**Task: Create a Note-Taking App with Basic Room Functionality**

**Prerequisites**

* Basic familiarity with Android development, Java, and Room concepts.
* A development environment (Android Studio) set up.

**Functionality**

* **Add a Note:** The user can enter a note's title and content and save it to the local database.
* **View Notes:** Display a list of existing notes in a visually organized manner.
* **Edit a Note:** The user can modify the title and content of an existing note and save the updates.
* **Delete a Note:** The user can permanently delete a note.

**Steps**

1. **Project Setup**
   * Create a new Android Studio project.
   * Add Room dependencies to your build.gradle file.
2. **Define the Note Entity**
   * Create a Note.java class.
   * Include properties: id (primary key), title, content, and timestamp.
   * Annotate this class with the relevant Room annotations (@Entity, @PrimaryKey, etc.).
3. **Create the Note DAO**
   * Create a NoteDao.java interface.
   * Define methods:
     + insertNote(Note note)
     + updateNote(Note note)
     + deleteNote(Note note)
     + getAllNotes(): LiveData<List<Note>>
4. **Build the Database Class**
   * Create a NoteDatabase.java class that extends RoomDatabase.
   * List the Note entity.
   * Provide an abstract method to get an instance of NoteDao
   * Implement a singleton pattern to ensure a single database instance.
5. **Create the UI (XML Layouts)**
   * activity\_main.xml**:** Design a layout with a RecyclerView to display the list of notes and a Floating Action Button (FAB) to trigger note creation.
   * activity\_add\_note.xml**:** Design a layout with EditText fields for title and content, and buttons for saving or updating notes.
   * list\_item\_note.xml**:** Design a layout representing a single note item in your list.
6. **Create Activities**
   * MainActivity.java:
     + Set up RecyclerView and an adapter.
     + Instantiate a ViewModel (see step 7).
     + Observe LiveData of notes and update the RecyclerView.
     + Handle FAB clicks to launch AddNoteActivity.
   * AddNoteActivity.java:
     + Get references to UI elements (EditTexts, Buttons)
     + In button click listeners, retrieve data, create a Note object, and call relevant ViewModel functions.
7. **Create a ViewModel**
   * Create NoteViewModel.java
   * Instantiate a repository (see step 8).
   * Expose LiveData<List<Note>> to the UI.
   * Define functions to insert, update, and delete notes, interacting with the repository.
8. **Create a Repository**
   * Create NoteRepository.java
   * Provide an abstraction over your Room database interactions.