W4-S3 *PRACTICE*

STATEFULL WIDGETS

Learning objectives

- ✓ create **Stateful widgets**
- ✓ be able to **choose the right states** and their type
- ✓ **Render** color or text labels depending on a state
- ✓ Use ex**panded widget**, **elevated buttons**, SizeBox, image, buttons
- ✓ Manage a **mix** of **stateless** and **stateful** widgets

No AI tools allowed to solve this practice

How to submit?

- ✓ **Push** your final code on **your GitHub repository**
- ✓ Then **attach the GitHub path** to the MS Team assignment and **turn it in**

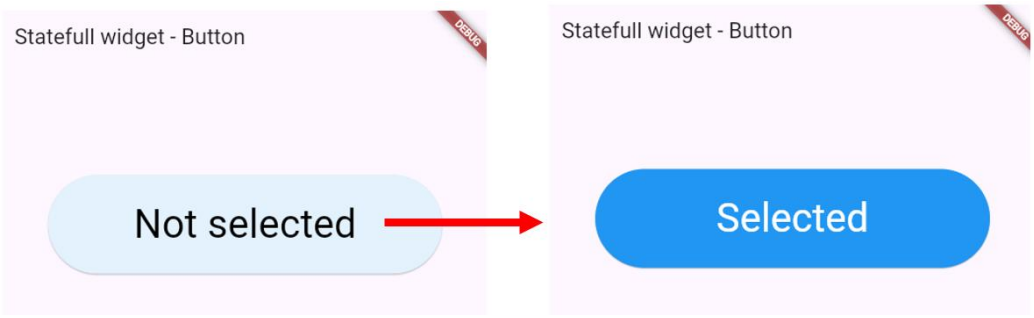
Before practice, to be prepared!

Read the following documentation to be ready for this practice:

https://www.classcentral.com/classroom/youtube-flutter-tutorial-for-beginners-45851/60c82bddaba3e
https://www.classcentral.com/classroom/youtube-flutter-tutorial-for-beginners-45851/60c82bddaba15
https://www.youtube.com/watch?v=GPoRjSjd1cI
https://api.flutter.dev/flutter/material/Card-class.html
https://www.classcentral.com/classroom/youtube-flutter-tutorial-for-beginners-45851/60c82bddaba0a
https://api.flutter.dev/flutter/widgets/Image-class.html

EX 1 – The selectable button

In this exercise, you need to create a Stateful widget to manage the **selection** of a button.



When clicked, the background button color and the text color should change

PART 1 – Understand the **start code**

Look at the start code and understand the different widgets and properties:

```
Center(
  child: SizedBox(
                width: 400,
                height: 100,
                child: ElevatedButton(
                        onPressed: () => {},
                        child: const Center(
                          child: Text("Not Selected"),
                        )),
          ),
)
```

- What is the goal of the **SizeBox** here? More information
- What is an **ElevatedButton**? More information

PART 2 – Create a **stateful widget**

Create a stateful widget **containing the SizeBox and its children**.

- ✓ Manage the selected condition
- ✓ Create getters to get the text, text color and background color depending on the state:

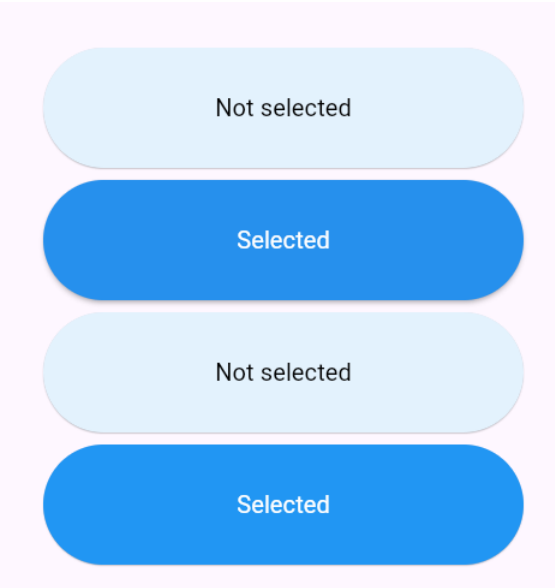
Status	Text label	Text color	Background color
selected	Selected	White	blue[500]
Not selected		Black	blue[50]

- ✓ Change the state value upon button click *(reverse the selection)*

TIP: to change the button background, you might need to use ElevatedButton.styleFrom

PART 3 – Try with different buttons!

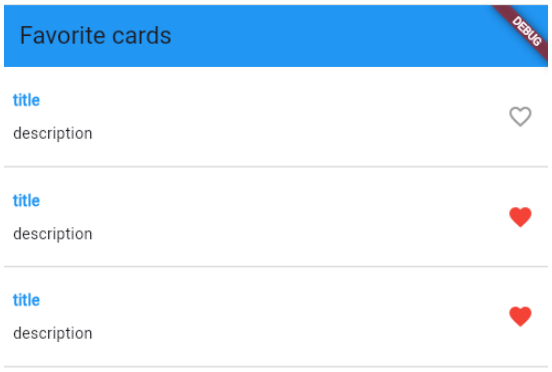
- ✓ Create a column in the main () to manage different buttons
- ✓ Check that each manage manages its own selected state



Each button should manage its own state

EX 2 – The best card

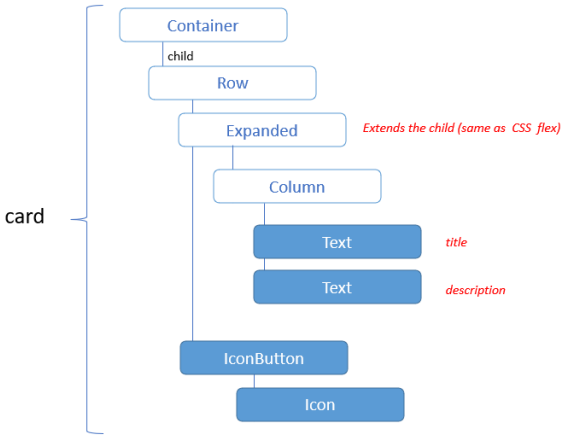
In this exercise, you need to create a Stateful widget to manage a **list of favorite cards**.



PART 1 – Layout the **card**

Important*: For this first step,* ***we provide the correction,*** *but try to do it first by yourself!*

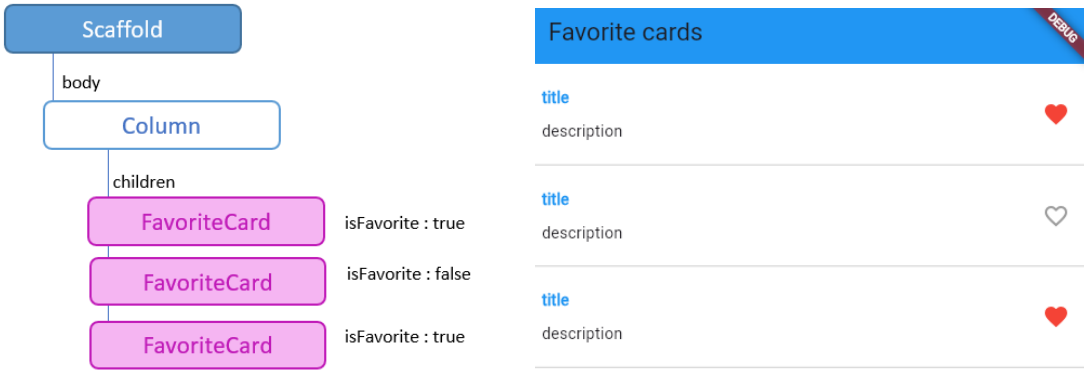
Create the layout for following mockup:



- ✓ You need to use the **expanded widget** to spread the column. More information
- ✓ You need to use the **IconButton** to display the heard in red. More information
- ✓ We also need a **border at the bottom** of the card: use Border and Border Side

PART 2 – Create a **stateless widget**

The second step is to create a stateless widget to manage many cards.
The widget takes are parameter the isFavorite value.



Example of usage of the Favorite card

PART 3 – Refactor to **stateful widget**

Now we want to change the favorite status **when we click on each icon**

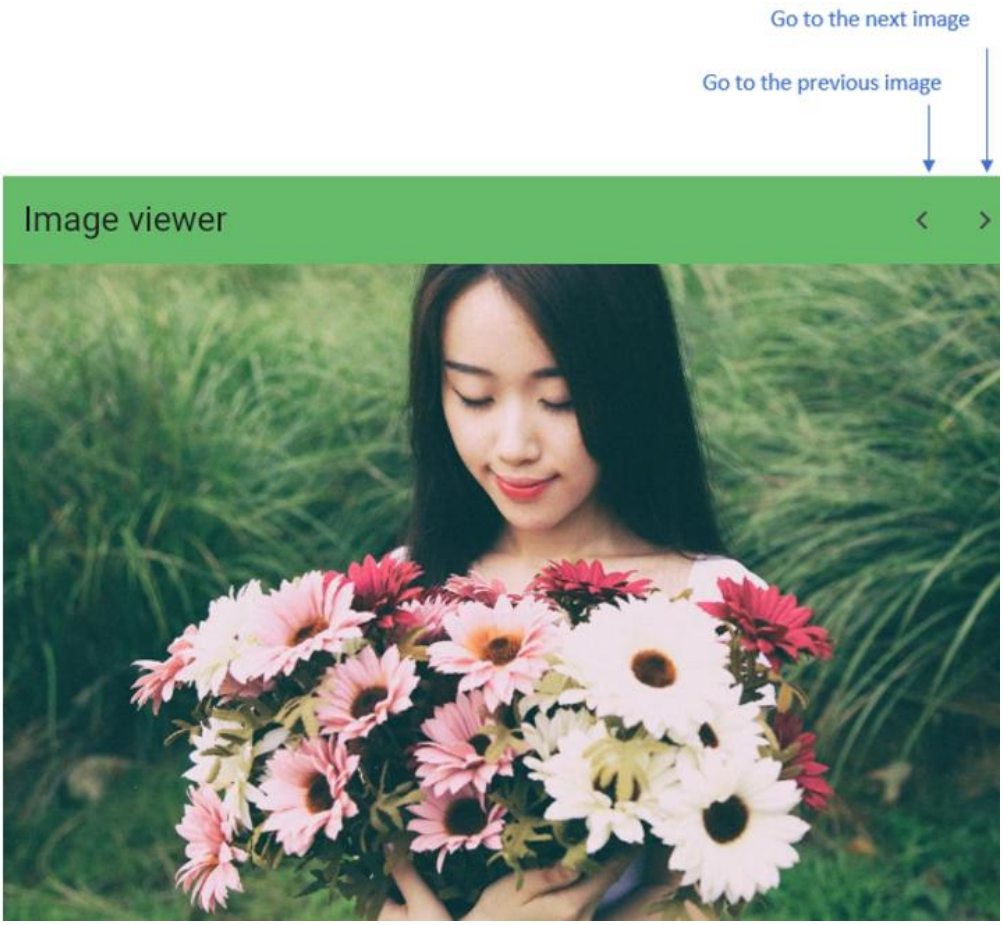
It's time to refactor the stateless widget to a stateful widget to **manage the favorite status**

- When we click on the heart icon**, the favorite status shall change**

Important: *each FavoriteCard manage its own state (is favorite)*

EX 3 – The Image Gallery

In this exercise, we want to create an **image Gallery** as follows:



Note:

- ✓ For this exercise you have the start code with an empty app bar and the first image displayed
- ✓ Also copy the w4-s2/ images folder into your assets and update your pubspec.yaml

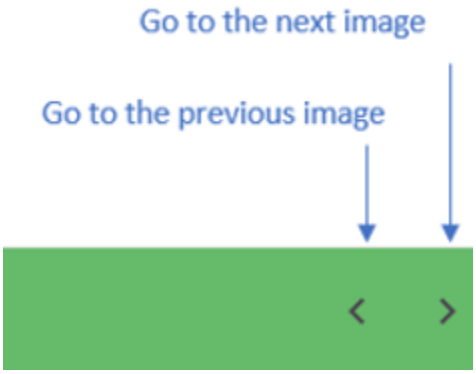
```
# To add assets to your application, add an assets section, like this:
assets:
  - assets/
  - assets/w4-s2/
```

PART 1 – Add the previous and next button on App bar

- Add 2 buttons as bellow.
- Add 2 function to handle the clicks on the 2 buttons

Note: We needed to HIDE the little DEBUG banner, with this line on MaterialApp :

```
debugShowCheckedModeBanner: false,
```

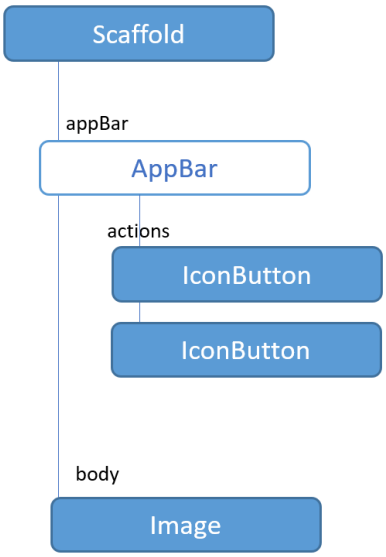


PART 2 – Manage the image state with a stateful widget

To manage the image gallery, with need a **state** and a **stateful widget** to handle which image is currently displayed.

Q1 – What **type of state** do you need to keep?

Q2 – Look at the current widget tree: **which widget** will you warp into your stateful widget?



Important:

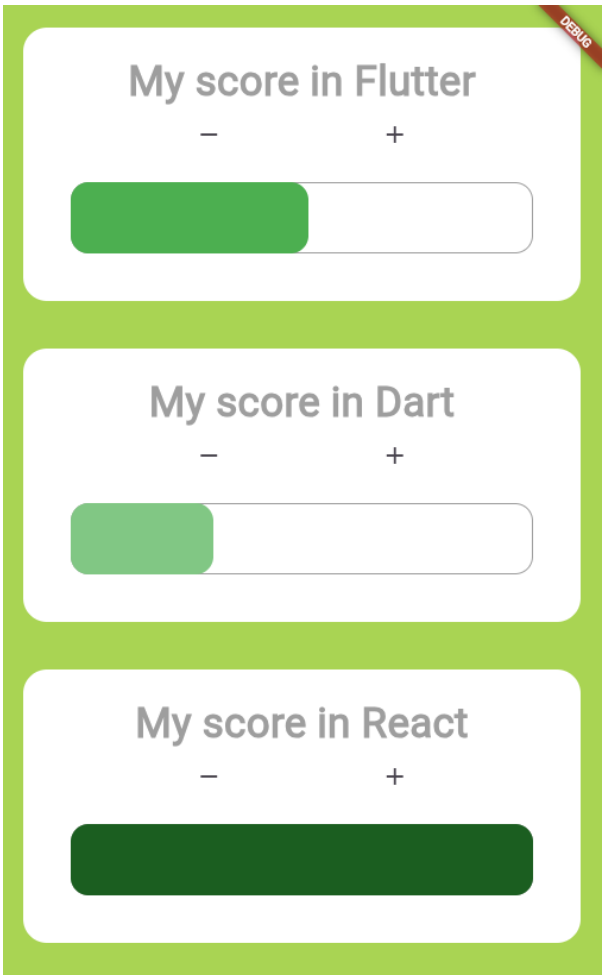
- It should be a circular carrousel: the *previous of the first image is the last image, and the next of the last is the first...* How can you manage this?

EX 4 – Score bar (BONUS)

FREE STYLE EXERCICE

In this exercise, the design, Flutter components and data structure are up to you.

We need to display score cards (1 to 10) as bellow



5 Tips to guide you!

1. Choose appropriate **layout widgets** like Column, Row, Expanded, Container
2. Think about **alignments**, **paddings**, **margins**
3. Think about **inputs** your need (checkbox etc.)
4. Create **reusable components** as much as possible
5. Determine
    - which widgets **need to be stateful** (when UI changes based on user interaction)
    - which can remain **stateless**
    - **which argument y**ou will pass to your stateful and stateless widgets

TIPS: you will need to use a Stack widget to render the progress bar with 1 container (the border and the progress rectangle