Event Management App

An event organizing company is managing its events using a mobile application. Event organizers record event details, and participants can view and engage with event information.
On the server side, at least the following details are maintained:

- Id - the internal event id. Integer value greater than zero.
- Name - A string of characters representing the event name.
- Team - A string of characters representing the event organizing team name.
- Details - A string of characters having the event details.
- Status - A string of characters representing the event status. Eg. “planning”, “in progress”, “on hold”, “completed”, etc.
- Participants - An integer value representing the number of participants involved in the event.
- Type - the event type. A string of characters. Eg. “conference”, “concert”, etc.

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

● Event Organizer Section (separate activity/screen)

A. (1p) Create a new event. Using **POST /event** call by specifying all the event details. Available both online and offline.

B. (2p) View all the events in the system, in a list. Using **GET /events** call, the event organizer will retrieve all of them. The list should display at least the id, name, team, and type. If offline, the app will display an offline message and a way to retry the connection and the call. Once retrieved, the data should be available on the device, regardless of whether online, offline, or restarts.

C. (1p) By selecting an event from the list, the event organizer will be able to view all the event details. To retrieve all the event details, **GET /event** call will be used by specifying the event id. Once retrieved, the data should be available on the device, regardless of whether online, offline, or restarts.

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

A. (1p) View all the events that are in progress in the system in a list. Using **GET /inProgress** call, the participant will retrieve all the events having this status.

B. (1p) By selecting an event from the list, the participant will be able to enroll in the ongoing event. Using **PUT /enroll** by specifying the event id, the user will be able to enroll in the event.

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

(1p) Using **GET /allEvents** call, create a top 5 events by the number of participants. The list will present the events in ascending order by status and descending by the number of participants. If no such events are available, the application will display a proper message. Also, note that the server is not ordering the list in any way.

● (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).