Flutter Developer Bootcamp

File picker

Workshop #04

# **Purpose**

This workshop demonstrates how to select any type of file like images,files,etc.

# **Problem**

In the workshop, the issue with this code is that while it allows the selection of an image from the file, it fails to display the selected image. Therefore, you should ensure that the image is properly displayed within it.

# **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. Display the selected image or file mention in the problem

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

# **You Will Achieve**

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

* **Imports**:

**import 'dart:io';**: Imports the Dart standard library's dart:io package, which provides classes for working with files, directories, and other input/output operations.

**import 'package:file\_picker/file\_picker.dart';:** Imports the file\_picker package, which allows Flutter apps to pick files from the device's storage.

import 'package:flutter/material.dart';: Imports the Flutter framework's material library, which provides widgets, classes, and utilities for building material design UIs in Flutter apps.

* **Gallery Widget:**

class Gallery extends StatefulWidget { ... }: Defines a Flutter widget named Gallery. It extends StatefulWidget, indicating that its state can change over time.

* **State**:

galleryState: This is the state class for the Gallery widget. It manages the widget's mutable state, such as the selected file information.

* **State Variables:**

exception: A string variable used to display any exceptions that occur during file picking.

* **filename**: A string variable used to display the name of the selected file.
* **filepath**: A string variable used to display the path of the selected file.
* Initially, all these variables are set to indicate that no file has been selected.
* **build() method:**

**Widget build(BuildContext context) { ... }:** Overrides the build method inherited from StatefulWidget. This method builds the UI of the widget.

It returns a Scaffold widget, which provides a basic layout structure for the app.

Inside the scaffold's body, a ListView widget is used to arrange the child widgets vertically.

Text widgets are used to display information about the exception, filename, and filepath.

An ElevatedButton widget triggers the openFile() method when pressed.

An Image widget is included to display the selected file, if any.

* **openFile() method:**

**void openFile() async { ... }:** Defines an asynchronous method openFile() which is called when the button is pressed to open the file picker.

Inside the method, a try-catch block is used to catch any exceptions that occur during file picking.

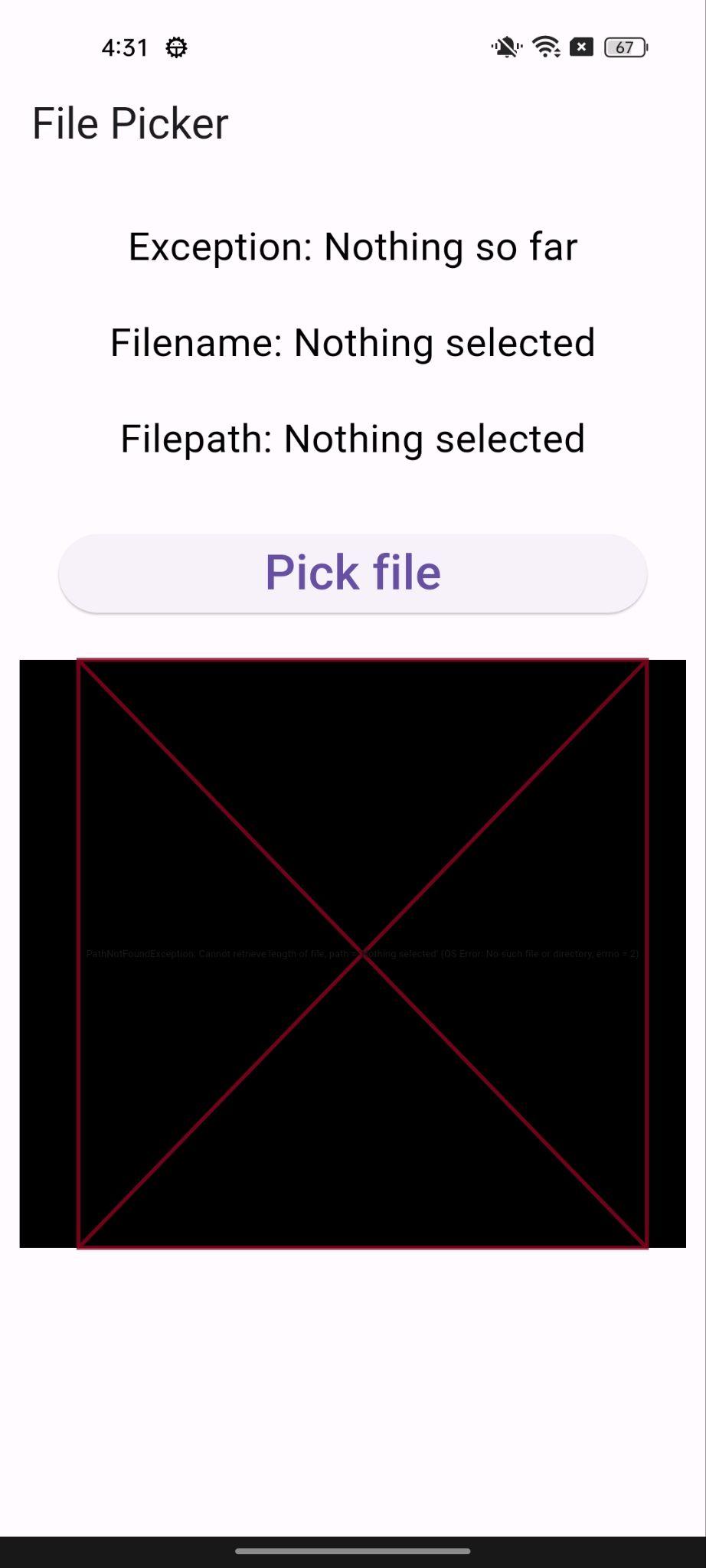
**FilePicker.platform.pickFiles()** is used to open the file picker dialog. It returns a list of picked files.

If a file is selected **(files != null),** the filepath and filename are updated accordingly by extracting information from the selected file's path. setState() is called to update the UI.

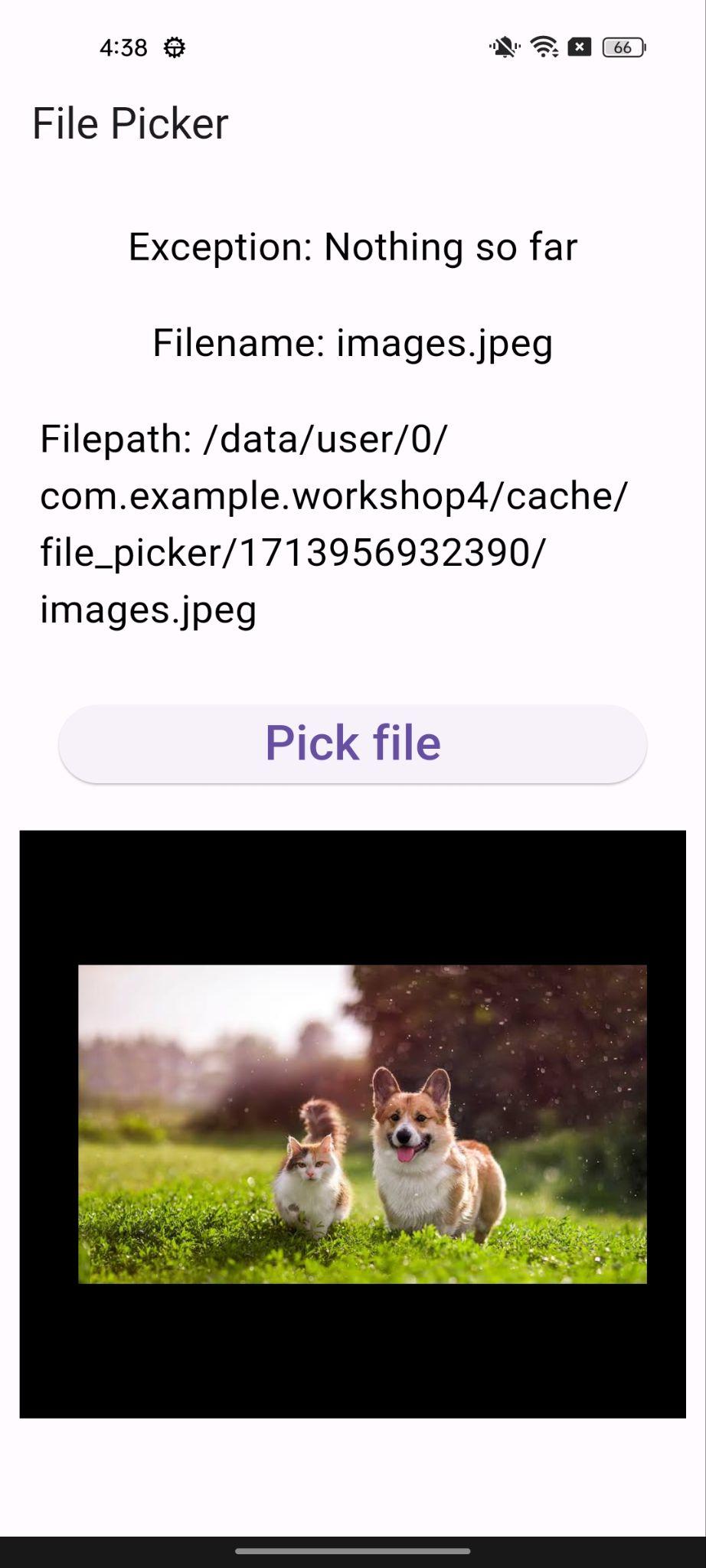
* Any exceptions that occur during this process are caught and printed to the console.

# **Screenshots**

## **Before implementation (without file image)**



## **After implementation(with file image)**



# **How to submit your workshop**

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

<command name>

# **Happy Coding!**