# Project 5 Natflix


- Project Requirement....1
- Software Used....3
- Project Flow -JIRA....3
- Backend (Spring Boot) - setup....8
- MySQL....9
- Front end -setup....10
- How They work all together (frontend & backend)....11
- Docker – setup....11
- Use Case Diagram....12
- ER diagram....13
- Notes to Reviewer....13
- Issues Faced and Future planning....18


## Project Requirement

As the pandemic has led to an increase in the demand for our streaming service, we need to expand the range of our movies, series, and documentaries, which means that we need a newer, more secure, more robust, and scalable web-based application. Our frontend development team has built a frontend website and requires you to complete the application using your fantastic skills in database management, Java development, and REST APIs to make it a fully working website to stream media.

### What was provide

Natflix provides a frontend web page designed to meet the needs, and I will start from it. React project containing the web page you will configure so it can communicate with a backend application created by you. This will allow Netflax to offer a service to stream videos to customers and manage the resources.

In addition, we provide a text file with data about the videos. The data of each video is the title, URL, source, and type of content like series, movies, or documentaries. Fixing the thumbnails and summary (if needed) are your tasks. Your app must enable the features requested in the mandatory requirements.

### Mandatory requirements

We expect you to use Spring Boot to build your application and all REST APIs needed in the program. To manage the data, you must use a relational DBMS (we recommend MySQL). The application must then be containerized using Docker. This is a more complex project. it has 2 separate user roles:

Administrators: people who create, update, delete, and fill the information of each tv-series, movie, and documentary.

Customers: the ones that consume the content.

1. AUTHENTIFICATION

Authentification is the umbrella term for both login and sign-up. Customers can create free accounts and log in using the login page on the website. Administrators must access a secret like on localhost:3000/admin to log in. You cannot create an admin account from the website. make an admin account from the database and provide the login and password in the readme file.

2. PLAY A VIDEO

The application must show a specific amount and category of tv-series, movies, and documentaries. For each movie, there must be a clickable thumbnail, the video length, publication year, brief summary, and a tag if it is a tv series telling the number of episodes.

3. SEARCH

The application must show the best matching results, and not only the videos with matching titles (or creator) must be shown, but also those whose description contains that search word.

4. FILTERED PAGES

In the home page, there must be several options that each takes you to a different category (or genre). The only genres available are:

- Series
- Documentaries
- Movies

In turn, every content is classified as: Horror, Action, Comedy, Drama, Other. This will be used in future iterations of the project, to personalize the content for our customers.

5. ADMIN PANEL

There is a unique menu for the admin, where they can edit the content and add new tv series and movies. Any errors or exceptions must be handled effectively on the backend to avoid crashing the web page or losing data.

**Extra requirements**

After completing the mandatory requirements, these requirements should only be attempted to be developed. Having these extra requirements developed at the expense of the mandatory requirements do not earn you any points. Note these requirements may require modifications on the frontend.

1. HOSTING

Both the backend and the frontend must be deployed on Amazon Web Services.

2. SHOW THE WATCHED HISTORY

A user must be able to review their history of the last 10 watched videos. And provide all watched content with a tag.

3. CUSTOMER OTHER AUTHS

People hate to create new accounts to access a new service, so in order to help Natflex compete against its rival, the mighty Blockbuster, they ask us if we can also add Facebook, Twitter and/or Gmail login.

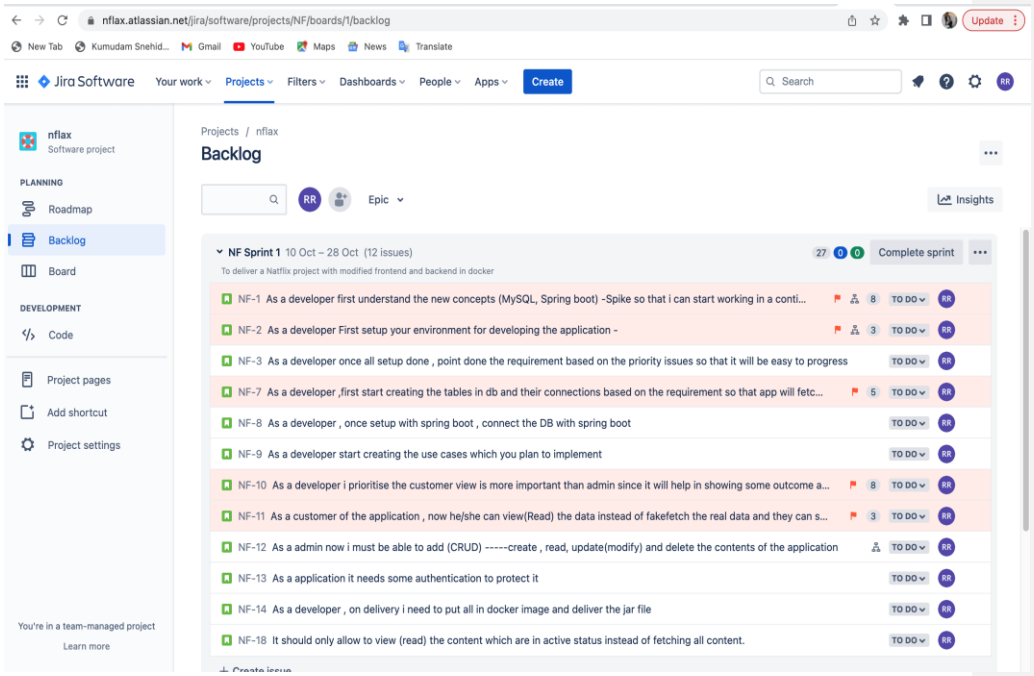
Deliverables

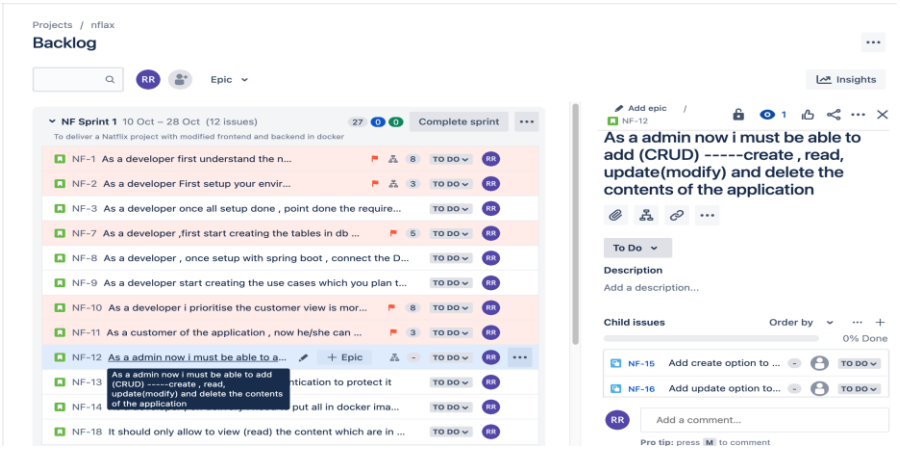
You must deliver your Git repository with a file Main.java as the starting point of the project inside the “src” folder. The .jar file and the container as well as a link to your AWS’s project must be delivered in the same repository

- ## Software Used
    - Jira
    - MySQL Workbench
    - Postman
    - IntelliJ IDEA
    - Docker

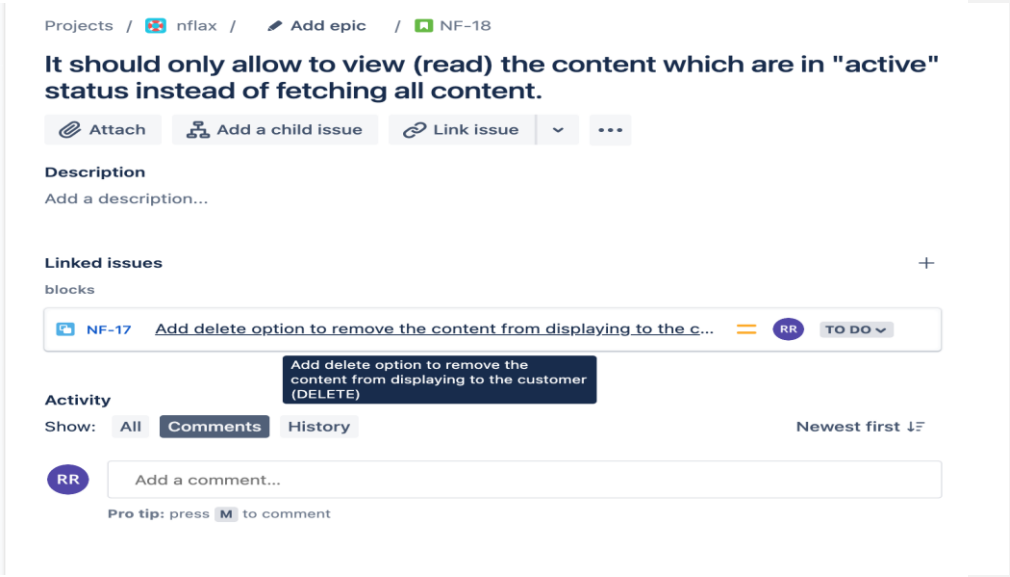
- ## Project Flow -JIRA

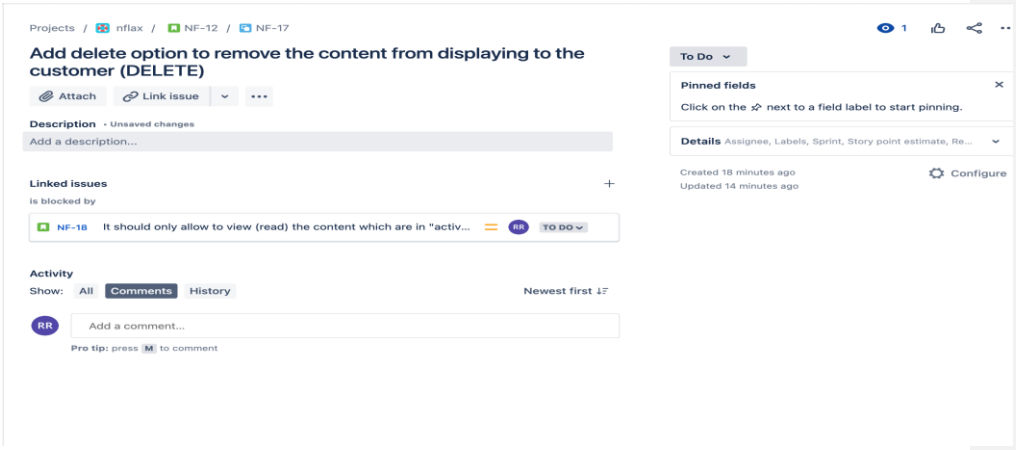
The Project flow started by Jira Planning

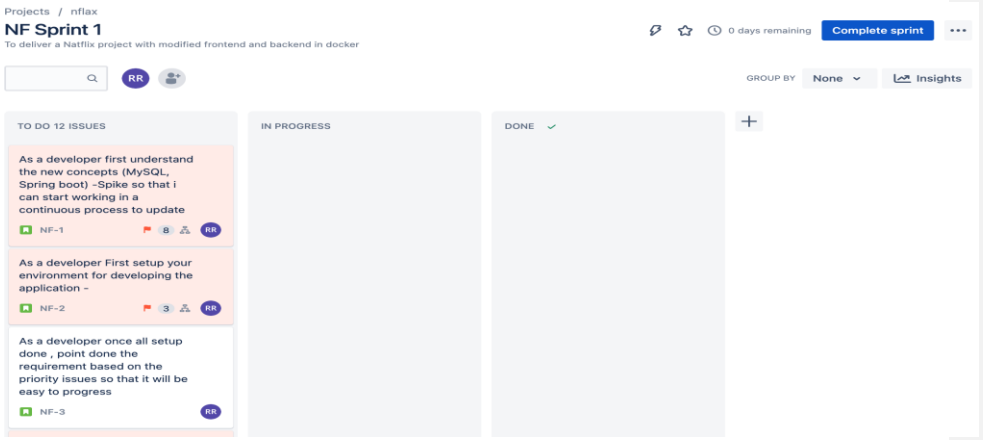


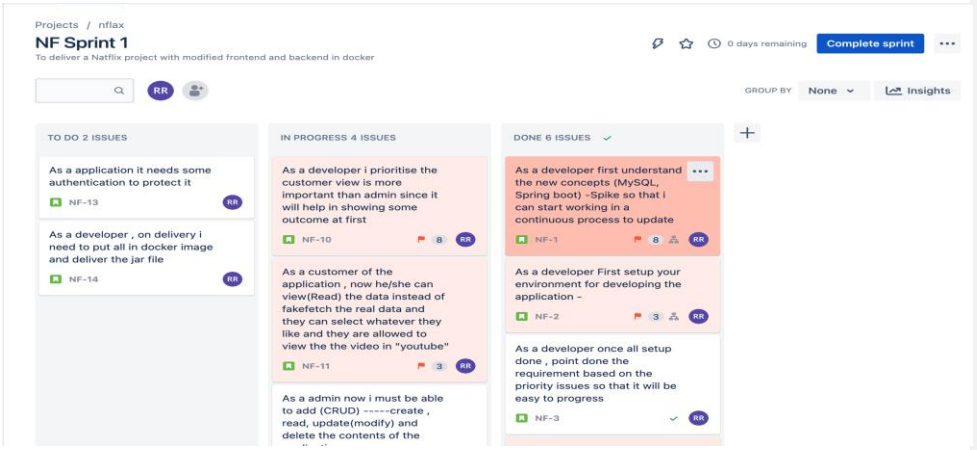



How one task is linked with another task

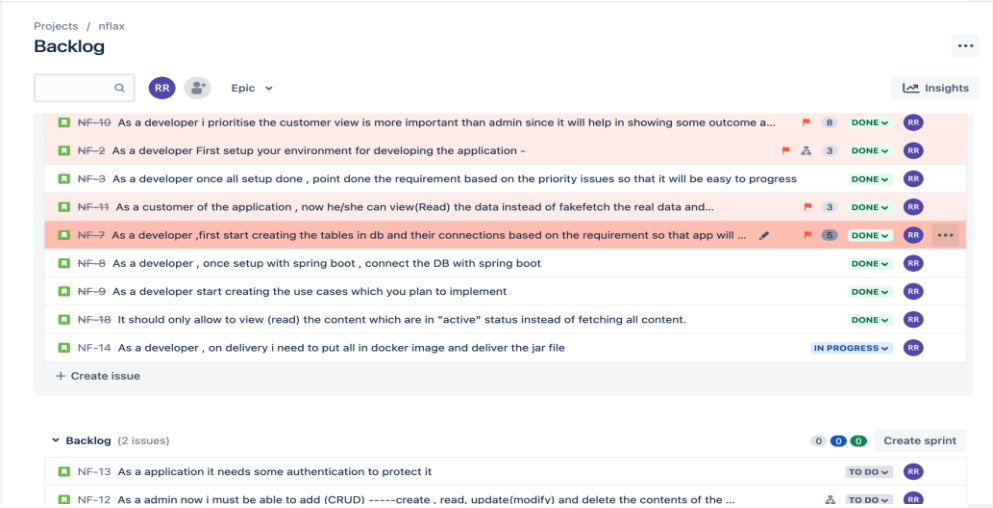

Projects / nflax / Add epic / NF-18
It should only allow to view (read) the content which are in "active" status instead of fetching all content.
Attach
Add a child issue
Link issue
Description
Add a description...
Linked issues
blocks
NF-17
Add delete option to remove the content from displaying to the c...
RR
TO DO
Add delete option to remove the content from displaying to the customer (DELETE)
Activity
Show:
All
Comments
History
Newest first
RR
Add a comment...
Pro tip: press M to comment


Projects / nflax / NF-12 / NF-17
Add delete option to remove the content from displaying to the customer (DELETE)
Attach
Link issue
Description • Unsaved changes
Add a description...
Linked issues
is blocked by
NF-18 It should only allow to view (read) the content which are in "activ...
TO DO
Activity
Show: All Comments History
Newest first
Add a comment...
Pro tip: press M to comment
1
To Do
Pinned fields
Click on the next to a field label to start pinning.
Details Assignee, Labels, Sprint, Story point estimate, Re...
Created 18 minutes ago
Updated 14 minutes ago
Configure


Projects / nflax
NF Sprint 1
To deliver a Natflix project with modified frontend and backend in docker
0 days remaining
Complete sprint
GROUP BY None
Insights
TO DO 12 ISSUES
IN PROGRESS
DONE
As a developer first understand the new concepts (MySQL, Spring boot) -Spike so that i can start working in a continuous process to update
NF-1
As a developer First setup your environment for developing the application -
NF-2
As a developer once all setup done , point done the requirement based on the priority issues so that it will be easy to progress
NF-3


Projects / nflax
NF Sprint 1
To deliver a Natflix project with modified frontend and backend in docker
0 days remaining
Complete sprint
GROUP BY None
Insights
TO DO 2 ISSUES
As a application it needs some authentication to protect it
NF-13
As a developer , on delivery i need to put all in docker image and deliver the jar file
NF-14
IN PROGRESS 4 ISSUES
As a developer i prioritise the customer view is more important than admin since it will help in showing some outcome at first
NF-10
As a customer of the application , now he/she can view(Read) the data instead of fakefetch the real data and they can select whatever they like and they are allowed to view the the video in "youtube"
NF-11
As a admin now i must be able to add (CRUD) -----create , read, update(modify) and delete the contents of the
DONE 6 ISSUES
As a developer first understand the new concepts (MySQL, Spring boot) -Spike so that i can start working in a continuous process to update
NF-1
As a developer First setup your environment for developing the application -
NF-2
As a developer once all setup done , point done the requirement based on the priority issues so that it will be easy to progress
NF-3


TO DO
IN PROGRESS 3 ISSUES
As a admin now i must be able to add (CRUD) -----create , read, update(modify) and delete the contents of the application
NF-12
As a application it needs some authentication to protect it
NF-13
As a developer , on delivery i need to put all in docker image and deliver the jar file
NF-14
DONE 9 ISSUES
As a developer first understand the new concepts (MySQL, Spring boot) -Spike so that i can start working in a continuous process to update
NF-1
As a developer i prioritise the customer view is more important than admin since it will help in showing some outcome at first
NF-10
As a developer First setup your environment for developing the application -
NF-2
As a developer once all setup




- Backend (Spring Boot) - setup

Install MySQL workbench and set root password which are need to set the connection with the spring boot.

IntelliJ IDEA spring boot application was created with maven using the following steps available from [https://start.spring.io/](https://start.spring.io/)

Set up -Maven Project and Java 17 and Spring Web dependency.

Once it's done add Database setup also in pom.xml file for mysql and jpa. Like this

```
<dependency>
   <groupId>mysql</groupId>
   <artifactId>mysql-connector-java</artifactId>
   <scope>runtime</scope>
</dependency>
```

Then in the application.properties  file under (src-main-resources) add the setup for port and database setup to connect the front end with the backend.

```
server.port =8000
spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:mysql://localhost:3306/natflix
spring.datasource.username=root
spring.datasource.password=(your database root password)
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
#spring.jpa.show-sql:true
spring.datasource.validationQuery =Select 1
spring.datasource.testWhileIdle=true
server.servlet.context-path=/api
```

**Commented [RR1]:** this is spring boot  with Database backend port number where u can see json files with data from database table

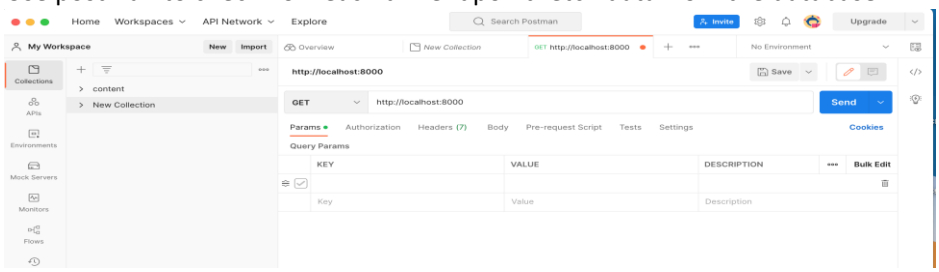
**Commented [RR2]:** to run the front end with backend using this endpoint /api.

- MySQL

Table structure

| content | details_content | -details_series | content_type | content_category | user |
|---|---|---|---|---|---|
| Id = 1,2,3, | Id | Id | Id | Id | Id |
| title = name of series,movie or document | Content id(foreign key) | content id(FK) | type(series, movie, document,) | category(comedy, horror, romantic) | Email |
| type_id =tv-series,document,movie | Video code: | Season number | | | Hashed password |
| category_id = ( comedy, drama, action | | Episode number | | | User_name |
| Summary =( description) | | Series title | | | Password |
| | | Summary | | | |
| | | thumbnail_url | | | |
| | | video_code | | | |

Use postman to check how each diff endpoint fetch data from the database.




- Front end -setup

Front end - [https://github.com/Novare-Potential/Natflix-frontend](https://github.com/Novare-Potential/Natflix-frontend)

Clone the project and  execute the commands

install npm

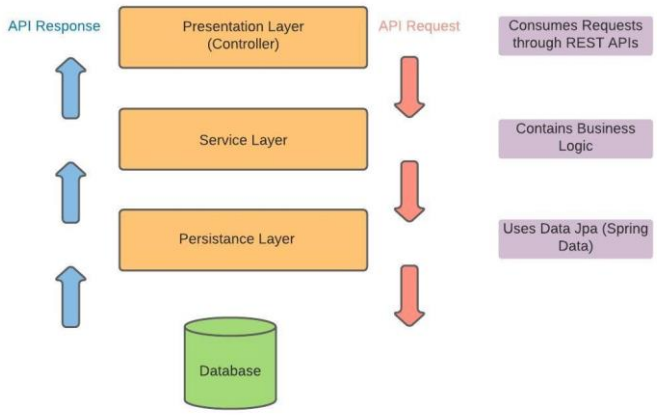
I added proxy to the package.json file in the frontend code and modified the fakefetch which was initially having fake items, which I changed to fetch from the (API)s /endpoints.

"proxy":"http://localhost:8000",

Then type in terminal in the source folder :npm run start

Now the application run in port- http://localhost:3000

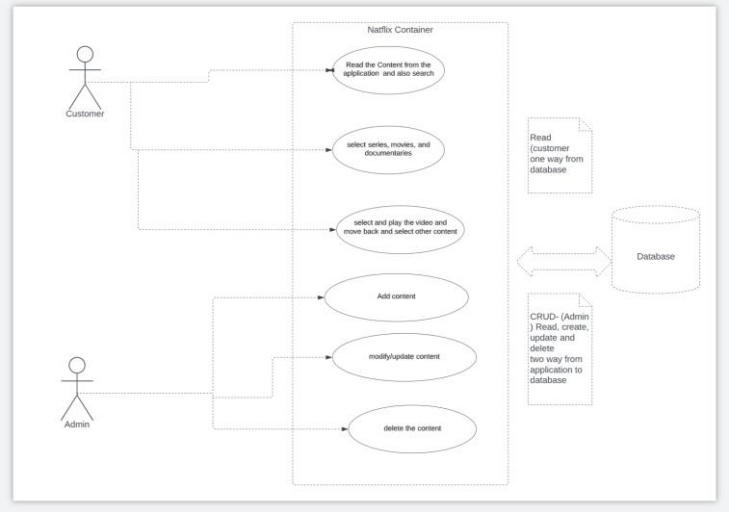
- How They work all together (frontend & backend)



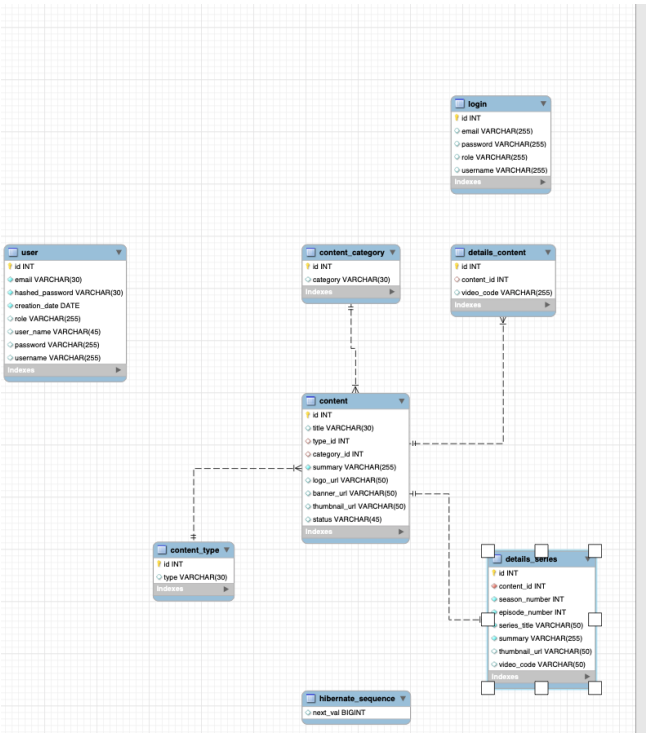

- Docker – setup

  https://github.com/bWNyeQ/novare-docker-compose

- Use Case Diagram

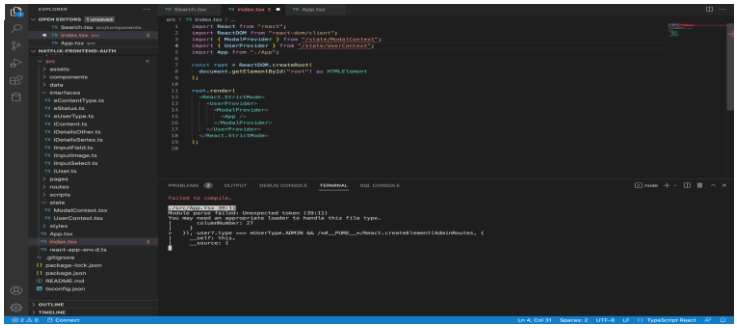


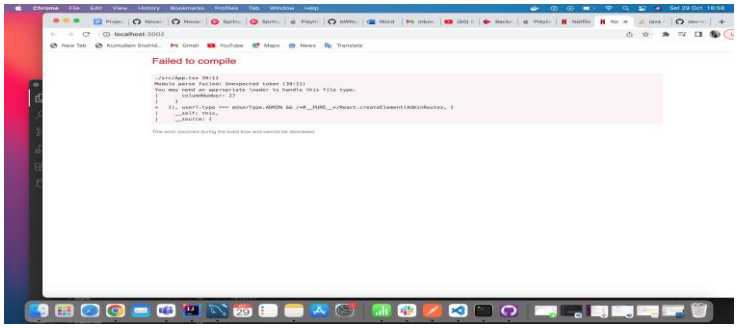
- ER diagram




- Notes to Reviewer

I tried authentication but I faced either page stuck without refreshing and page not found error, then I tried even other updated version I get some kinds of error like this  with path error I fix one I get others ike this ...I tried with my best

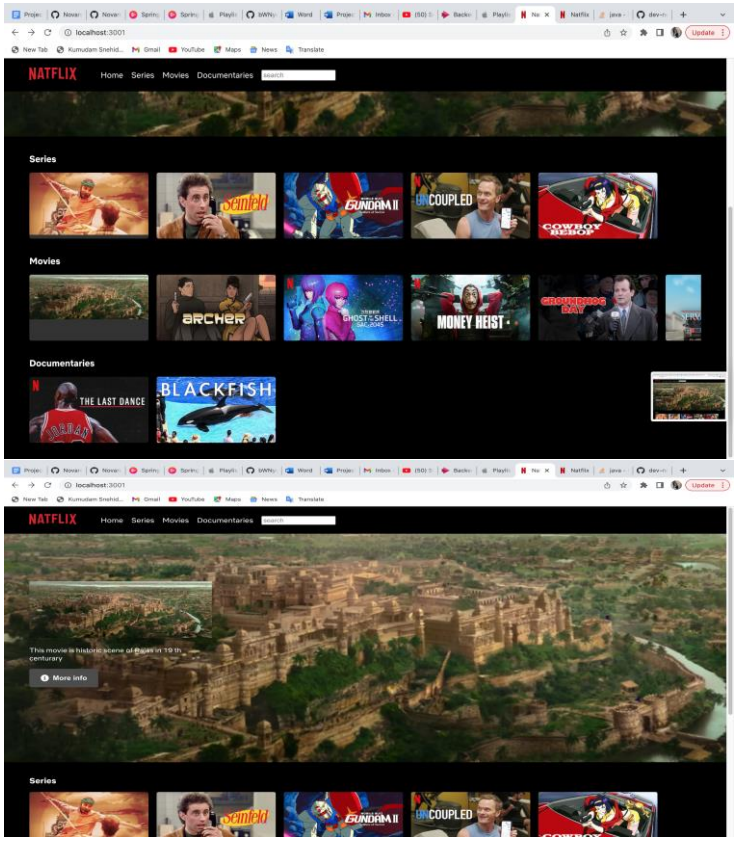




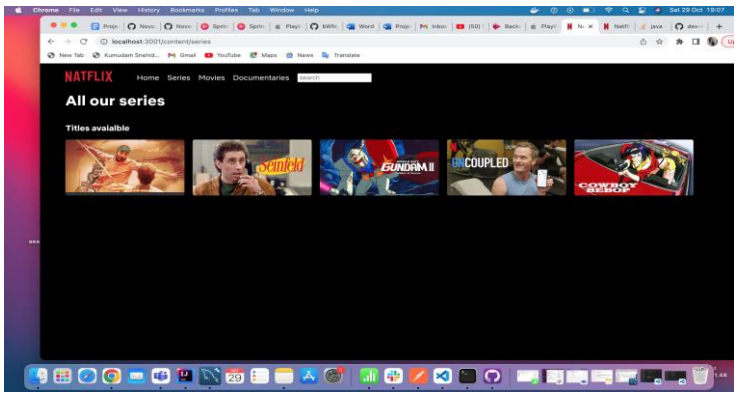

And since authorization was not working, I didn't include role wise access to the CRUD operations in backend.

I did with Content page and also for series, movies and documentaries and search option.

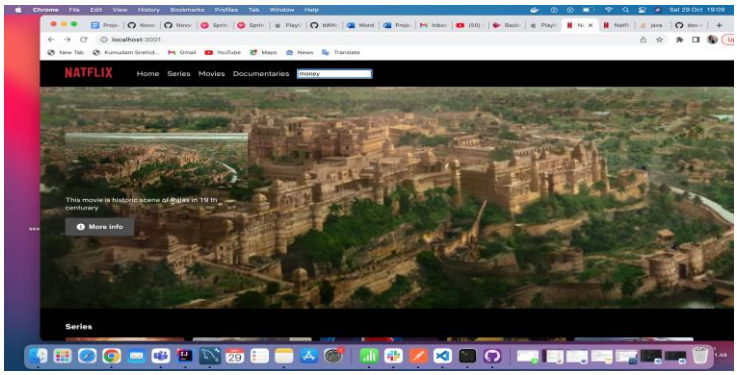
The outcome  for content/Home page




Series Page as well as series category



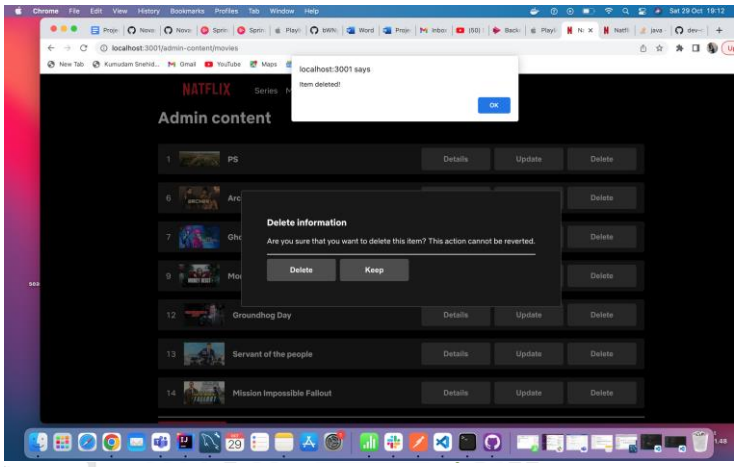

Search – enter search word and  press enter

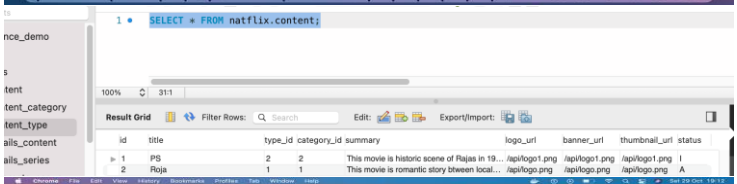


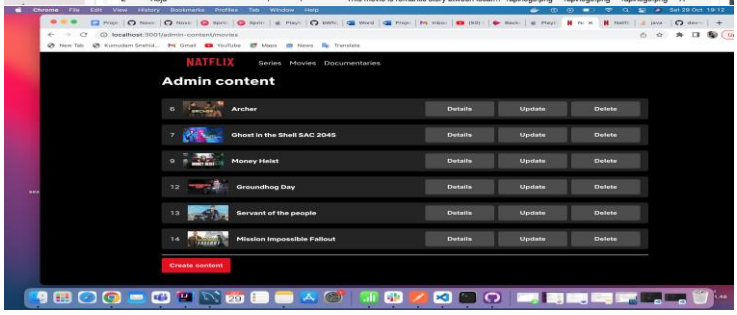


For admin

I have done for delete (I choose to instead of deleting completely I choose to make the status as inactive which will make the movie not available for view)



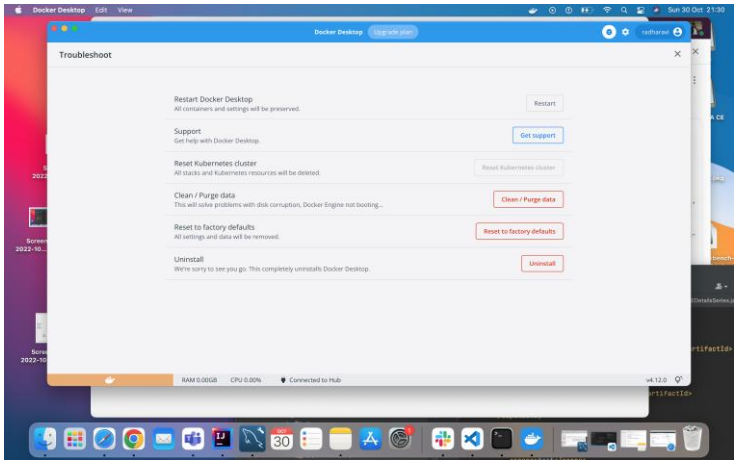



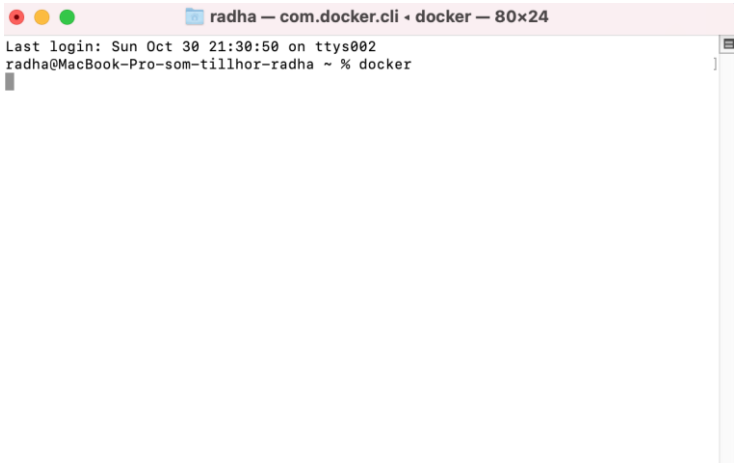
I can able to play the video:


- Issues Faced and Future planning

I am currently trying to do POST and PUT – with more time I would have done and also try to deliver the project with docker.

I tried with docker where I faced issues in mac …




radha — com.docker.cli ◂ docker — 80×24
Last login: Sun Oct 30 21:30:50 on ttys002
radha@MacBook-Pro-som-tillhor-radha ~ % docker