

TODO List app

✓ Project: To-Do List App (Jetpack Compose Edition)

🔧 Core Features

- **Task Management:** Users can create, view, update, and delete to-do items.
 - **Task List Screen:** Shows all tasks in a scrollable list.
 - **Add/Edit Task Screen:** For adding new tasks or modifying existing ones.
-

🧱 Tech Stack

- **Room Database:**
 - Handles **local storage** for tasks.
 - Provides efficient, asynchronous database operations using Kotlin coroutines.
 - Defines an `Entity`, `DAO`, and `Database` setup for structured data persistence.
 - **Jetpack Compose UI:**
 - Uses **Composable functions** to build UI declaratively.
 - **LazyColumn** to efficiently render task items in a list.
 - Supports modern, reactive UI design.
 - **Navigation Compose:**
 - Enables smooth navigation between the **Task List** and **Add/Edit Task** screens.
 - Easily handles back stack and argument passing (e.g. `taskId` for editing).
 - **ViewModel + LiveData** (*Bonus/Best Practice*):
 - **ViewModel** manages UI-related data and handles logic.
 - **LiveData** or **StateFlow** observes data changes and updates the UI reactively.
 - Keeps UI state across configuration changes like screen rotation.
-

📌 Why This Is a Solid Project

- Covers essential concepts: **CRUD**, **local database**, **state management**, and **UI composition**.
- Encourages **MVVM architecture**—clean separation of concerns.

- Scalable: great base for adding features like due dates, categories, or alarms later on.