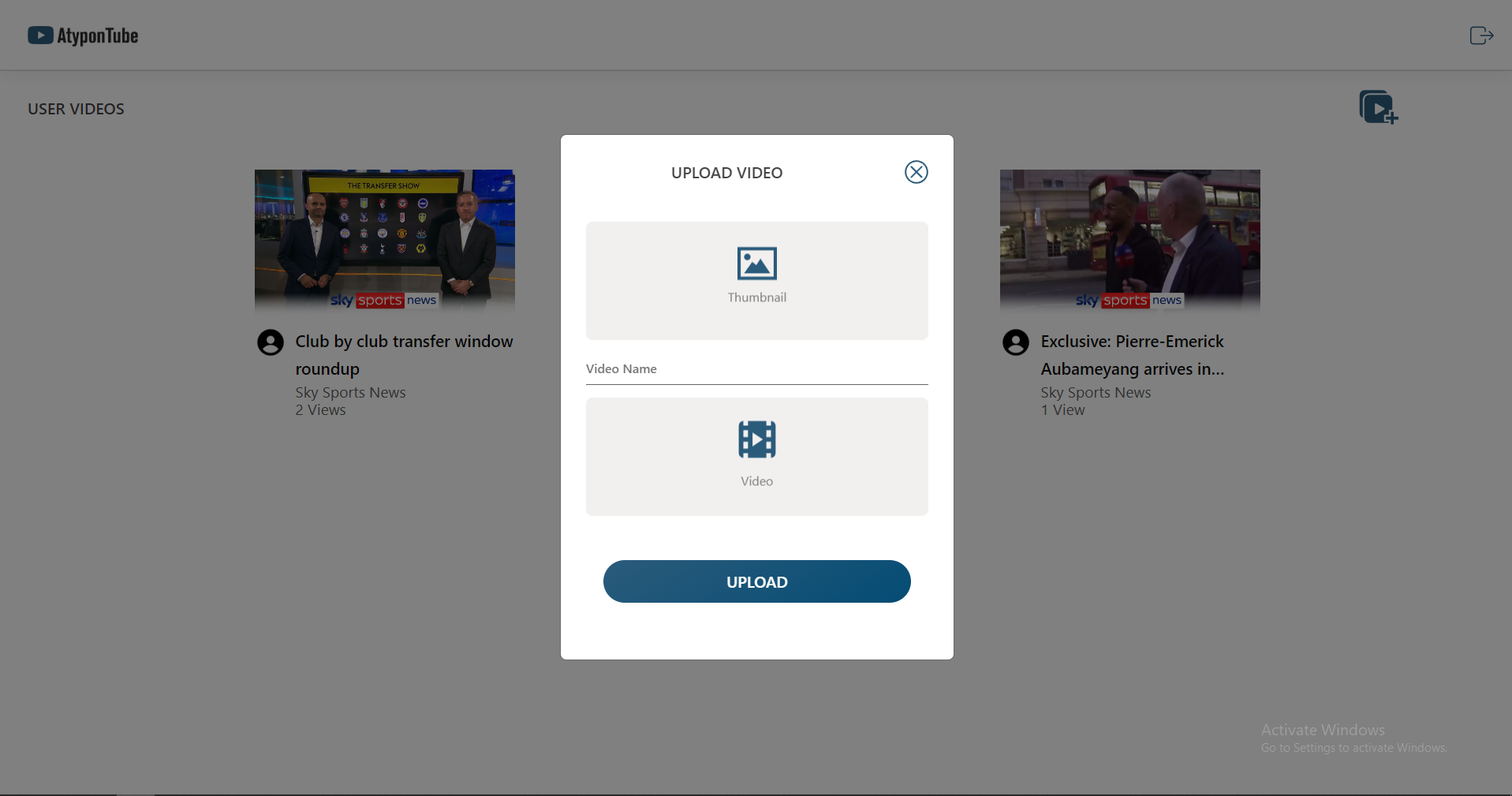
the users.



-Home Page that has all the user videos that came from Upload Service and a button

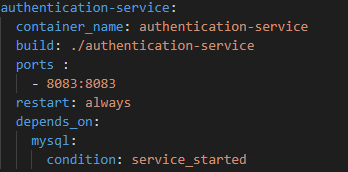
triggers a window that takes video information to upload it.



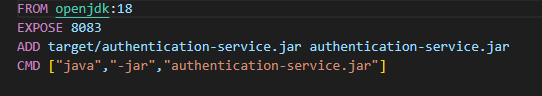


Authentication Service

* The Implementation for this image in the Docker Compose file



* The Docker file



**Description**

This image is built using Spring Boot (back-end framework), it communicate with the database image to save the users and generate JWT to authenticate the users and it has three endpoints.

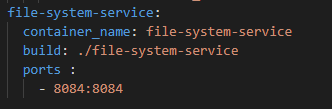
-Login endpoint that authenticate the user and generate the token

-Register endpoint that create the user and save it in the database

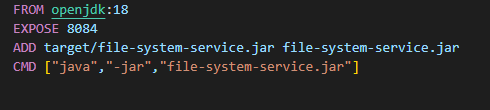
-Verify endpoint that check if the token is not expired and valid

File System Service

* The Implementation for this image in the Docker Compose file



* The Docker file



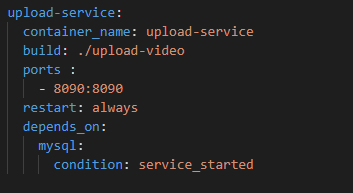
**Description**

This image is built using Spring Boot (back-end framework) to upload files, I used S3 Bucket service from AWS and it has one endpoint.

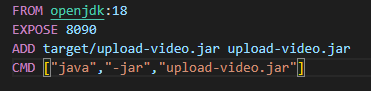
-Upload endpoint that upload files to S3 and return link for the uploaded file.

Upload Service

* The Implementation for this image in the Docker Compose file



* The Docker file



**Description**

This image is built using Spring Boot (back-end framework) to upload videos by save the information of the videos in the database, it has five endpoints

-Upload end point that take video information then communicate with the File System.

Service to upload the video and thumbnail then communicate with the database to

save the information.

-Get Video endpoint that communicate with the database to take video information by

It’s Id.

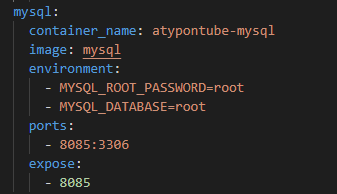
-Get All Videos endpoint that get all the videos details from the database

-View Video endpoint that increase the video views after giving the endpoint video id.

-Get User Videos endpoint that get the user videos from the database by user id.

MySql Database

* The Implementation for this image in the Docker Compose file



**Description**

This image is built using MySql (Structured Query Language Database) that saves the users and videos information and it has two tables.

-User Table that contains (userId,email,password,name).

-Video Table that contains (videoId,videoLink,thumbnailLink,name,views,author,userId).