

Problem Statement:

You are tasked with developing a Kotlin based Android app using the MVVM architecture, Retrofit for API communication, and ROOM for local data storage. The assignment involves fetching and displaying posts from the placeholder API: `https://jsonplaceholder.typicode.com/posts``.

Requirements:

1. API Integration:

Use Retrofit to fetch a list of posts from the provided API endpoint.

The API returns a JSON array of post objects, each containing ``userId``, ``id``, ``title``, and ``body`` fields.

2. Data Model:

Create a data class, ``Post``, to represent the post model with fields ``userId``, ``id``, ``title``, and ``body``.

3. Database Setup:

Implement ROOM to store the fetched posts locally.

Define a DAO (Data Access Object) for managing post data.

4. Repository:

Develop a repository class to handle data operations. This class should fetch data from the API, store it in the local database, and provide data to the ViewModel.

5. ViewModel:

Create a ViewModel class to interact with the repository and expose the list of posts to the UI.

6. UI:

Design a simple Android UI with a RecyclerView to display the list of posts and details of post.

Each RecyclerView item should show the post's title and body.

7. Offline Mode:

Develop a mechanism to handle offline scenarios gracefully. If the app is launched without an internet connection, display locally cached posts. Implement a smart sync strategy to update the local database when the device regains connectivity.

Note:

Consider best coding practices, maintainability, and code readability. Pay attention to error handling and edge cases, especially when dealing with asynchronous Retrofit and ROOM operations.