Event Management System

In an event management system, users can access a mobile application to explore and engage with various events. Organizers, as employees, can manage the event inventory.
On the server side, at least the following details are maintained:

- Id: An internal identifier for each event. Integer value greater than zero.
- Name: The name of the event. A string of characters representing the event title.
- Organizer: The name of the event organizer. A string of characters.
- Category: The category of the event. A string of characters.
- Capacity: The maximum number of attendees for the event. An integer value.
- Registered: The number of attendees registered for the event. An integer value.

The application should provide the following features (available without restarting the app):

● Client Section (separate activity/screen)

A. (2p) Users can view a list of available events in the system in a list. Using **GET /events** call, the user will retrieve the list of all events in the system. Only the event name, organizer, and category should be displayed for each event. Once retrieved, the server call should not be performed again, and the data should be available on the device, regardless of whether it is online, offline, or restarts.

B. (1p) By selecting an event from the list, users can reserve a spot by specifying the event id using **PUT /reserve** call. This action is available only while online.

C. (1p) List Reserved Events: Users can view the list of reserved events using **GET /reserved** calls. Once retrieved, the server call should not be performed again, and the data should be available on the device, regardless of whether it is online, offline, or restarts.

D. (1p) Attend an Event: Users can attend a reserved event by using **PUT /attend** call and specifying the event id from the reserved list. This action is available only while online.

● Employee Section (separate activity/screen) - Available online only.

A. (1p) View All Events: Employees can view the list of all events available in the system, sorted by category and capacity. The list should be presented in ascending order by capacity. Using the same **GET /events** call. The call should be performed each time when the employee enters this section.

B. (1p) Add a New Event: Employees can add a new event to the system by using **POST /event** call and sending all the event details.

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

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

(0.5p) On all server or DB 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).