Task manager(using sharedpreferences)

Workshop #1

Flutter Developer Bootcamp

# **Purpose**

This workshop demonstrates how to create a basic task manager app using Flutter, with functionality for adding, viewing, and deleting tasks, along with data persistence to ensure tasks are not lost when the app is closed or restarted.

# **Problem**

In the given workshop, there will be a Task manager app containing an " New Task" text field for noting necessary data. You need to create an delete option for deleting each data you have added.

# **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. Enter the necessary data in the field ‘ new task’

5. Create an delete option for deleting each data you have added.

6. Go To File: <specific-file-with-task manager-method> à <method-name>, implement your logic.

**You will Achieve**

When you complete this workshop you will learn the following:

* **Text Input:** Users can input new tasks via a TextField widget.
* **Task Listing:** The tasks entered by the user are displayed in a list format using a ListView.builder widget.
* **Task Deletion:** Each task displayed in the list has a delete button (IconButton) associated with it. When clicked, it removes the corresponding task from the list.
* **Data Persistence:** Tasks added by the user are stored locally on the device using the shared\_preferences package, ensuring that they persist across app restarts.
* **UI Customization:**

The app's UI is designed using various widgets from the Flutter framework, including Scaffold, AppBar, Column, Padding, Row, Expanded, Text, TextField, IconButton, and ListView.builder.

The app's theme uses the primary color swatch provided by Flutter (Colors.blue).

* **Initialization:**

The initState method is used to load tasks from local storage when the app starts.

* **Dependency:**

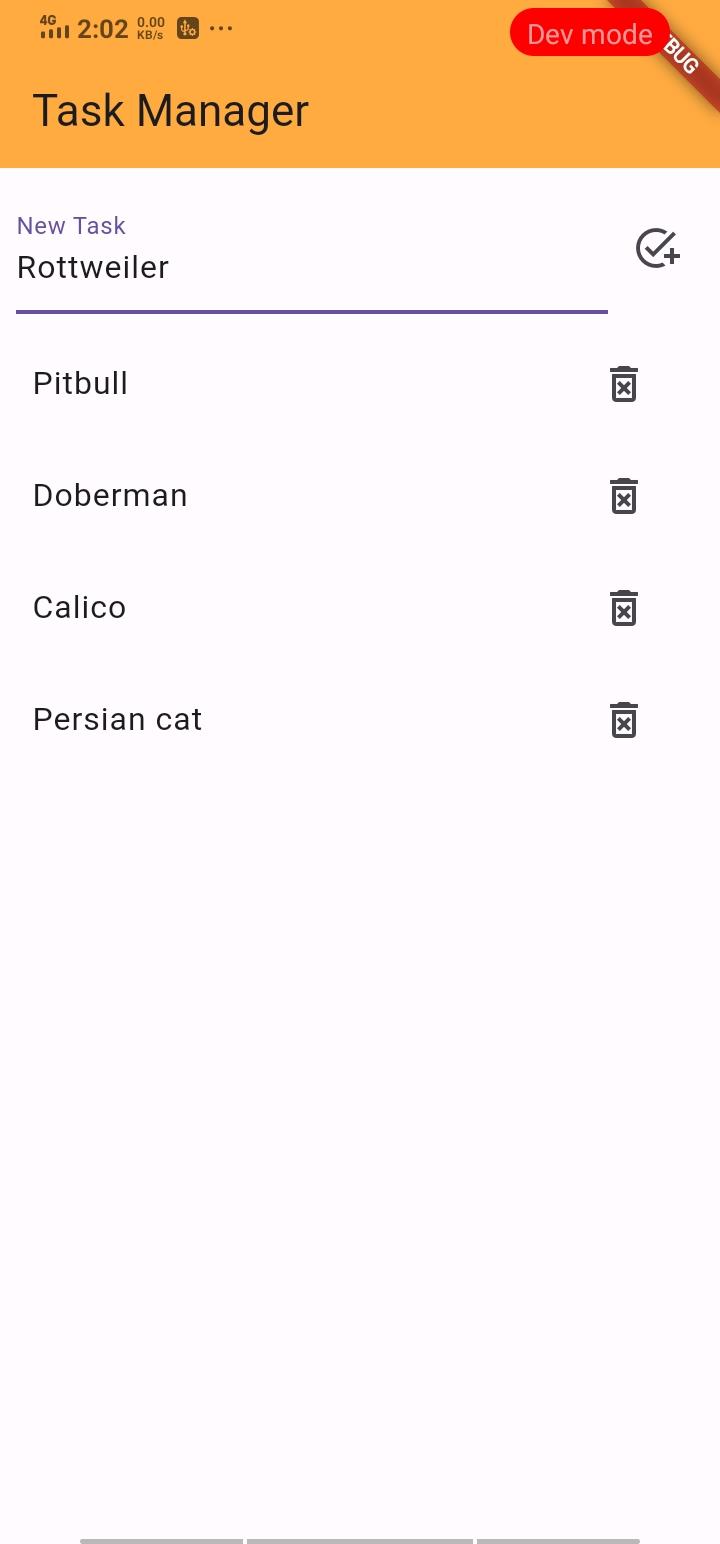
The app relies on the shared\_preferences package for storing and retrieving task data locally.

# **Screenshots**

## **Before implementation (without delete option)**



## **After implementation (With delete option)**



# **How to submit your workshop**

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

<command name>

# **Happy Coding!**