**EXPERIMENT NO 2**

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

**Theory:**

**Widgets:**

Each element on a screen of the Flutter app is a widget. The view of the screen

completely depends upon the choice and sequence of the widgets used to build the apps and the structure of the code of an app is a tree of widgets.

**Category of Widgets:**

There are mainly 14 categories in which the flutter widgets are divided. They are mainly

segregated on the basis of the functionality they provide in a flutter application.

1. **Accessibility:** These are the set of widgets that make a flutter app more easily

accessible.

1. **Animation and Motion:** These widgets add animation to other widgets.
2. **Assets, Images, and Icons:** These widgets take charge of assets such as display images

and show icons.

1. **Async:** These provide async functionality in the flutter application.
2. **Basics:** These are the bundle of widgets that are absolutely necessary for the

development of any flutter application.

1. **Cupertino:** These are the iOS designed widgets.
2. **Input:** This set of widgets provides input functionality in a flutter application.
3. **Interaction Models:** These widgets are here to manage touch events and route users to

different views in the application.

1. **Layout:** This bundle of widgets helps in placing the other widgets on the screen as

needed.

1. **Material Components:** This is a set of widgets that mainly follow material design by

Google.

1. **Painting and effects:** This is the set of widgets that apply visual changes to their child

widgets without changing their layout or shape.

1. **Scrolling:** This provides scrollability of to a set of other widgets that are not scrollable by

default.

1. **Styling:** This deals with the theme, responsiveness, and sizing of the app.
2. **Text:** This displays text.

Description of few of the widgets are as follows:

● **Scaffold**– Implements the basic material design visual layout structure.

● **App-Bar**- To create a bar at the top of the screen.

● **Text**- To write anything on the screen.

● **Container**– To contain any widget.

● **Center**– To provide center alignment to other widgets.

**The code in main.dart:**

import 'package:flutter/material.dart';

import 'package:activibe/home.dart';

import 'package:activibe/rsvp.dart';

void main() {

runApp(MyApp());

}

class MyApp extends StatelessWidget {

@override

Widget build(BuildContext context) {

return MaterialApp(

debugShowCheckedModeBanner: false,

home: AuthScreen(),

onGenerateRoute: (settings) {

if (settings.name == '/rsvp') {

final event = settings.arguments as Map<String, dynamic>?;

return MaterialPageRoute(

builder: (context) => RsvpPage(event: event ?? {}),

);

}

return null; // Fallback to default routes

},

routes: {

'/home': (context) => HomePage(),

},

);

}

}

class AuthScreen extends StatefulWidget {

@override

\_AuthScreenState createState() => \_AuthScreenState();

}

class \_AuthScreenState extends State<AuthScreen> {

bool isLogin = true;

@override

Widget build(BuildContext context) {

return Scaffold(

body: Container(

decoration: BoxDecoration(

image: DecorationImage(

image: NetworkImage(

"https://images.unsplash.com/vector-1738323940405-3f5632492ed6?q=80&w=1800&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D"),

fit: BoxFit.cover,

),

),

child: Center(

child: Padding(

padding: EdgeInsets.all(20),

child: Column(

mainAxisAlignment: MainAxisAlignment.center,

children: [

Icon(Icons.groups, size: 100, color: Colors.black),

SizedBox(height: 20),

Text(

isLogin ? "Welcome Back!" : "Join Us Today!",

style: TextStyle(

fontSize: 24,

fontWeight: FontWeight.bold,

color: Colors.black),

),

SizedBox(height: 20),

TextField(

decoration: InputDecoration(

filled: true,

fillColor: Colors.white,

hintText: "Email",

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(10)),

),

),

SizedBox(height: 10),

TextField(

obscureText: true,

decoration: InputDecoration(

filled: true,

fillColor: Colors.white,

hintText: "Password",

border: OutlineInputBorder(

borderRadius: BorderRadius.circular(10)),

),

),

SizedBox(height: 20),

ElevatedButton(

onPressed: () {

// Navigate to the HomePage when login or signup is successful

Navigator.pushReplacementNamed(context, '/home');

},

child: Text(isLogin ? "Login" : "Signup"),

style: ElevatedButton.styleFrom(

padding: EdgeInsets.symmetric(horizontal: 50, vertical: 15),

),

),

TextButton(

onPressed: () {

setState(() {

isLogin = !isLogin;

});

},

child: Text(

isLogin

? "Don't have an account? Sign up"

: "Already have an account? Login",

style: TextStyle(

fontSize: 16,

fontWeight: FontWeight.bold,

color: Colors.black),

),

),

],

),

),

),

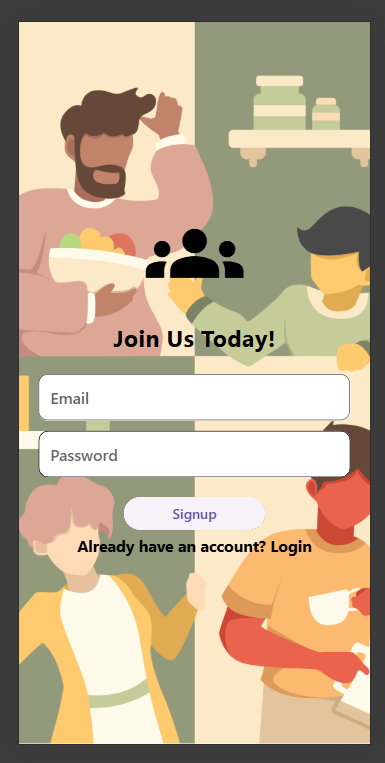
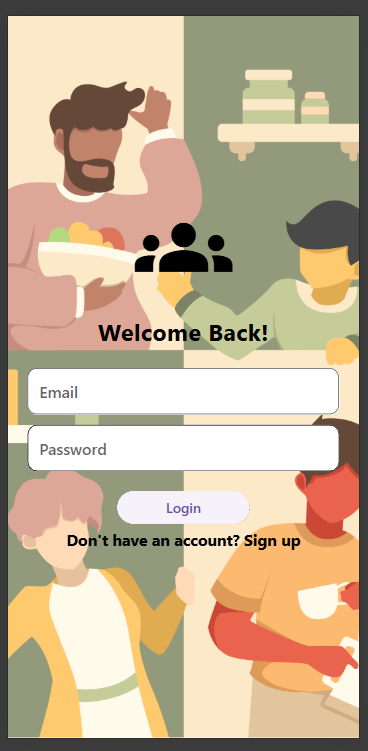
),

);

}

}

**Output:**

** **

**Signup page Login Page**

**Conclusion:**

This experiment helped in understanding the implementation of Flutter widgets to design user-friendly UI screens. Widgets are the core elements of Flutter, and using them effectively enhances UI/UX.

**GitHub Link**: <https://github.com/sadneya145/Activibe.git>