Movie App

A group of users is sharing their favorite movies using a mobile application. Each user is able to manage their own movies.

On the server side, at least the following details are maintained:

- Id - the internal movie id. Integer value greater than zero.
- Name - the movie name. A string of characters representing the movie name.
- Description - the movie description. A string of characters.
- Genre - the movie genre. A string representing the genre of the movie.
- Director - the movie director. A string representing the director of the movie.
- Year - the movie release year. An integer value.

The application should provide at least the following features:

- Main Section (separate activity)

  A. (1p) View the genres available in the system in a list. Using the **GET /genres** call, the user will retrieve the list of all movie genres in the system. If offline, the app will display an offline message and a way to retry the connection and the call. Once retrieved it should be available, even offline.

  B. (2p) By selecting a genre, the user will be able to get to the list of movies that belong to that genre. The **GET /movies** call can be used by specifying the genre to retrieve the list of movies belonging to the specified genre. Once retrieved the list should be available, even offline.

  C. (1p) Add a movie. Using **POST /movie** call by specifying all the movie details the user will be able to create a new movie. Available online only.

  D. (1p) Delete a movie. By selecting a movie from the list, and using the **DELETE /movie** call, the user will be able to delete a movie. Available online only.

- Release Year Section (separate activity)

  A. (1p) View the list of active years. The list will include for each year and the number of released movies, ordered descending by the number of movies. From the server, using **GET /all** calls, all the existing movies will be retrieved.

  B. (1p) View the most popular genres. Using the same **GET /all** call determine and display the top 3 genre that contains the most number of movies.

- (1p) On the server side once a new movie is added to the system, the server will send, using a WebSocket channel, a message to all the connected clients/applications with the new movie object. Each application, that is connected, will display the received movie details, in human form (not JSON text) using an in-app “notification” (like snackbar or toast or a dialog or a message on the screen).

(0.5p) On all server operations, a progress indicator will be displayed.

(0.5p) On all server interactions, if an error message is received, the app should display the error message using a toast or snackbar. A log message should be recorded on all interactions (server or DB calls).