Flutter Developer Bootcamp

url launcher

Workshop #03

# **Purpose**

This workshop demonstrates how to use the url launcher plugin in the contact us page in flutter.

# **Problem**

In the given workshop, do you need to make the two icons responsive? And when the call icon is tapped, should it navigate to dials, and when the location icon is tapped, should it navigate to the current location on the map?

# **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. Execute the code by running command from the root: <full-command>

4. Make the icons responsive.Tapping the call icon navigates to dials, while tapping the location icon goes to the current map location.

6. Go To File: <specific-file-with-url -method> à <method-name>, implement your url logic. Make the URL work

# **You Will Achieve**

When you complete this workshop, you will learn the following:

* Flutter Widgets:

Widgets are like building blocks used to create the user interface (UI) of a Flutter application.

* In the code, various widgets like Scaffold, AppBar, IconButton, TextField, ElevatedButton, etc., are used to create different UI components such as app bars, buttons, text input fields, icons, etc.
* Navigation:

Navigation refers to moving between different screens or pages within an app.

The Navigator.of(context).pop() function is used when the back button is pressed to navigate back to the previous screen.

* Text Input Handling:

Text input handling involves allowing users to input text into specific fields.

In the code, TextField widgets are used to create input fields where users can input their name, email, and message.

* Launching External Apps:

This involves opening other apps installed on the device, like email, phone, or maps, from within the Flutter app.

Functions like \_launchEmail, \_launchPhone, and \_launchMap use the url\_launcher package to open the email, phone, and maps apps respectively, based on user interaction with icons.

* Asynchronous Operations:

Asynchronous operations are tasks that don't block the main thread of the application, allowing the app to remain responsive.

In the code, launching external applications is an asynchronous operation, so functions like \_launchEmail, \_launchPhone, and \_launchMap are marked with async and use await to wait for the operation to complete.

* Error Handling:

Error handling involves dealing with unexpected situations or errors that may occur during the execution of the code.

In the code, try-catch blocks are used to catch any errors that might occur while launching external applications and handle them gracefully.

* Styling:

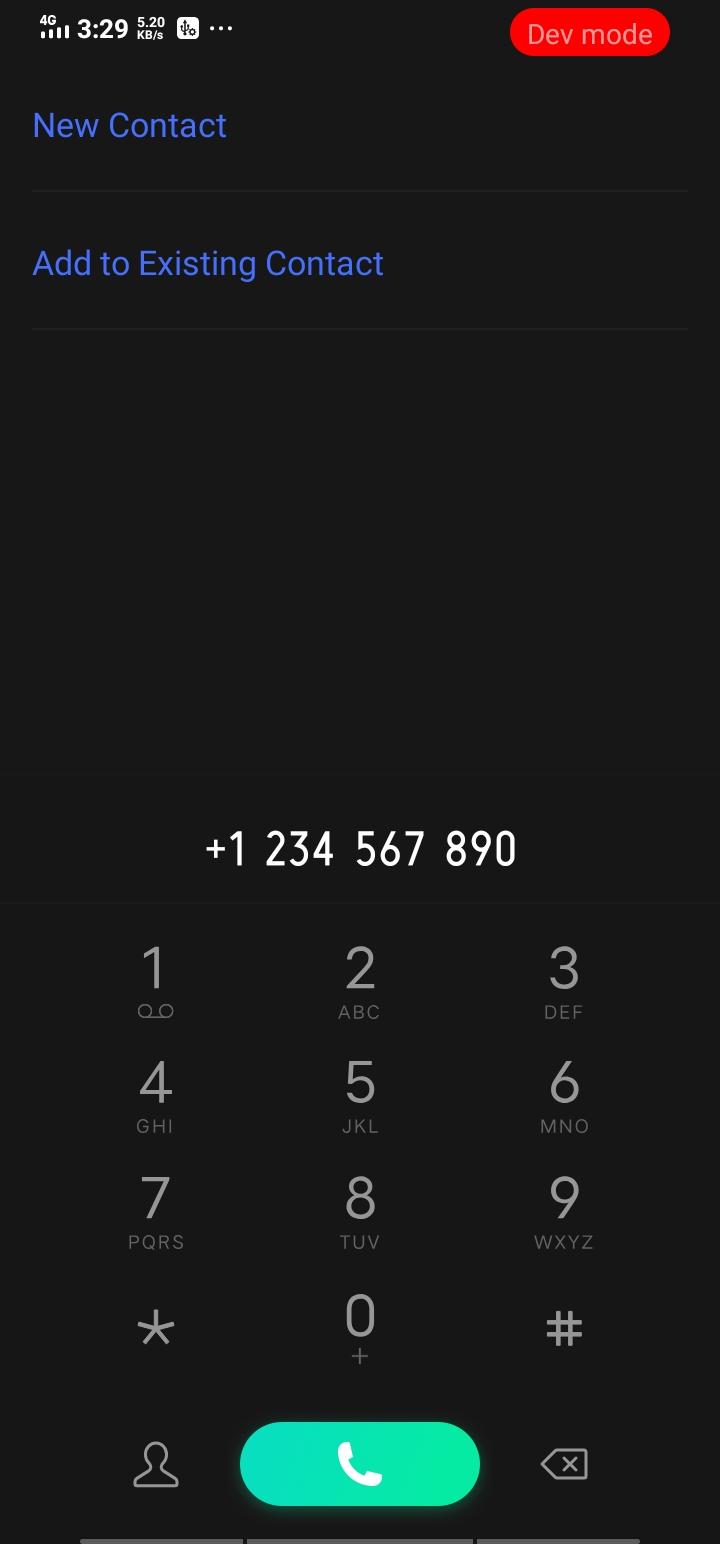
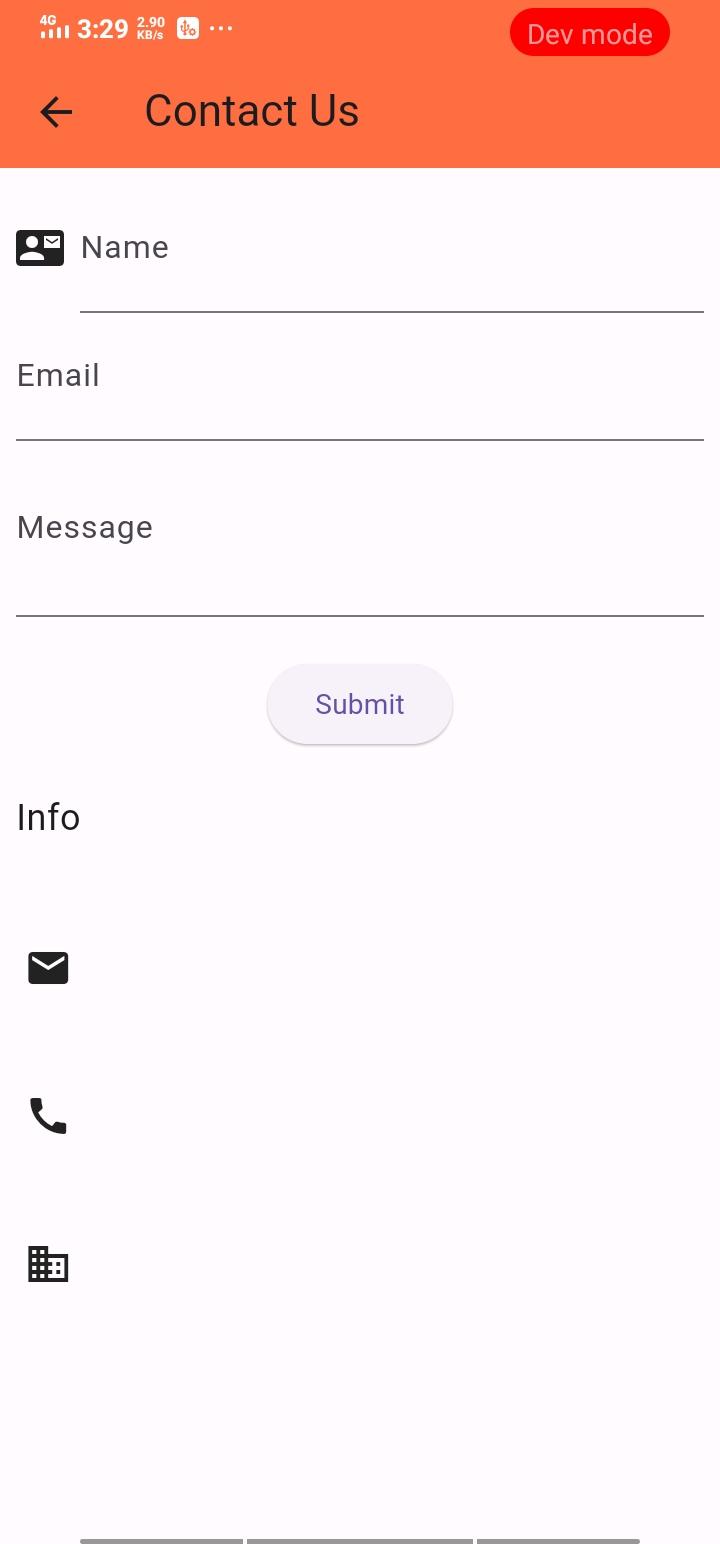
Styling involves customizing the appearance of UI elements to make them visually appealing.

In the code, properties like backgroundColor, fontSize, labelText, etc., are used to style various UI components such as app bars, text input fields, and icons.

# **Screenshots**

## **Before implementation (without icons)**

## **After implementation (With icons)**



# **How to submit your workshop**

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

<command name>

# **Happy Coding!**