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. • Login + Register + Account Details.

Features
User can view the list of movies by using the following api : *https://api.themoviedb.org/3/movie/now_playing?api_key={apiKey}*
User can view the movie details page using the following api : *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}*
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
Notifications (Snackbar): how user feedback via Angular Material Snackbar.
Genre Filtering: Use https://api.themoviedb.org/3/genre/movie/list?api_key=... to show genres.
Allow filtering on home/search page by selected genres. Movie Sorting: Sort movies by rating, release date, popularity using query param sort_by like /discover/movie?sort_by=popularity.desc
[Bonus] Lazy Loading Routes: Convert modules to be lazy-loaded
Movie Trailer Embed: On details page, fetch /movie/{id}/videos and embed official trailer using YouTube iframe if type == Trailer.
Dark Mode Toggle + Loading Spinner + Dynamic Page Title + Back to Top Button + Skeleton

User can register using email/username +password stored in LocalStorage, or using Firebase
[Bonus] Login with TMDB Account: Use TMDB authentication (token + session ID) to login and allow users to mark "Favorites" or "Watchlist" via real TMDB account.
Favorites List using TMDB API: After auth, use real /account/{account_id}/favorite/movies API.
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*
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}*
User can change the language from a language dropdown that contains the following languages [‘en’, ‘ar’,’fr’, ‘zh’], when user select an new language should request the API with the updated language query param https://api.themoviedb.org/3/movie/now_playing?api_key={apiKey}&language={language} Also change the content direction from ltr to rtl if user selected ar as language else should direction remain ltr

UI Reference :
https://www.figma.com/file/jvGvsGLq6X3T7JPU3E2rNl/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}

APIs docs link: https://developer.themoviedb.org

Requirments

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

- Create a reusable components
- 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/1NRvrJ8pSKXGn2Y4JCvZKdcJJBlVlGqiewtQYZrlkH2U/edit?usp=sharing
- Project should be linked with github and should push commits per feature and deliver the github link, add me to the repo => My GitHub username: hassaneldash
- 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 within max **10 days**. [Starting from 1/10 , deadline 10/10 to deliver a send email].

Please send the final sheet on the following email: hassanmeldash@gmail.com, and email subject should be: **ITI BeniSuef | .net46R1 | Angular Project | Movie App | Group No.?**

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

Best of luck,