

Project Title: BigBasket  
27

RollNo.

MAD & PWA Lab Journal

Experiment No.	02
Experiment Title.	To design Flutter UI by including common widgets.
Roll No.	27
Name	Rujuta Medhi
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO1: Understand cross platform mobile application development using Flutter framework
Grade:	

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

## **THEORY:**

Designing a Flutter UI using **common widgets** involves leveraging the foundational widgets provided by Flutter to build interactive, responsive, and visually appealing user interfaces. These common widgets allow developers to construct the basic building blocks of the app interface, such as text, buttons, images, forms, lists, and more. Here's a breakdown of how Flutter UI design incorporates common widgets:

### **1. Container**

- **Role:** A container is a fundamental widget in Flutter, used to create a rectangular box with various properties like padding, margin, decoration, color, and alignment. It can be used to wrap other widgets and customize their appearance.
- **Usage:** Containers are useful for aligning, decorating, and sizing widgets. For example, it can be used to create a card-like structure, set background colors, or add rounded corners.

### **2. Row and Column**

- **Role:** These are layout widgets used for arranging children widgets either horizontally (Row) or vertically (Column).
- **Usage:** These are commonly used to create flexible layouts. Rows and columns are used for simple linear layouts, but they can also be combined with other layout widgets to create more complex structures (e.g., GridView, Flex).

### **3. Text**

- **Role:** Displays a string of text.
- **Usage:** The Text widget is used to display static text on the screen. It can be customized with various styles (font size, color, weight, etc.).

### **4. TextField and Form**

- **Role:** TextField allows users to enter text, while Form and TextFormField are used for form validation and handling text input.
- **Usage:** Common for building login forms, registration forms, or any interactive form where user input is required.

### **5. Buttons (ElevatedButton, TextButton, IconButton)**

- **Role:** Buttons are used to perform an action when pressed.
- **Usage:** These are the primary way to handle user interaction in Flutter. You can customize their color, shape, padding, and interaction effects.

### **6. Font**

- **Role:** A widget used to display and customize the text within the application.
- **Usage:** The font is applied to text widgets to define their appearance, including font family, size, weight, and style. It's essential for customizing the typographic design of your app, whether you're using default fonts, custom fonts, or fonts from Google Fonts. You can apply different fonts to various text elements to enhance readability and visual appeal.

## PROGRAM:

Login and Register

```
import 'package:flutter/material.dart';
import 'package:bigbasket_website/home.dart'; //
void main() {
  runApp(const MyApp());
}
```

```
class MyApp extends StatelessWidget {
  const MyApp({super.key});
```

```
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Login Screen',
      theme: ThemeData(
        primarySwatch: Colors.red,
      ),
      home: const LoginScreen(),
    );
  }
}
```

```
class LoginScreen extends StatelessWidget {
  const LoginScreen({super.key});
```

```
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SingleChildScrollView(
        child: Column(
          children: [
            // Top image section
            Container(
              height: 250,
              decoration: const BoxDecoration(
                image: DecorationImage(
                  image: NetworkImage(
                    'assets/loginnn.png'),
                  fit: BoxFit.cover,
                ),
              ),
            ),
            child: const Padding(
```

```

padding: EdgeInsets.all(24.0),
child: Align(
  alignment: Alignment.bottomLeft,

),
),
),
// Login form section
Padding(
  padding: const EdgeInsets.all(24.0),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.stretch,
    children: [
      const Text(
        'Login',
        style: TextStyle(
          fontSize: 24,
          fontWeight: FontWeight.bold,
        ),
      ),
      const SizedBox(height: 24),
      // Social login buttons
      Row(
        children: [
          Expanded(
            child: ElevatedButton.icon(
              onPressed: () {},
              icon: const Icon(Icons.facebook, color: Colors.white),
              label: const Text('FACEBOOK',
                style: TextStyle(color: Colors.white)),
              style: ElevatedButton.styleFrom(
                backgroundColor: const Color(0xFF3B5998),
                padding: const EdgeInsets.symmetric(vertical: 12),
              ),
            ),
          ),
          const SizedBox(width: 16),
          Expanded(
            child: OutlinedButton.icon(
              onPressed: () {},
              icon: const Icon(Icons.g_mobiledata,
                color: Colors.black87),
              label: const Text('GOOGLE',
                style: TextStyle(color: Colors.black87)),
              style: OutlinedButton.styleFrom(
                padding: const EdgeInsets.symmetric(vertical: 12),

```

```
),  
    ),  
    ),  
  ],  
),  
const SizedBox(height: 24),  
const Text(  
  'or',  
  textAlign: TextAlign.center,  
  style: TextStyle(color: Colors.grey),  
),  
const SizedBox(height: 24),  
// Email field  
TextField(  
  decoration: InputDecoration(  
    hintText: 'Email',  
    border: OutlineInputBorder(  
      borderRadius: BorderRadius.circular(4),  
    ),  
  ),  
),  
),  
const SizedBox(height: 16),  
// Password field  
TextField(  
  obscureText: true,  
  decoration: InputDecoration(  
    hintText: 'Password',  
    border: OutlineInputBorder(  
      borderRadius: BorderRadius.circular(4),  
    ),  
  ),  
),  
),  
const SizedBox(height: 16),  
// Remember me checkbox  
Row(  
  children: [  
    Checkbox(  
      value: false,  
      onChanged: (bool? value) {},  
    ),  
    const Text('Remember me'),  
    const Spacer(),  
    TextButton(  
      onPressed: () {},  
      child: const Text('TERMS & CONDITIONS',  
        style: TextStyle(color: Colors.green)),
```

```

    ),
  ],
),
const SizedBox(height: 24),
// Login button
ElevatedButton(
  onPressed: () {
    Navigator.pushReplacement(
      context,
      MaterialPageRoute(builder: (context) => const HomePage()), // Navigate to HomePage
    );
  },
  style: ElevatedButton.styleFrom(
    backgroundColor: Colors.green,
    padding: const EdgeInsets.symmetric(vertical: 16),
  ),
  child: const Text('LOGIN', style: TextStyle(color: Colors.white)),
),

const SizedBox(height: 24),
// Bottom text
const Text(
  'New to the platform?',
  textAlign: TextAlign.center,
),
TextButton(
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(builder: (context) => const RegisterScreen()),
    );
  },
  child: const Text('REGISTER HERE', style: TextStyle(color: Colors.green)),
),

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

```

```

class RegisterScreen extends StatefulWidget {

```



```

        prefixIcon: Icon(Icons.person),
      ),
      validator: (value) {
        if (value == null || value.isEmpty) {
          return 'Please enter your name';
        }
        return null;
      },
    ),
    const SizedBox(height: 15),

    // Email Field
    TextFormField(
      controller: _emailController,
      keyboardType: TextInputType.emailAddress,
      decoration: const InputDecoration(
        labelText: 'Email',
        border: OutlineInputBorder(),
        prefixIcon: Icon(Icons.email),
      ),
      validator: (value) {
        if (value == null || value.isEmpty) {
          return 'Please enter your email';
        }
        if (!RegExp(r'^^[^@]+@^[^@]+\.[^@]+').hasMatch(value)) {
          return 'Enter a valid email';
        }
        return null;
      },
    ),
    const SizedBox(height: 15),

    // Phone Number Field
    TextFormField(
      controller: _phoneController,
      keyboardType: TextInputType.phone,
      decoration: const InputDecoration(
        labelText: 'Phone Number',
        border: OutlineInputBorder(),
        prefixIcon: Icon(Icons.phone),
      ),
      validator: (value) {
        if (value == null || value.isEmpty) {
          return 'Please enter your phone number';
        }
        if (value.length != 10) {

```



```

        return 'Phone number must be 10 digits';
    }
    return null;
  },
),
const SizedBox(height: 15),

// Password Field
TextFormField(
  controller: _passwordController,
  obscureText: true,
  decoration: const InputDecoration(
    labelText: 'Password',
    border: OutlineInputBorder(),
    prefixIcon: Icon(Icons.lock),
  ),
  validator: (value) {
    if (value == null || value.isEmpty) {
      return 'Please enter a password';
    }
    if (value.length < 6) {
      return 'Password must be at least 6 characters long';
    }
    return null;
  },
),
const SizedBox(height: 15),

// Address Field
TextFormField(
  controller: _addressController,
  decoration: const InputDecoration(
    labelText: 'Address',
    border: OutlineInputBorder(),
    prefixIcon: Icon(Icons.location_on),
  ),
  validator: (value) {
    if (value == null || value.isEmpty) {
      return 'Please enter your address';
    }
    return null;
  },
),
const SizedBox(height: 25),

// Register Button

```

```

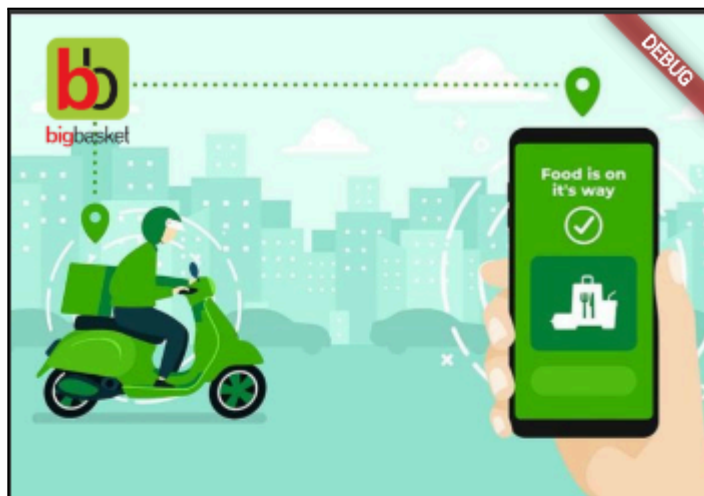
    SizedBox(
      width: double.infinity, // Makes the button take the full width of the parent
      child: ElevatedButton(
        onPressed: () {
          if (_formKey.currentState!.validate()) {
            ScaffoldMessenger.of(context).showSnackBar(
              const SnackBar(content: Text('Registration Successful!')),
            );
            Navigator.pop(context);
          }
        },
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.green,
          padding: const EdgeInsets.symmetric(vertical: 16),
        ),
        child: const Text('REGISTER', style: TextStyle(color: Colors.white)),
      ),
    ),
  ),

```

```

const SizedBox(height: 15),
// Already Registered? Login Text
TextButton(
  onPressed: () {
    // Navigate to login screen
    Navigator.pop(context); // Replace with correct navigation if needed
  },
  child: const Text(
    'Already registered? Login',
    style: TextStyle(fontSize: 16, color: Colors.green),
  ),
),
],
),
),
),
),
),
),
),
),
);
}
}

```



## Login



FACEBOOK



GOOGLE

or



Remember me

[TERMS & CONDITIONS](#)

LOGIN

New to the platform?

[REGISTER HERE](#)



Register

DEBUG

## Create Your Account



Full Name



Email



Phone Number



Password



Address

REGISTER

Already registered? [Login](#)

## Order Form

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

void main() {
  runApp(const MyApp());
}

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        textTheme: GoogleFonts.poppinsTextTheme(Theme.of(context).textTheme),
      ),
      home: const RegistrationForm(),
    );
  }
}

class RegistrationForm extends StatefulWidget {
  const RegistrationForm({Key? key}) : super(key: key);

  @override
  _RegistrationFormState createState() => _RegistrationFormState();
}

class _RegistrationFormState extends State<RegistrationForm> {
  final _formKey = GlobalKey<FormState>();
  final _controllers = List.generate(5, (_) => TextEditingController());
  final _focusNodes = List.generate(5, (_) => FocusNode());

  @override
  void dispose() {
    for (var controller in _controllers) {
      controller.dispose();
    }
    for (var node in _focusNodes) {
      node.dispose();
    }
    super.dispose();
  }
}
```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Container(
      decoration: const BoxDecoration(
        gradient: LinearGradient(
          begin: Alignment.topLeft,
          end: Alignment.bottomRight,
          colors: [Color(0xFFE8F5E9), Colors.white],
        ),
      ),
    child: SafeArea(
      child: CustomScrollView(
        slivers: [
          SliverAppBar(
            floating: true,
            backgroundColor: Colors.transparent,
            elevation: 0,
            flexibleSpace: FlexibleSpaceBar(
              title: Text(
                'Order Form',
                style: GoogleFonts.poppins(
                  color: Colors.green[800],
                  fontWeight: FontWeight.w600,
                ),
              ),
            ),
          ),
          SliverToBoxAdapter(
            child: Padding(
              padding: const EdgeInsets.all(24.0),
              child: Form(
                key: _formKey,
                child: Column(
                  crossAxisAlignment: CrossAxisAlignment.start,
                  children: [
                    Text(
                      'Please fill out the form to order',
                      style: GoogleFonts.poppins(
                        fontSize: 16,
                        color: Colors.green[800],
                        fontWeight: FontWeight.w500,
                      ),
                    ),
                    const SizedBox(height: 24),
                    ..._buildFormFields(),
                  ],
                ),
              ),
            ),
          ),
        ],
      ),
    ),
  );
}

```

```

        const SizedBox(height: 32),
        _buildSubmitButton(),
      ],
    ),
  ),
),
),
],
),
),
),
);
}

```

```

List<Widget> _buildFormFields() {
  final labels = ['First Name', 'Last Name', 'Address', 'Landmark', 'Mobile Number'];
  final icons = [Icons.person, Icons.person, Icons.home, Icons.location_on, Icons.phone];

  return List.generate(
    5,
    (index) => Padding(
      padding: const EdgeInsets.only(bottom: 16),
      child: TextFormField(
        controller: _controllers[index],
        focusNode: _focusNodes[index],
        decoration: _getInputDecoration(labels[index], icons[index]),
        validator: (value) {
          if (value == null || value.isEmpty) {
            return 'Please enter ${labels[index].toLowerCase()}'; // Ensures the field is required
          }
          if (index == 4 && value.length != 10) {
            return 'Mobile number must be 10 digits'; // Mobile number validation
          }
          return null;
        },
        onFieldSubmitted: (_) {
          if (index < 4) {
            FocusScope.of(context).requestFocus(_focusNodes[index + 1]);
          }
        },
        keyboardType: index == 4 ? TextInputType.phone : TextInputType.text,
      ),
    ),
  );
}

```

```

InputDecoration _getInputDecoration(String label, IconData icon) {
  return InputDecoration(
    labelText: label,
    prefixIcon: Icon(icon, color: Colors.green[800]),
    labelStyle: TextStyle(color: Colors.green[800]),
    enabledBorder: OutlineInputBorder(
      borderRadius: BorderRadius.circular(12),
      borderSide: BorderSide(color: Colors.green[200]!),
    ),
    focusedBorder: OutlineInputBorder(
      borderRadius: BorderRadius.circular(12),
      borderSide: BorderSide(color: Colors.green[800]!, width: 2),
    ),
    errorBorder: OutlineInputBorder(
      borderRadius: BorderRadius.circular(12),
      borderSide: BorderSide(color: Colors.red[400]!),
    ),
    focusedErrorBorder: OutlineInputBorder(
      borderRadius: BorderRadius.circular(12),
      borderSide: BorderSide(color: Colors.red[400]!, width: 2),
    ),
    filled: true,
    fillColor: Colors.white,
  );
}

```

```

Widget _buildSubmitButton() {
  return SizedBox(
    width: double.infinity,
    child: ElevatedButton(
      onPressed: _submitForm,
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.green[800],
        padding: const EdgeInsets.symmetric(vertical: 16),
        shape: RoundedRectangleBorder(
          borderRadius: BorderRadius.circular(12),
        ),
      ),
      child: Text(
        'SUBMIT',
        style: GoogleFonts.poppins(
          color: Colors.white,
          fontSize: 18,
          fontWeight: FontWeight.w600,
        ),
      ),
    ),
  );
}

```



```
),  
);  
}
```

```
void _submitForm() {  
  if (_formKey.currentState!.validate()) {  
    ScaffoldMessenger.of(context).showSnackBar(  
      SnackBar(  
        content: Text(  
          'Order Successful!',  
          style: GoogleFonts.poppins(),  
        ),  
        backgroundColor: Colors.green[800],  
      ),  
    );  
  }  
}
```

## Order Form

Please fill out the form to order

 First Name

 Last Name

 Address

 Landmark

 Mobile Number

SUBMIT

## Order Form

Please fill out the form to order

First Name

 Rujuta

Please enter first name

 Last Name

Please enter last name

 Address

Please enter address

 Landmark

Please enter landmark

 Mobile Number

Please enter mobile number

SUBMIT