Name: Shantanu Saraf
Class: D15B
Roll No.: 59

Experiment No:02

Aim: To design Flutter UI by including common widgets.

Theory:

Text Widgets:
Text is a fundamental widget for displaying textual content in Flutter. It allows developers to customize various text attributes such as font size, color, font family, and alignment. Moreover, Text supports internationalization and localization by integrating with Flutter's localization features. RichText, on the other hand, provides advanced text styling capabilities by allowing different parts of the text to have different styles within the same widget. This is achieved using TextSpan objects, which enable inline styling like bold, italic, underline, and more.
Input Widgets: TextField and TextFormField are essential for capturing user input. TextField offers a blank field where users can enter text, while TextFormField extends TextField by providing additional features such as built-in validation, input formatting, error messages, and integration with Flutter's Form widget for managing form state. Developers can customize the appearance and behavior of these input widgets to suit the specific requirements of their applications. They can also utilize input controllers to interact with the entered text and perform actions such as validation or manipulation.

Button Widgets:
Buttons are crucial for user interaction in Flutter apps. ElevatedButton is typically used for primary actions that require more prominence, such as submitting a form or confirming a critical action. TextButton provides a minimalist button design with text only, suitable for secondary actions or less critical operations. IconButton allows developers to add buttons with icons, facilitating intuitive navigation and interaction. FloatingActionButton is a specialized button widget that floats above the content and is commonly used for primary and high-priority actions like adding a new item or composing a message.

Selection Widgets:
Flutter offers several widgets for user selection tasks. Checkbox provides a binary selection mechanism, allowing users to toggle a state between checked and unchecked. Radio buttons are used when users need to select a single option from a list of mutually exclusive choices. Switch widget presents a toggle switch for turning an option on or off, such as enabling or disabling a feature. Slider widget allows users to select a value from a continuous range by dragging a thumb along a track, making it suitable for tasks like adjusting volume or setting preferences.
Layout Widgets: Layout widgets play a vital role in organizing the UI components of a Flutter app. Row and Column widgets are used to arrange child widgets horizontally and vertically,

respectively. Developers can use flex properties and MainAxisAlignment/CrossAxisAlignment to control the alignment and spacing of widgets within Rows and Columns. Stack widget overlays multiple widgets on top of each other, enabling complex UI compositions like overlapping images or implementing floating action buttons. Container widget is a versatile layout widget that allows developers to customize the positioning, size, padding, margin, and decoration of its child widget, providing fine-grained control over the layout and appearance of UI elements.

Scrolling Widgets:

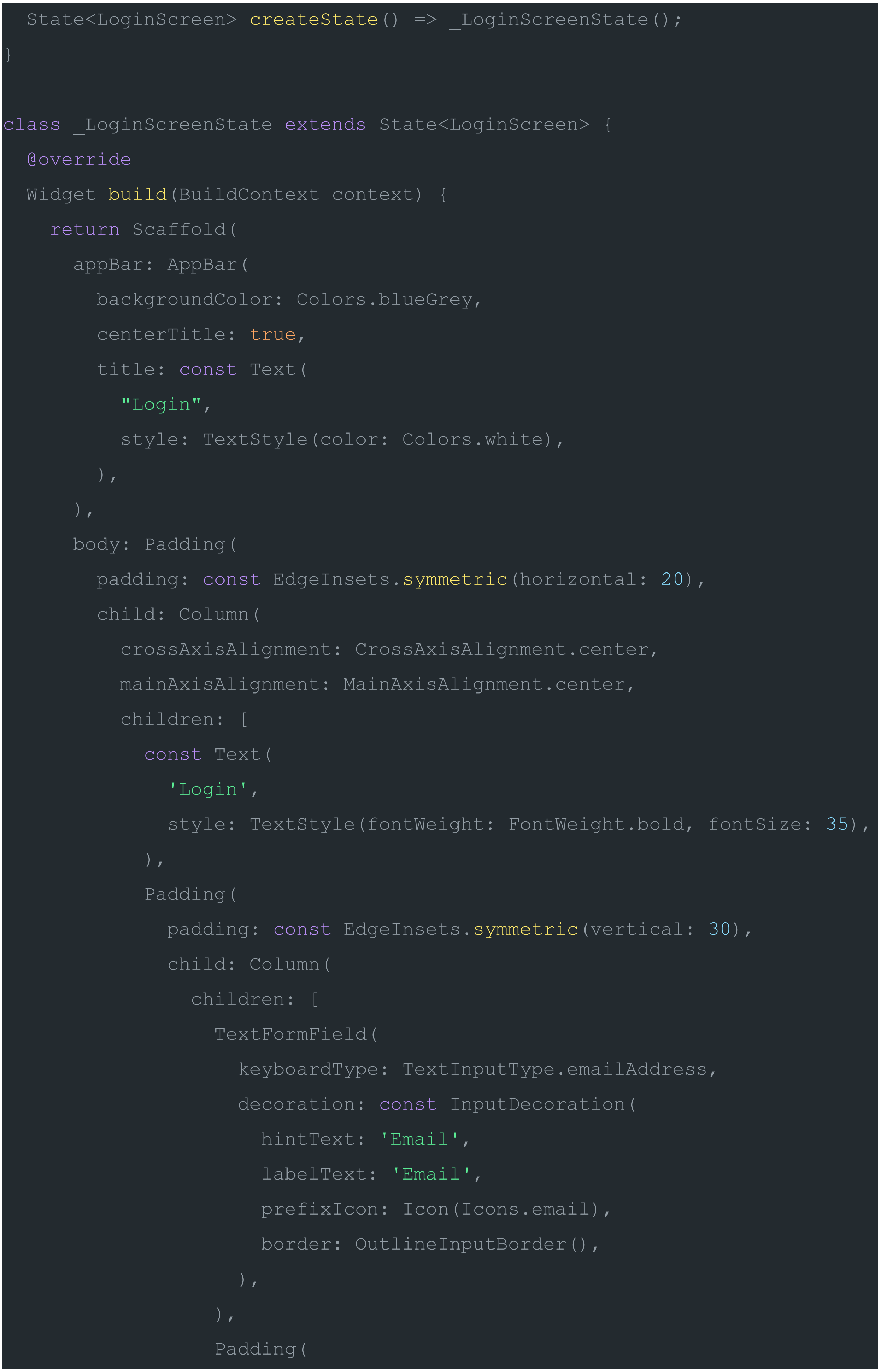
Scrolling widgets are essential for displaying content that exceeds the available screen space. ListView widget provides a scrollable list of items, allowing users to scroll vertically or horizontally through the content. GridView organizes items in a grid layout, making it suitable for displaying collections of data in a structured manner. SingleChildScrollView widget enables scrolling in a single direction, useful for scenarios where the content needs to be scrolled vertically or horizontally but not both simultaneously. Developers can customize the behavior and appearance of scrolling widgets by configuring properties such as scroll direction, physics, and scroll controllers.

Material Design Widgets:

Flutter's Material Design widgets adhere to the design principles outlined in Google's Material Design guidelines. AppBar widget represents the app bar at the top of the screen, providing space for titles, icons, and navigation actions. Scaffold widget implements the basic layout structure of a Material Design app, including app bars, drawers, bottom navigation bars, and floating action buttons. Card widget displays content within a material design card, featuring rounded corners, elevation, and shadow effects. These widgets help developers create visually consistent and aesthetically pleasing user interfaces that align with the Material Design language. Interaction Widgets: Interactivity is a key aspect of modern app development, and Flutter provides widgets to facilitate various user interactions. GestureDetector widget detects gestures such as taps, drags, and pinches on its child widget, allowing developers to respond to these gestures and implement custom touch-based interactions. InkWell widget provides visual feedback to touch interactions by displaying a ripple effect when tapped, enhancing the perceived responsiveness and usability of the UI. Developers can leverage these interaction widgets to create engaging and intuitive user experiences that encourage user engagement and interaction with their Flutter apps.

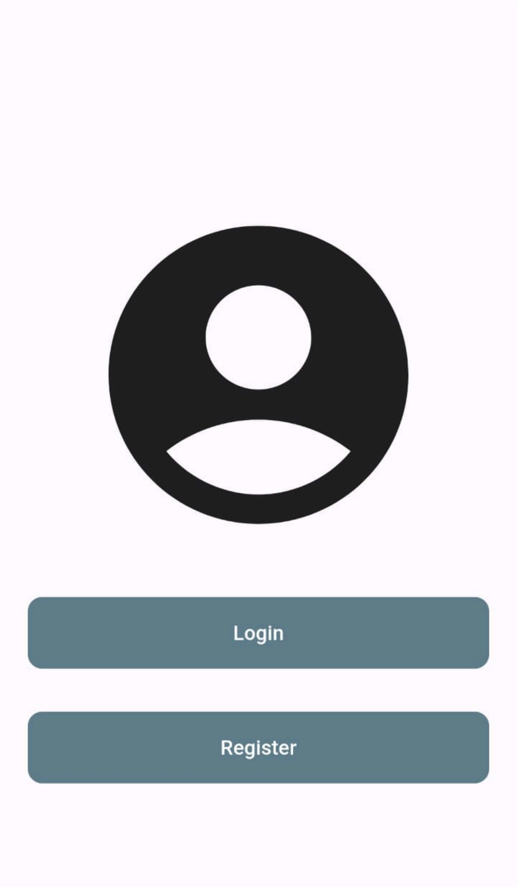
Code:

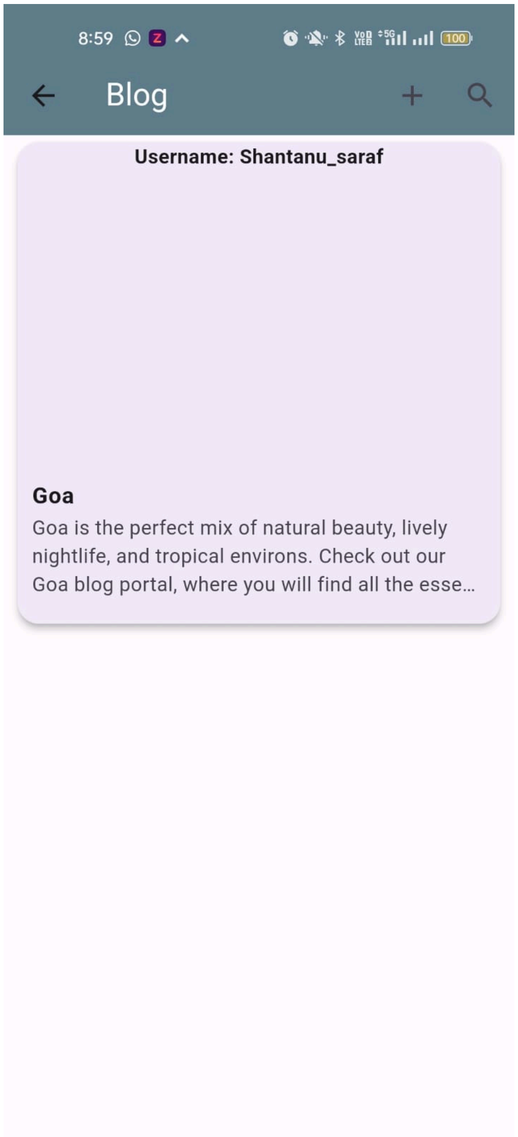
```dart
// login_screen.dart
import 'package:flutter/material.dart';
import 'package:bloggerapp/sceens/blog_page.dart'; // Import the BlogPage

class LoginScreen extends StatefulWidget {
  const LoginScreen({Key? key});

  @override
```



```
  State<LoginScreen> createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.blueGrey,
        centerTitle: true,
        title: const Text(
          "Login",
          style: TextStyle(color: Colors.white),
        ),
      ),
      body: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 20),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.center,
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            const Text(
              'Login',
              style: TextStyle(fontWeight: FontWeight.bold, fontSize: 35),
            ),
            Padding(
              padding: const EdgeInsets.symmetric(vertical: 30),
              child: Column(
                children: [
                  TextFormField(
                    keyboardType: TextInputType.emailAddress,
                    decoration: const InputDecoration(
                      hintText: 'Email',
                      labelText: 'Email',
                      prefixIcon: Icon(Icons.email),
                      border: OutlineInputBorder(),
                    ),
                  ),
                  Padding(
                    padding: const EdgeInsets.symmetric(vertical: 15),
                    child: TextFormField(
                      keyboardType: TextInputType.visiblePassword,
                      decoration: const InputDecoration(
                        hintText: 'Password',
                        labelText: 'Password',
                        prefixIcon: Icon(Icons.lock),
                        border: OutlineInputBorder(),
                      ),
                    ),
                  ),
                  ElevatedButton(
                    onPressed: () {
                      Navigator.push(
                        context,
                        MaterialPageRoute(builder: (context) =>
BlogPage()),
                      );
                    },
                    child: Text('Login'),
                  ),
                ],
              ),
            ),
          ],
        ),
      ),
    );
  }
}

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

class BlogPage extends StatelessWidget {
  const BlogPage({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Blog', style: TextStyle(color: Colors.white)),
        backgroundColor: Colors.blueGrey,
        actions: [
          IconButton(
            icon: Icon(Icons.add),
            onPressed: () {
              // Add functionality for plus button
            },
          ),
          IconButton(
            icon: Icon(Icons.search),
            onPressed: () {
              // Add functionality for search button
            },
          ),
        ],
      ),
      body: ListView.builder(
        itemCount: 1, // Assuming only one blog post for demonstration
        itemBuilder: (context, index) {
          return Card(
            elevation: 4,
            margin: EdgeInsets.symmetric(horizontal: 10, vertical: 5),
            shape: RoundedRectangleBorder(
              borderRadius: BorderRadius.circular(15),
            ),
            child: Column(
              children: [
                Container(
                  width: double.infinity,
                  height: 200, // Adjust the height as needed
                  decoration: BoxDecoration(
                    image: DecorationImage(
                      image: AssetImage('images/goa.jpgc'), // Add image
                      fit: BoxFit.cover,
                    ),
                  ),
                ),
                ListTile(
                  contentPadding: EdgeInsets.all(10),
                  title: Text(
                    'Goa',
                    style: TextStyle(fontWeight: FontWeight.bold),
                  ),
                  subtitle: Text(
                    'Goa is the perfect mix of natural beauty, lively
nightlife, and tropical environs. Check out our Goa blog portal, where you
will find all the essential information to help you plan an awesome sauce
vacation to the party-state. From the best tourist attractions to plush
resorts, this collection of travel blogs on Goa will make your trip fun and
hassle-free. Discover the tropical state\'s best-kept secrets in these
blogs. Each Goa travel blog showcase essential information such as the best
time to visit, how to reach, and other nitty-gritty tips and information
for you to enjoy your vacation fully. We have treated every Goa tour blog
as a platform to showcase our love and fervor for India’s favorite party
destination. Don\'t limit yourself to crowded beaches and attractions when
you can explore new and offbeat places with the help of these Goa trip
blogs. Whether it\'s North Goa or South Goa, these Goa travel blogs have
got you covered for both parts of the Portuguese corner of India. Pin your
favorite articles and feel free to share on your social media channels to
kick start everyone\'s wanderlust to Goa. Watch out our this space for the
latest updates and new exciting information every day.',
                    maxLines: 3, // Limiting description to 3 lines
                    overflow: TextOverflow.ellipsis,
                  ),
                  onTap: () {
                    // Add functionality for tapping on blog post
                  },
                ),
              ],
            ),
          );
        },
      ),
    );
  }
}
```

Output:





Username: Shantanu_saraf

Goa

Goa is the perfect mix of natural beauty, lively nightlife, and tropical environs. Check out our Goa blog portal, where you will find all the esse...

Conclusion:

I have successfully studied and used different Common Widgets used in Flutter UI such as Column Widget, Scaffold,Text,SizedBox etc.