

To-Do List Mobile App with Offline Functionality

Overview

This mobile application is a simple, yet functional, To-Do List app designed to work fully offline. Users can add, edit, delete, and mark tasks as completed. Additionally, tasks can be sorted based on their deadlines. The app is built to run on both Android and iOS platforms.

Core Features

1. Task Management

- **Add Task:** Users can create a new to-do item with a title, description, and deadline.
- **Edit Task:** Users can modify the details of an existing task.
- **Delete Task:** Users can remove a task from the list.
- **Mark as Completed:** Users can mark tasks as done, which then updates the task's status in the list.

2. Deadline Sorting

- Tasks are automatically sorted based on their deadlines, with the most urgent tasks appearing at the top.

3. Offline Functionality

- The app fully functions offline by storing data locally on the device using SQLite. Users can continue to manage their tasks even without an internet connection.

Custom Feature

Gesture-Based Task Management

- **Swipe Gestures:** Users can swipe left or right on a task to reveal quick actions, such as delete or mark as completed.
- **Drag-and-Drop Reordering:** Tasks can be rearranged in the list using drag-and-drop, allowing users to prioritize tasks visually.

User Interface

Design

- The app features a clean and intuitive UI, with a focus on simplicity and usability.
- Basic animations are implemented when adding or deleting tasks to enhance the user experience.

Custom UI Component

- A collapsible category view allows users to organize tasks into categories. Each category can be expanded or collapsed, making it easier to manage large numbers of tasks.

Technology Stack

- **Framework:** Flutter
- **Local Storage:** SQLite (for offline data persistence)
- **Animations:** Flutter's built-in animation framework for smooth transitions

Code Structure

1. Main Components

- **Task Model:** Represents the structure of a task, including title, description, deadline, and status.
- **Task List View:** Displays the list of tasks, allowing users to interact with them.
- **Task Editor:** Interface for adding and editing tasks.

2. Storage Management

- **SQLite Database:** Used to store tasks locally on the device. The database is queried to fetch, add, update, and delete tasks as needed.
- **Data Provider:** Manages the interaction between the app and the SQLite database, ensuring that the data is always in sync.

3. UI Components

- **Task Item Widget:** A reusable widget that displays individual tasks in the list.
- **Category View Widget:** Custom collapsible component for organizing tasks into categories.

Challenges and Solutions

1. Offline Data Persistence

- **Challenge:** Ensuring the app works fully offline and data is reliably saved.
- **Solution:** Implemented SQLite for local storage, which efficiently handles the storing and retrieval of task data.

2. Gesture-Based Interactions

- **Challenge:** Designing a natural and intuitive swipe-based interaction for task management.
- **Solution:** Used Flutter's Dismissible widget for swipe actions and implemented custom drag-and-drop functionality.

Conclusion

This To-Do List app demonstrates a simple yet effective approach to task management on mobile devices, with a strong emphasis on offline functionality and an intuitive user experience.