

Param keswani

D15A 27

Batch B

Experiment No 2

AIM: To design Flutter UI by including common widgets.

Theory:

In summary, Flutter widgets are fundamental components in constructing the user interface of a Flutter application. They can be broadly categorized into two types:

'StatelessWidget' representing immutable parts of the UI and 'StatefulWidget' representing mutable components that can change over time.

Some key Flutter widgets include:

1. Scaffold: The basic structure for a Flutter app, providing layout elements such as AppBar, BottomNavigationBar, and a body for main content.
2. Container: A versatile box model used for layout, padding, margin, decoration, and constraints, capable of containing other widgets.
3. Row & Column: Widgets for arranging child widgets horizontally (Row) or vertically (Column), essential for creating flexible and responsive layouts.
4. Text: Used for displaying text on the screen with support for various styling options like font size, color, and alignment.
5. TextField: Captures user input, such as text, numbers, or passwords, with the 'onChanged' property for dynamic updates based on user input.
6. Buttons: Various button widgets like 'ElevatedButton' or 'TextButton' trigger actions when pressed, providing a means for user interaction.
7. Forms: The 'Form' widget manages a group of 'TextField' widgets, facilitating input validation and submission.
8. Icons: The 'Icon' widget displays icons from libraries, enhancing visual elements and conveying meaning through symbols.

Key Design Principles highlighted include:

- Consistency: Common widget usage fosters a consistent design language throughout the app.
- Responsive Layouts: Widgets like 'Row' and 'Column' aid in creating responsive and flexible layouts, adapting to different screen sizes.
- User Input Handling: 'TextField' and 'Form' widgets facilitate proper handling, ensuring data integrity and validation.
- Interactive Elements: Buttons and icons contribute to interactivity and user engagement within the app.
- Visual Styling: The 'Container' widget and styling properties of other widgets allow for visual customization and theming.

Hero Card: A hero card typically refers to a prominent or featured card within a user interface, often used in web design or mobile app design. A hero card is usually larger in size compared to other cards and is placed prominently on a page or screen to draw attention to a specific piece of content, product, or feature. It may contain a title, description, image, and call-to-action button, among other elements.

Card One: "Cardone" could potentially refer to a variety of things depending on the

context. In software development or user interface design, it might refer to the first card in a series of cards, or it could be a specific component or module named "CardOne" within a codebase or design system.

Code :

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

class n_search extends StatefulWidget {
  const n_search({super.key});

  @override
  State<n_search> createState() => _n_searchState();
}

class _n_searchState extends State<n_search> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: TextField(
          decoration: InputDecoration(
            hintText: 'Search',
          ),
        ),
        backgroundColor: Colors.amber,
        automaticallyImplyLeading: false,
        leading: IconButton(
          onPressed: () {
            Navigator.of(context).pop();
          },
          icon: const Icon(Icons.arrow_back_ios),
        ),
        actions: <Widget>[
          Padding(
            padding: EdgeInsets.only(right: 20.0),
            child: GestureDetector(
              onTap: () {},
              child: Icon(
                Icons.search,
                size: 26.0,
              ),
            ),
          ),
        ],
    );
  }
}
```

```
),
body: Column(
  children: [
    Card(
      shadowColor: Colors.black,
      elevation: 10,
      child: Container(
        child: Flex(
          direction: Axis.horizontal,
          children: [
            Container(
              margin: EdgeInsets.fromLTRB(20, 10, 10, 10),
              padding: EdgeInsets.fromLTRB(0, 0, 10, 0),
              decoration: BoxDecoration(
                border: Border(
                  right: BorderSide(
                    //           <--- right side
                    color: Colors.black,
                    width: 3.0,
                  ),
                ),
              )),
            child: Image.network(
              "https://i.pinimg.com/736x/23/ea/0c/23ea0c17068f4e290bada3457c5fef0b.jpg",
              height: 700,
              width: 80),
          ],
        ),
        Container(
          padding: EdgeInsets.fromLTRB(25, 5, 0, 0),
          margin: EdgeInsets.fromLTRB(0, 20, 90, 0),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.start,
            children: [
              Text("Apple",
                style: TextStyle(
                  color: Colors.black,
                  fontSize: 30,
                  fontWeight: FontWeight.bold)),
              Text("Visit profile",
                style: TextStyle(
                  color: Colors.black,
                  fontSize: 15,
                  fontWeight: FontWeight.bold)),
            ],
          ),
        ),
      ],
    ),
  ],
);
```

```
        ),
        Container(
            child: Icon(Icons.email),
            padding: EdgeInsets.fromLTRB(20, 0, 0, 10),
        ),
    ],
),
color: Color.fromRGBO(255, 194, 63, 53),
height: 100,
width: double.infinity,
margin: EdgeInsets.all(10),
),
),
],
));
}
}
```

Screenshot :

9:38 ⚡ 🔋

< Search 🔎

 **Architect Engineer**

Apple
We Need an Architect Engineer >
for our new office design
and its construction .
Architectural engineers apply
practical and