Gardening App

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

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

- Id - the internal tip id. Integer value greater than zero.
- Name - the tip name. A string of characters representing the tip name.
- Description - the tip description. A string of characters.
- Materials - the materials needed for the tip. A string represents the materials needed for the tip.
- Steps - the steps for the tip. A string representing the steps for the tip.
- Category - the tip category. A string of characters. Eg. “flower gardening”, “vegetable gardening”, “pest control”, etc.
- Difficulty - the tip difficulty. A string of characters. Eg. “easy”, “medium”, or “hard”.

The application should provide at least the following features:

- Main Section (separate activity)
  A. (1p) View the categories available in the system in a list. Using the **GET /categories** call, the user will retrieve the list of all tip categories found 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 offline.
  B. (2p) By selecting a category, the user will be able to get to the list of tips that are having that category. To retrieve the list of tips having the specified category the **GET /tips** call can be used by specifying the category. Once retrieved the list should be available offline.
  C. (1p) Add a tip. Using **POST /tip** call by specifying all the tip details the user will be able to create a new tip. Available online only.
  D. (1p) Delete a tip. By selecting a tip from the list, and using the **DELETE /tip** call, the user can delete a tip. Available online only.
- Difficulty Section (separate activity)
  A. (1p) The list of the top 10 easiest tips sorted ascending by difficulty and category. The list will be retrieved using the **GET /easiest** call, in this list along with the name, materials, and steps, the app will display the current difficulty and the category. Note that from the server you are retrieving all the tips.
  B. (1p) Change the difficulty level of a tip. From the above list, the user should be able to select a tip and change its difficulty level using the **POST /difficulty** call by specifying the tip id and the new difficulty level.
- (1p) On the server side once a new tip is added to the system, the server will send, using a WebSocket channel, a message to all the connected clients/applications with the new tip object. Each application, that is connected, will display the received tip 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. On all interactions (server or DB calls), a log message should be recorded.