

# Movie App

## Main Idea

User can browse the movie app and able to see movies and see movie details in addition to add movies to wishlist.

## Pages

Create an angular app with the following pages:

- Movies list page ( main page ).
- Movie details.
- Movies wishlist.
- Movies Search Results.

## Features

User can view the list of movies by using the following api :

[https://api.themoviedb.org/3/movie/popular?api\\_key={apiKey}](https://api.themoviedb.org/3/movie/popular?api_key={apiKey})

User can paginate between the movies list pages by sending a query param to the movies list api like the following:

[https://api.themoviedb.org/3/movie/popular?api\\_key={api\\_key}&page=4](https://api.themoviedb.org/3/movie/popular?api_key={api_key}&page=4)

User can view the movie details page using the following api :

[https://api.themoviedb.org/3/movie/{id}?api\\_key={apiKey}](https://api.themoviedb.org/3/movie/{id}?api_key={apiKey})

And can view the recommended movies in the details using the following api:

[https://api.themoviedb.org/3/movie/{movie\\_id}/recommendations?api\\_key={apiKey}](https://api.themoviedb.org/3/movie/{movie_id}/recommendations?api_key={apiKey})

User can search movies and redirect to a new page to view search results using the following api :

[https://api.themoviedb.org/3/search/movie?api\\_key={api\\_key}&query={MovieName}](https://api.themoviedb.org/3/search/movie?api_key={api_key}&query={MovieName})

User can add movie to wishlist and when added should increase the counter of wishlist in navbar, and heart icon should be filled with the main website color

User can toggle the wishlist action on the movie card, if movie added to wishlist and user clicked on the heart icon again should remove movie from wishlist, if movie not added to wishlist and clicked on the heart icon should add movie to wishlist.

User can visit the wishlist page when click on the wishlist in navbar and will redirect to the page, Wishlist page will contain all the movies added to wishlist and user can remove it from wishlist

#### [BONUS]

- Use the following api to add and remove movie to watchlist :

**[POST]** ] [https://api.themoviedb.org/3/account/{account\\_id}/watchlist](https://api.themoviedb.org/3/account/{account_id}/watchlist)

For post request you need to send a json body

```
{  
  "media_type": "movie", //keep this with movie  
  "media_id": 11, // which is movie id  
  "watchlist": true // this will toggle with true or false you add or remove from  
watchlist  
}
```

- Use the following api to add and remove movie to watchlist :

**[GET]** [https://api.themoviedb.org/3/account/{account\\_id}/watchlist/movies](https://api.themoviedb.org/3/account/{account_id}/watchlist/movies)

UI Reference :

<https://www.figma.com/file/jvGvsGLg6X3T7JPU3E2rNI/Movie-App?type=design&node-id=0%3A1&mode=design&t=GaXQ9V2yV3EruZxZ-1>

#### **General Information:**

- Create account on the movie database website : <https://www.themoviedb.org/login>.
- Use the API key provided by this website with each API for authentication.
- To render images you need to append the base part of the image to the poster\_path like the following : [https://image.tmdb.org/t/p/w500/\\${poster\\_path}](https://image.tmdb.org/t/p/w500/${poster_path})

APIs docs link: <https://developer.themoviedb.org/reference/movie-popular-list>

## Requirments

You should use the angular modules that helps you deliver the needed requirements.

- Create a reusable components
- Organize your code using modules ( feature modules and shared module )
- Share data between related components
- Services to share data between unrelated components
- Services to separate http requests
- Use interface
- Use pipes to transform data
- Create needed routes and handle 404 page

## Delivery

- Provide a sheet with the features and its initial estimate and the actual spent time.  
Sheet Link (Make a copy of this sheet)  
<https://docs.google.com/spreadsheets/d/1NRvrJ8pSKXGn2Y4JCvZKdcJJBIVlGqiewtQYZrIkH2U/edit?usp=sharing>
- Project should be linked with github and should push commits per feature and deliver the github link
- Project should be hosted using ( vercel, netlify or github pages ) whatever is easy to you and provide the link

## Timeline

Project should be done and submitted withing **5 days**.

*Evaluation will be based on the code quality and the working functionality and github history commits during your work.*

*Best of luck,*