

Angular Front End

Tasks part 1 - Pages

Part 1:

Home component

Login component

Register component

...

Part2:

Movie List page component

card component

Movie Item page component

youtube component

...

Reuse:

NavBar component

Page Not Found component

...

Tasks part 2 - TMDB

TMDB Key:

https://kb.synology.com/en-us/DSM/tutorial/How_to_apply_for_a_personal_API_key_to_get_video_info

TMDB API: <https://developers.themoviedb.org/3/getting-started/introduction>

Tasks part 3 - Auth

Authentication and Authorization

Token strategy: Local storage

Maintain user info in a service

Use Router Guards to control authorization in Angular app.

Request through HttpInterceptor

Tasks for team:

- 1. Create Home Page for Movie App
- 2. Create Register Page for Movie App
- 3. Create Login Page for Movie App
- 4. Fetch data and render cards in movies page
- 5. Design and Create a Movie card, list info poster img and infos on it
- 6. Design and create movie list page.
- 7. Design and implement navbar component (can be use as a header or side navbar)

- 8. infinity scroll for Movie page
- 10. use resolver to hold http request response for next page
- 11. Design and implement movie-item page
- 12. Implement lazy loading for each page
- 13. implement signUp and signIn with our Movie Back end API
- 14. decode jwt and maintain user info in a service
- 15. Refresh browser, auth still work.

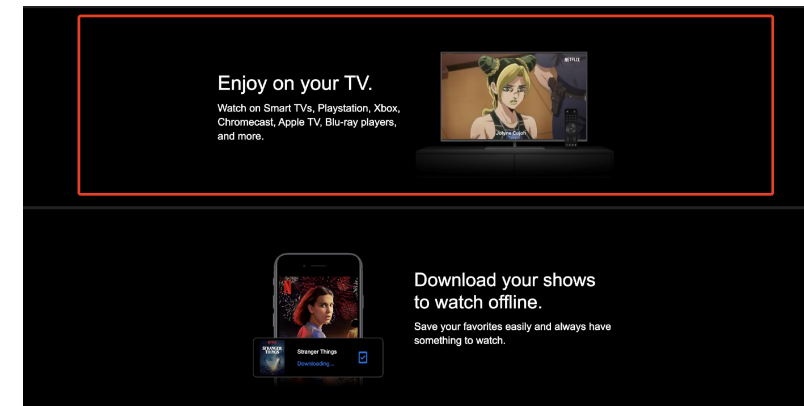
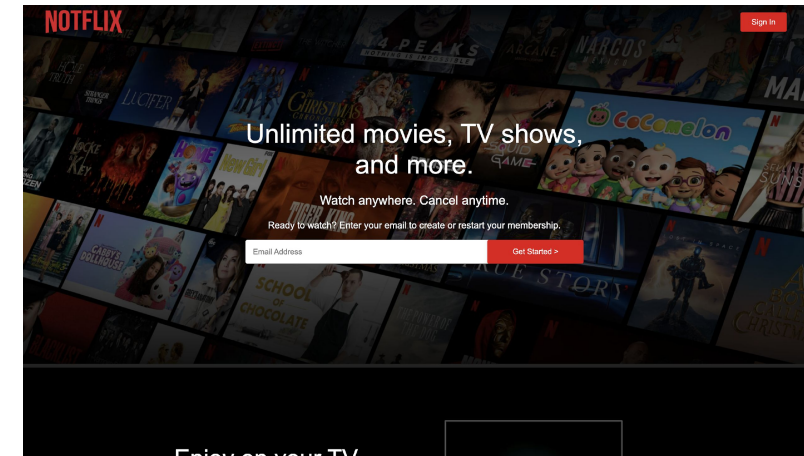
- 16. refresh token
 - 17. add guards for movies list and movies-details page
 - 18. show username if login
-
- 19. user can change his role in front end
 - 20. all these api key should get from jwt token

Basic setting for Our Movie Pages

- A UI library (any UI library as Angular Material)
- Add router and SCSS in our web.

Page1_HomePage

- Create Home Page for Movie App. (Consider Netflix Home page design, or any another design, after your team meeting)
- Create a nav bar on top on of this page, this nav bar should be a reuse component, at least include a logo and a login button.
- Try to split your page to multiple components

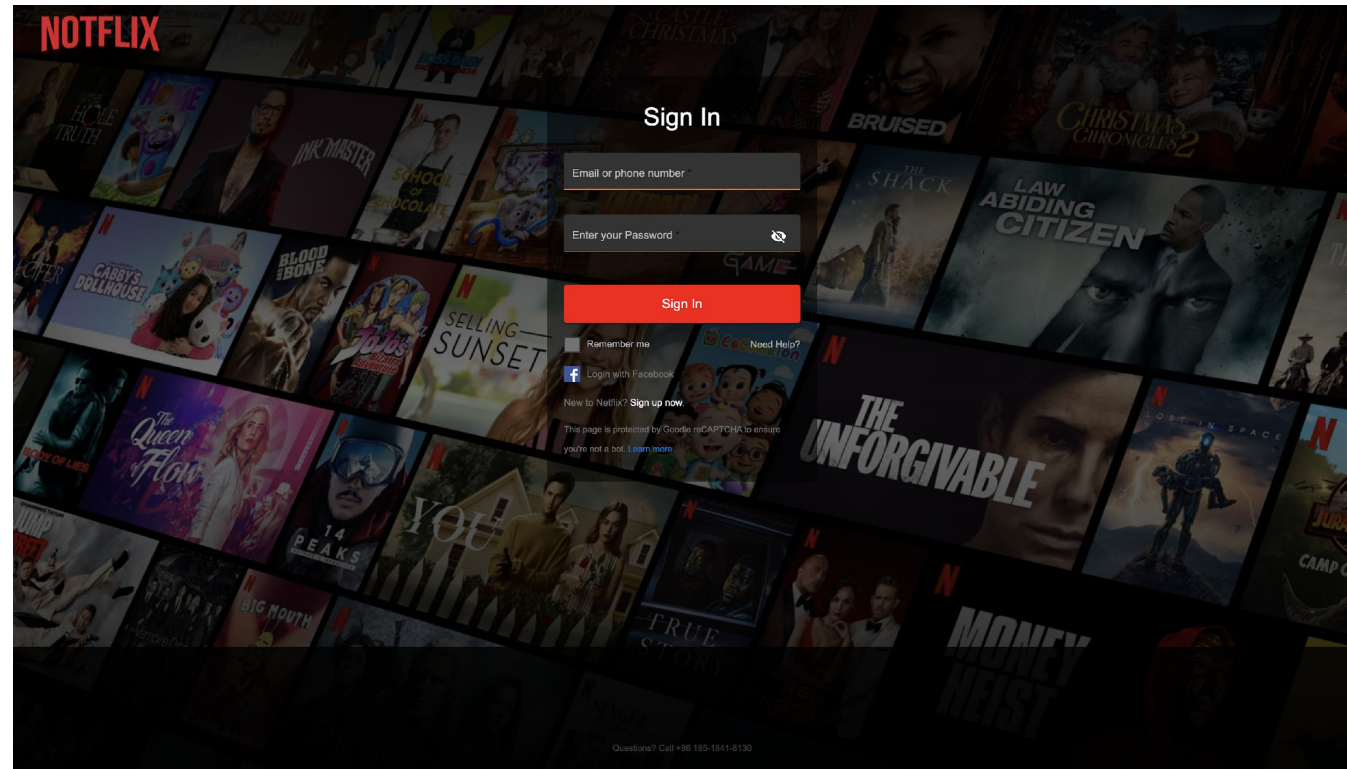


Page2_LoginPage

- Create a login page

This is a standard page, not a popup dialog.

Use *form* element to hold email and password input box.



Page3_RegisterPage

- Create Register component.

Use *form* element to hold following inputs

```
{
  "username": "string",
  "password": "string",
  "email": "string",
  "role": "string",
  "tmdb_key": "stringstringstr"
}
```

Use Multiple pages to create user Signup.
Spatially the page help us create your role(the plan page)

NOTFLIX

Sign Up

STEP 1 OF 3

Create a password to start your membership

Just a few more steps and you're done!

We hate paperwork, too.

Next

NOTFLIX

Sign Up

STEP 1 OF 3

Create a Username & Add your Tmdb Api Key

Just a few more steps and you're done!

We hate paperwork, too.

Next

NOTFLIX

Sign In

STEP 3 OF 3

Choose the plan that's right for you

☒ Watch all you want. Ad-free.

☒ Recommendations just for you.

☒ Change or cancel your plan anytime.

	Basic	Standard	Premium
Monthly price	\$9.99	\$15.49	\$19.99
Video quality	1080p	4K	4K + HDR
Resolution	1080p	4K	4K + HDR
Watch on your TV, computer, mobile phone and tablet	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Next

Fetch Data from TMDB

Apply your tmdb Key from TMDB:

TMDB Key:

https://kb.synology.com/en-us/DSM/tutorial/How_to_apply_for_a_personal_API_key_to_get_video_info

TMDB API: <https://developers.themoviedb.org/3/getting-started/introduction>

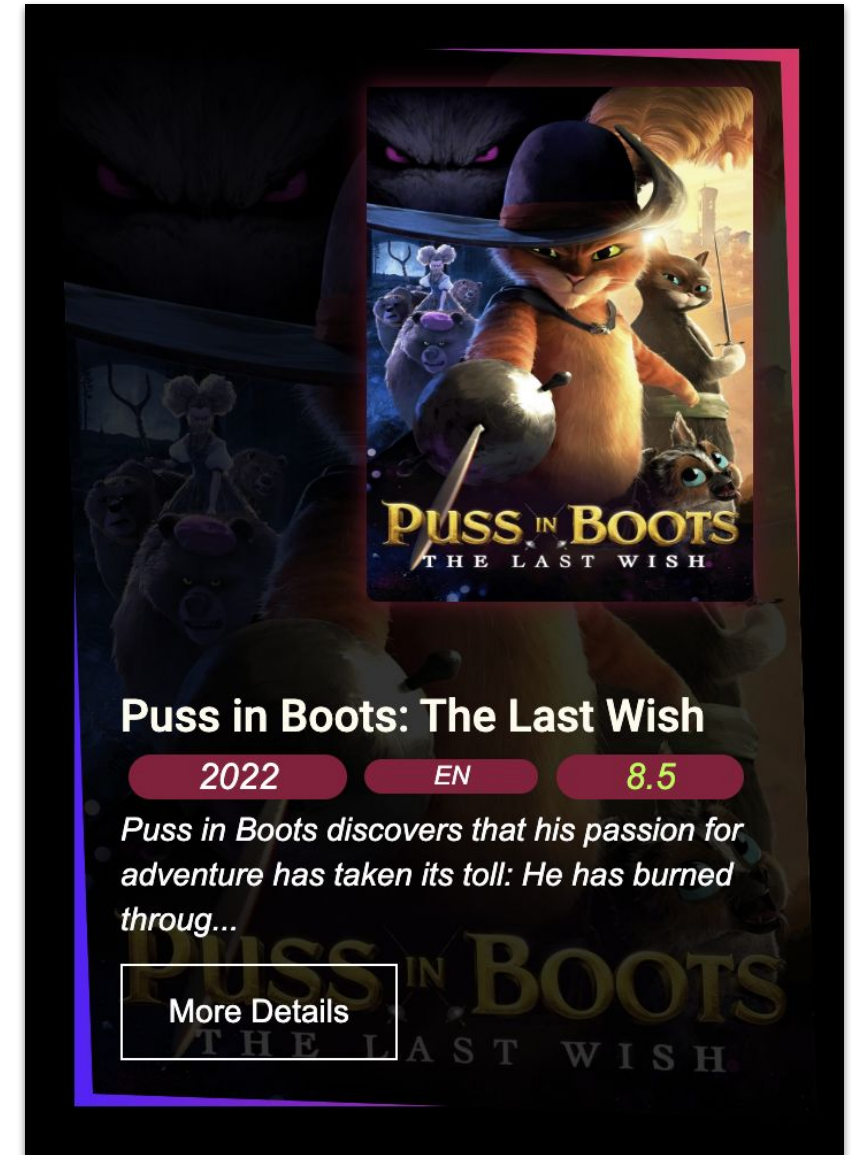
Dig into this api, using discover movie to get a movie list from the API:

<https://developers.themoviedb.org/3/discover/movie-discover>

Create a Movie interface in your project.

Page4_MovieItem

- Create Movie Item component.
 - create a reusable component , by mock data (Movie)
 - get the movie item (Movie) from the Movie List Component.
 - on the Movie component, show the following data on the component:
 - the Poster Image,
 - the movie name,
 - add more info on the Movie component base your movie card design.



Page5_MovieList

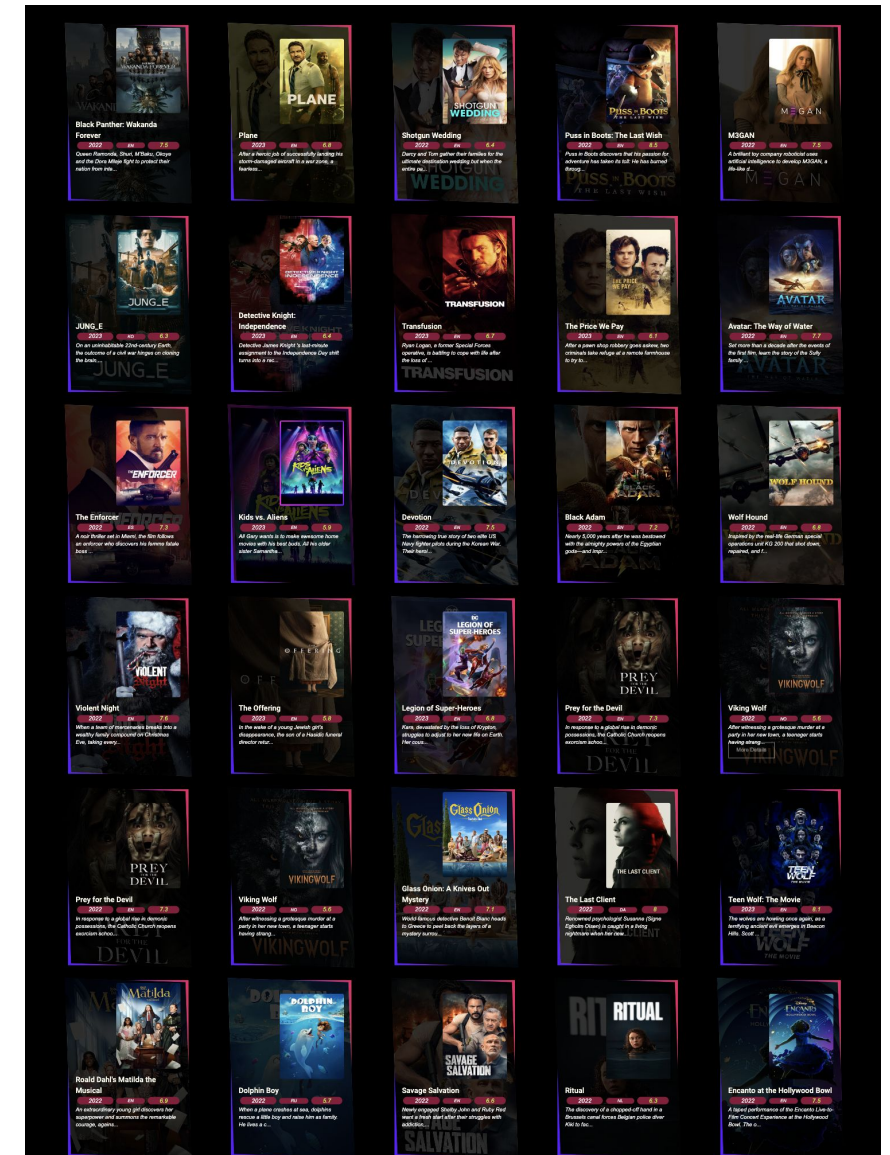
- Create MovieList component.

Create a Movie service to handle all movie data,

- 1) send requests;
- 2) maintain data in it;

Create Movie list component, to render all movie data on it

- 1) trigger requests to get data;
- 2) using FOR loop render multiple MovieItem element on this components.



Install the NestJs Backend

1. `$git clone https://github.com/Show3567/nestjs-training-backend-movie`
2. in this project: `$npm i`
3. `$npm start`
4. You can change the port in main.ts file, line 41.
5. In your browser → Open brief API intro(Swagger): <http://localhost:4231/api/>

Edit your forms as Reactive Form

1. Add ReactiveFormModule in AppModule
2. Edit your ***login/register*** form template, using form builder to create your ***login/register*** form logic.

Sign In

Email or phone number *


Email is required

Enter your Password *

Please enter a valid password
Password is required

Sign In

☐ Remember me [Need Help?](#)

 Login with Facebook

New to Netflix? [Sign up now.](#)

This page is protected by Google reCAPTCHA to ensure you're not a bot. [Learn more.](#)

Add Validators for your Forms

1. Add validators for your *login/register* form.
2. Add async validator for email inputbox.
Use */auth/check-email* API to check if the input email exist

STEP 1 OF 3

Create a password to start
your membership

Just a few more steps and you're done!

We hate paperwork, too.

Enter your email *

Add a password *

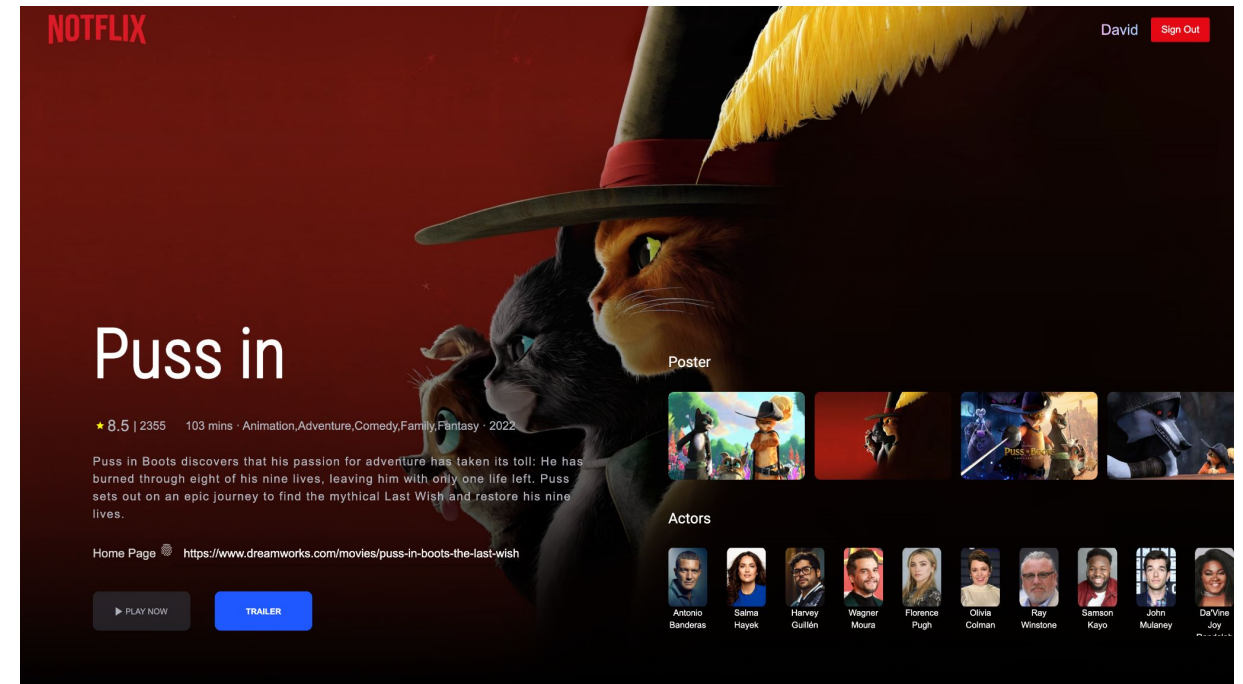
Next

Reflect your project structure & Lazy Loading

1. Create ***Core*** Module and ***Shared*** Module;
2. Create Module for each page in your application;
3. Lazy load all these page modules
4. Create sub route list in these page modules if necessary.

Page6_MovieDetails

1. Switch to movie details page when click the movie item card in movie list page.
2. Get the movie *id* from router, send request to get more movie info for this page:
<https://developers.themoviedb.org/3/movies/get-movie-details>
3. Design and Create movie details page to show more details for a movie.



Create YouTube player to show trailer

1. Dig into this lib to get the official youtube component:
<https://www.npmjs.com/package/@angular/youtube-player>
2. On the Movie Details page, click a button to launch movie trailer from YouTube on a pop up window. (angular Material or some lib)
<https://developers.themoviedb.org/3/movies/get-movie-videos>
3. The dialog can switch different trailer(you have a trailer list from the movie/video api)

