To-do list(using Sqlite)

Workshop #2

Flutter Developer Bootcamp

# **Purpose**

This workshop demonstrates how to integrate SQLite database functionality into a Flutter application for storing, retrieving, updating, and deleting data. It's a practical example for developers looking to implement database operations in their Flutter apps.

# **Problem**

In the given workshop, there will be a TodoList app containing a FloatingActionButton for adding data. You need to create a new task field for adding data instead of the FloatingActionButton, and add update and delete options for each data entry.

# **How to Solve**

1. Checkout the workshop from Git Repo:

git clone -b <user-branch> <repo-URL>

2. Open the root folder inside VS Code

3. To build the app click the run option in the main method{}

4. Create the text field ‘ new task’

5. Additionally create update and delete option for each data entry.

6. Go To File: <specific-file-with-to-do list-method> à <method-name>, implement your logic.

**You will Achieve**

When you complete this workshop you will learn the following:

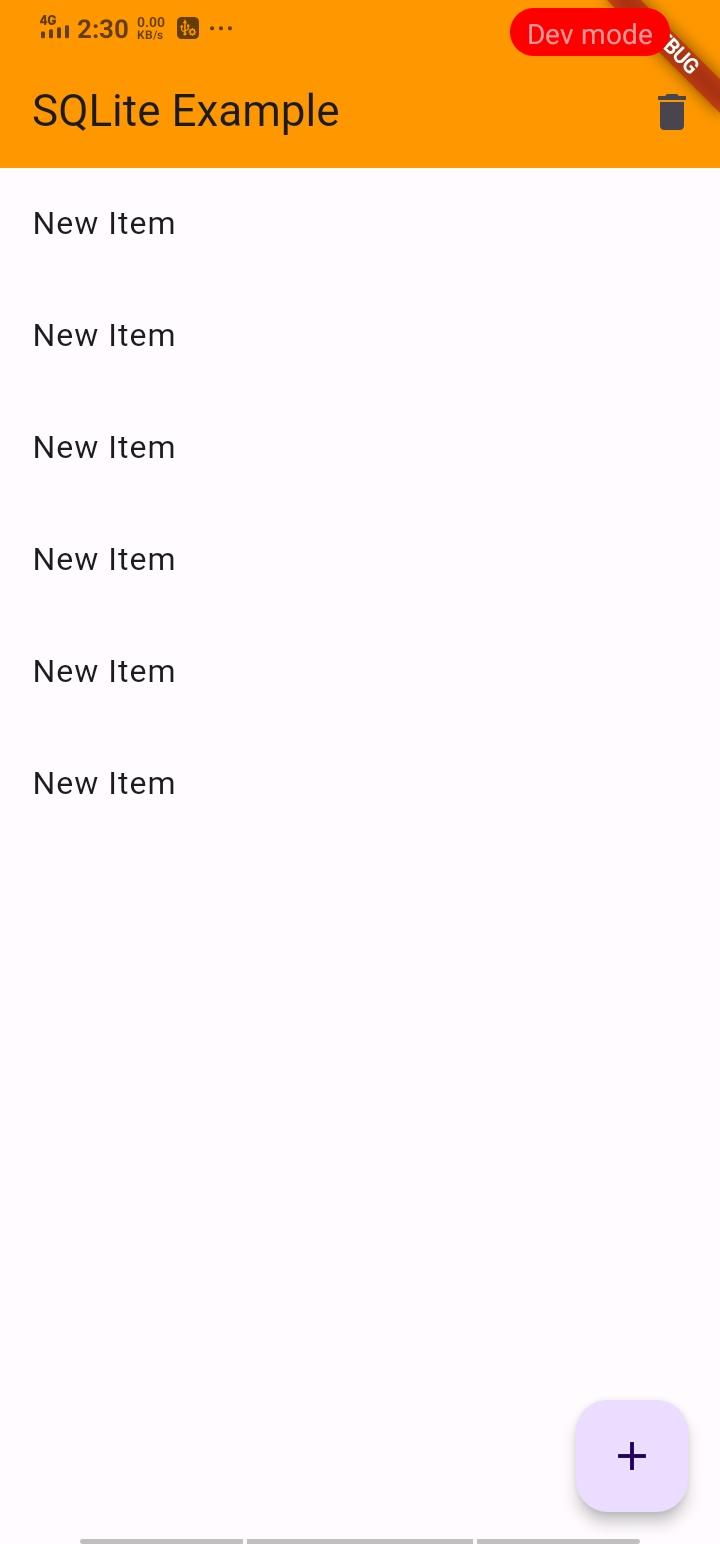
* **SQLite Database Integration:**
* The code integrates the SQLite database functionality into a Flutter application to store and manage data.
* **UI Components:**
* The UI is built using various Flutter widgets:
* **Scaffold:** Provides the basic structure for the app.
* **AppBar:** Displays the app bar with the title and delete action button.
* **ListView.builder:** Renders a scrollable list of items.
* **ListTile:** Represents each item in the list with options for deletion and update**.**
* **FloatingActionButton:** Provides a button to add new items.
* **Dependencies:**
* **The code imports dependencies from two external packages:**
* **sqlite:** For working with SQLite databases in Flutter.
* **path:** For handling file paths.
* **Database Operations:**
* The \_initDatabase() method initializes the SQLite database and creates a table named 'items' with columns 'id' and 'name'.
* The \_fetchItems() method retrieves items from the database and updates the \_items list.
* The \_addItem() method inserts a new item into the 'items' table.
* The \_updateItem(int id) method updates an existing item in the 'items' table.
* The \_deleteItem(int id) method deletes an item from the 'items' table.
* **UI Customization:**
* The app's theme is set to use the primary color swatch of blue.
* The app bar's color is customized to orange.
* Icons for delete and add actions are provided.

# **Screenshots**

## **Before implementation (without Textfield and CRUD operation)**



## **After implementation (With Textfield and CRUD operation)**



# **How to submit your workshop**

Push your project back to the same git branch using command:

<command name>

# **Happy Coding!**