# MAD LAB EXPT 2

**Name:Swarali Dhobale_D15A_13**

**Aim: To design Flutter UI by including common widgets.**

**Theory:**

Flutter employs a reactive framework, allowing for fast development and expressive, flexible UI designs. Central to Flutter are widgets, which are the building blocks used to construct user interfaces.

**Container**: A versatile widget used to contain other widgets. It allows you to customize properties such as alignment, padding, margin, color, and more.

**Row and Column:** Widgets used to arrange child widgets horizontally (Row) or vertically (Column). They automatically size and position their children according to their properties.

**Text:** Widget used to display text with styling options like font size, color, weight, and alignment.

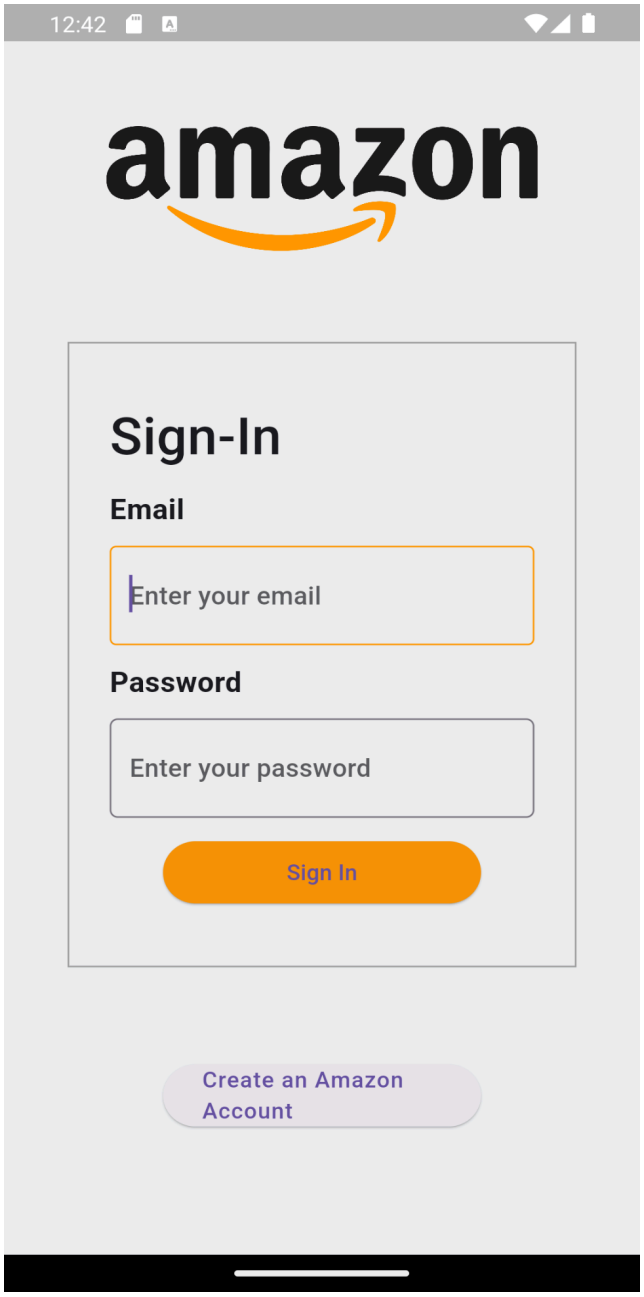
**Image:** Widget used to display images from various sources such as assets, network, or memory. It supports various image formats and provides options for resizing and scaling.

**Icon**: Widget used to display icons from the Material Icons or custom icon sets.

**Output:**

```
import 'package:amazon_clone/utils/utils.dart';
import 'package:flutter/material.dart';

class CustomMainButton extends StatelessWidget {
  final Widget child;
  final Color color;
  final bool isLoading;
  final VoidCallback onPressed;
  const CustomMainButton({
    Key? key,
    required this.child,
```

```
    required this.color,
    required this.isLoading,
    required this.onPressed,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    Size screenSize = Utils().getScreenSize();
    return ElevatedButton(
      style: ElevatedButton.styleFrom(
          primary: color,
          fixedSize: Size(
            screenSize.width * 0.5,
            40,
          )),
      onPressed: onPressed,
      child: !isLoading
          ? child
          : const Padding(
              padding: EdgeInsets.symmetric(vertical: 5),
              child: AspectRatio(
                aspectRatio: 1 / 1,
                child: CircularProgressIndicator(
                  color: Colors.white,
                ),
              ),
            ),
    );
  }
}
```



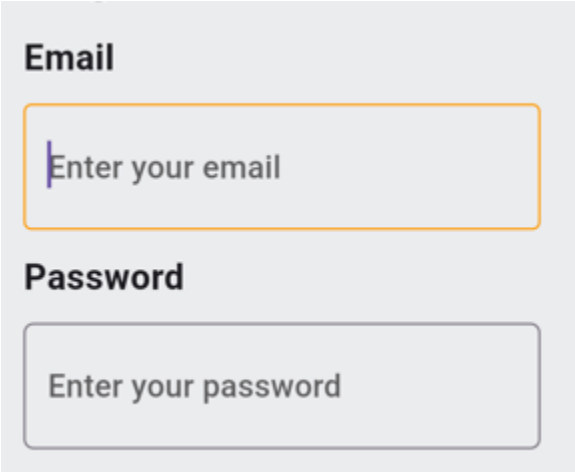

**Text field Widget-**

```
import 'package:flutter/material.dart';

class TextFieldWidget extends StatefulWidget {
```

```
  final String title;
  final TextEditingController controller;
  final bool obscureText;
  final String hintText;
  const TextFieldWidget({
    Key? key,
    required this.title,
    required this.controller,
    required this.obscureText,
    required this.hintText,
  }) : super(key: key);

  @override
  State<TextFieldWidget> createState() => _TextFieldWidgetState();
}

class _TextFieldWidgetState extends State<TextFieldWidget> {
  late FocusNode focusNode;
  bool isInFocus = false;

  @override
  void initState() {
    super.initState();
    focusNode = FocusNode();

    focusNode.addListener(() {
      if (focusNode.hasFocus) {
        setState(() {
          isInFocus = true;
        });
      } else {
        setState(() {
          isInFocus = false;
        });
      }
    });
  }

  @override
  Widget build(BuildContext context) {
    return Column(
      mainAxisSize: MainAxisSize.min,
      crossAxisAlignment: CrossAxisAlignment.start,
      children: [
```

```
Padding(
  padding: const EdgeInsets.only(bottom: 15),
  child: Text(
    widget.title,
    style: const TextStyle(
      fontWeight: FontWeight.bold,
      fontSize: 17,
    ),
  ),
),
Container(
  decoration: BoxDecoration(boxShadow: [
    isInFocus
        ? BoxShadow(
            color: Colors.orange.withOpacity(0.4),
            blurRadius: 8,
            spreadRadius: 2,
          )
        : BoxShadow(
            color: Colors.black.withOpacity(0.2),
            blurRadius: 8,
            spreadRadius: 2,
          )
  ]),
  child: TextField(
    focusNode: focusNode,
    obscureText: widget.obscureText,
    controller: widget.controller,
    maxLines: 1,
    decoration: InputDecoration(
      fillColor: Colors.white,
      filled: true,
      hintText: widget.hintText,
      border: OutlineInputBorder(
        borderRadius: BorderRadius.circular(3),
        borderSide: const BorderSide(
          color: Colors.grey,
          width: 1,
        ),
      ),
      focusedBorder: const OutlineInputBorder(
        borderSide: BorderSide(
          color: Colors.orange,
          width: 1,
```

```
                ),
              ),
            ),
          ),
        )
      ],
    );
  }
}
```




**Conclusion: Common widgets for text and button have been implemented for Flutter application.**