Flutter Developer Bootcamp

Lottie Animation

Workshop #02

# **Purpose**

This workshop demonstrates how to create an animation by using the lottie plugin.

# **Problem**

In the given workshop, the issue you have to solve is to add buttons for three more pets (e.g., dog, bird, fish) along with corresponding Lottie animations. Users should be able to tap on any pet button to view its animation.

# **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. Add buttons for three more pets (e.g., dog, bird, fish) along with corresponding Lottie animations.

5. Go To File: <specific-file-with-lottie animation-method> à <method-name>, implement your animation logic. Make the animation work

# **You Will Achieve**

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

* Flutter Packages:

The package:flutter/material.dart package provides the core Flutter framework for building UIs with material design components.

The package:lottie/lottie.dart package allows the app to use Lottie animations.

* Scaffold: Provides a basic structure for a material design app. It includes functionality like an app bar, body, and floating action button.
* AppBar: Represents the app bar at the top of the screen. It typically contains a title and actions.
* Column: Arranges its children widgets vertically, one below the other.
* Row: Arranges its children widgets horizontally, side by side.
* ElevatedButton: A material design button with a filled background color when pressed.
* LottieScreen is a stateful widget. It means it can maintain state that might change during the lifetime of the widget. Stateful widgets are used when the UI needs to update dynamically based on changes in data or user interactions.
* Managing State:

LottieScreenState class manages the state of the currentAnimation variable. The setState() function is called when changeAnimation is invoked, indicating that the state has changed. Flutter then automatically rebuilds the UI with the updated state.

* Asset Management:

Animation assets (kitty.json, puppy.json, etc.) are stored in the assets folder. They're loaded into the app using Lottie.asset, which displays Lottie animations.

* Handling User Input:

buildButton function creates buttons for selecting different animations. When a button is pressed, it calls the \_changeAnimation function, which updates the currentAnimation state.

* UI Layout:

Widgets like Column, Row, and SizedBox are used to structure the UI. Column and Row help arrange widgets vertically and horizontally, respectively. SizedBox provides spacing between UI elements.

* AppBar:

The AppBar widget creates a standard Material Design app bar. It contains a title ("PET SHOP") and is centered within the app.

# 

# 

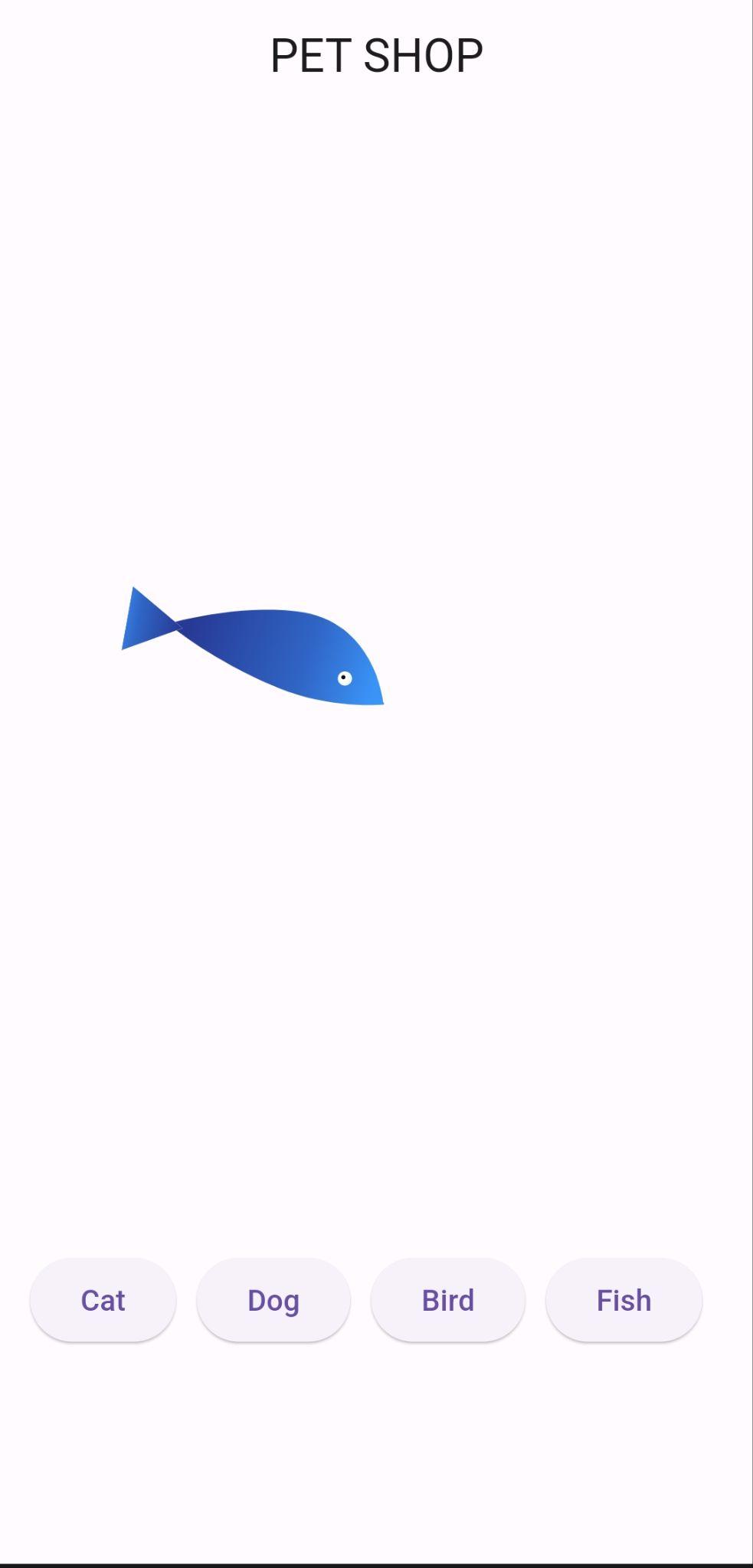
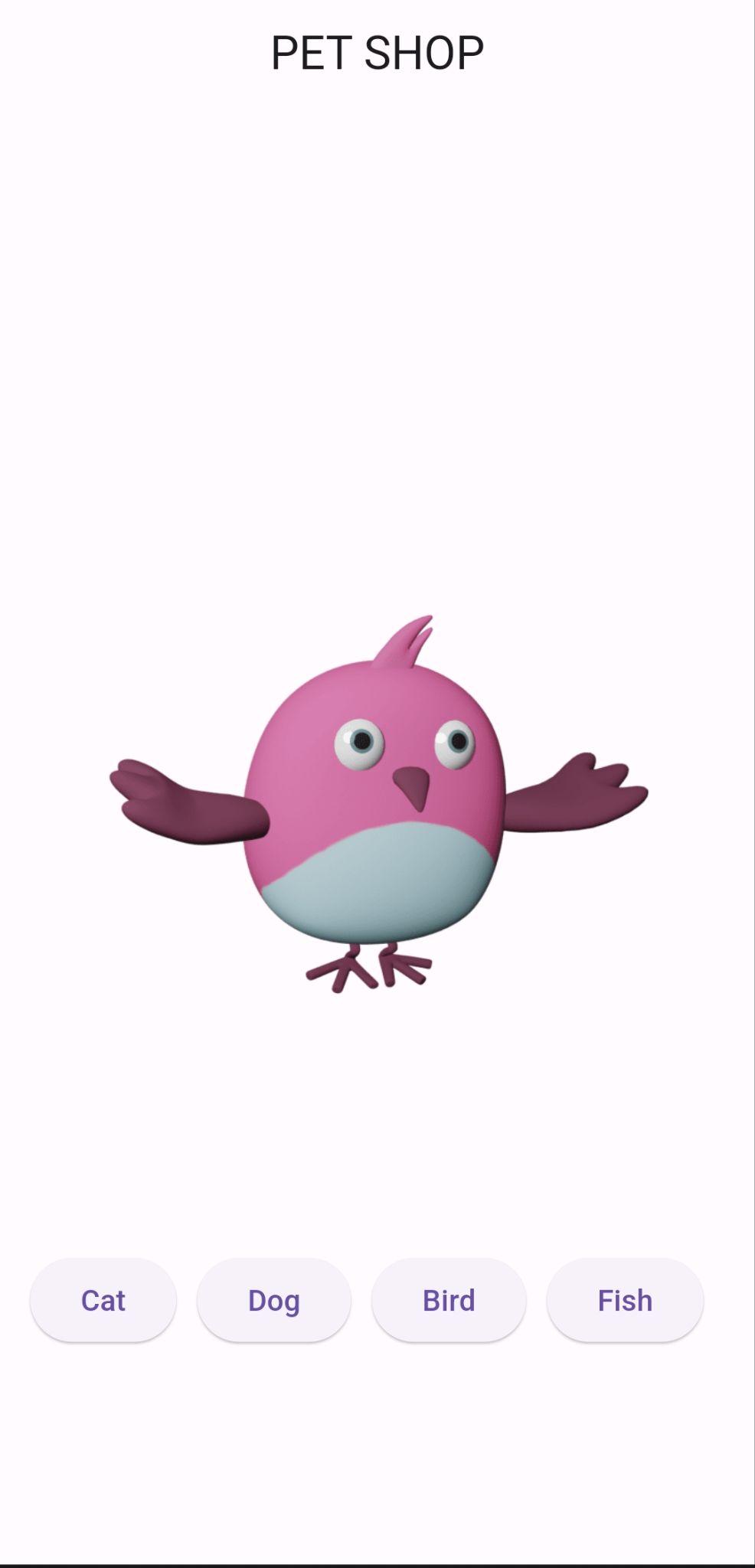
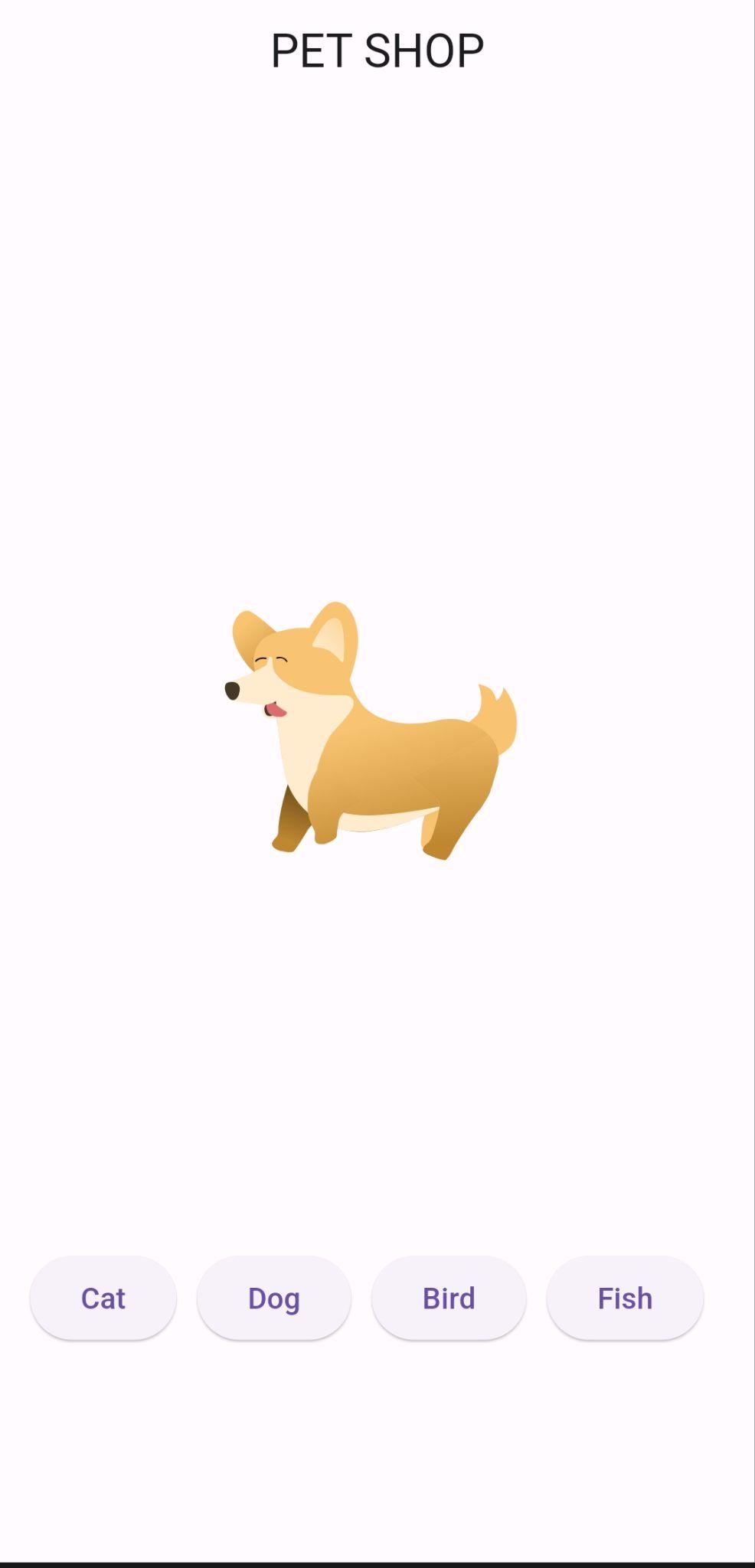
## 

# **Screenshots**

## **Before implementation (one Lottie animation)**



## **After implementation (multiple Lottie animation)**



# **How to submit your workshop**

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

<command name>

# **Happy Coding!**