EXPERIMENT NO 3

Aim: To create a interactive form using widgets.

Theory:

An interactive form in Flutter is a user interface that allows users to enter, select, and submit data, while responding dynamically to user actions such as input validation, selections, and button clicks.

Flutter provides powerful widgets and state management to build interactive forms that are responsive, user-friendly, and platform-independent.

Key Widgets Used in Flutter Forms

1. Form Widget yu7

- Acts as a container for form fields
- Groups multiple input widgets together
- Helps in managing validation and submission

2. TextFormField

- Used to accept text input from users
- Supports validation, hints, labels, and keyboard types

- Commonly used for name, email, password, etc.

3. DropdownButtonFormField

- Allows users to select one option from a list
- Useful for selecting categories, courses, countries, etc.

4. Radio Buttons

- Used when only one option can be selected
- Commonly used for gender or choice-based questions

5. Checkbox / CheckboxListTile

- Allows multiple selections or agreement confirmation
- Often used for terms and conditions

6. Switch

- Represents on/off choices
- Used for settings like notifications or dark mode

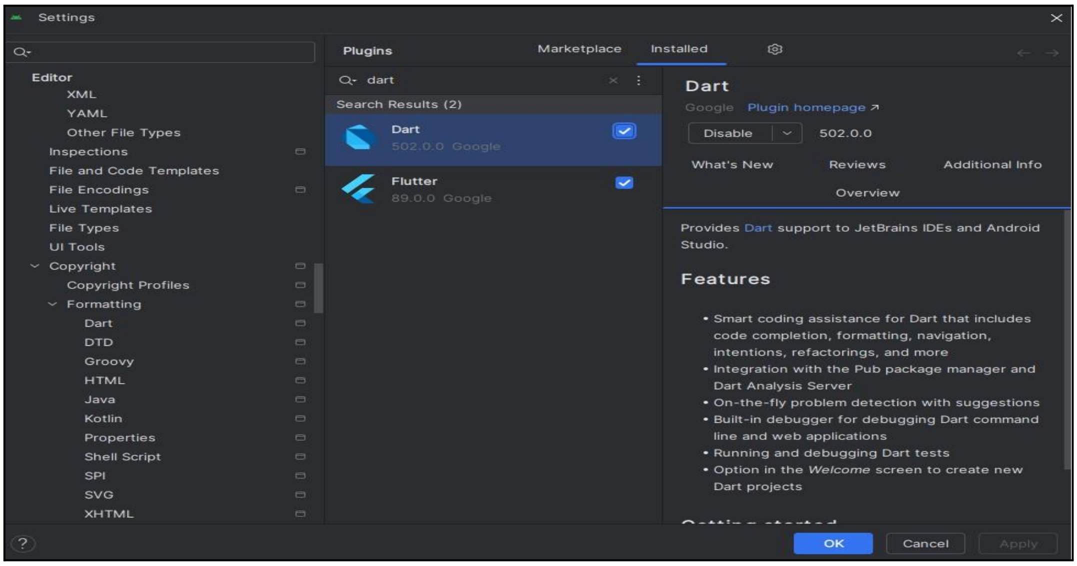
7. Buttons (ElevatedButton, TextButton)

- Trigger form submission or reset actions
- Connected to validation and data processing logic

Procedures:

STEP 1: Install Flutter & Dart Plugins in Android Studio

1. Open Android Studio
2. Go to Settings → Plugins
3. Search and Install- Flutter & Dart
4. Restart android studio



Provides Dart support to JetBrains IDEs and Android Studio.

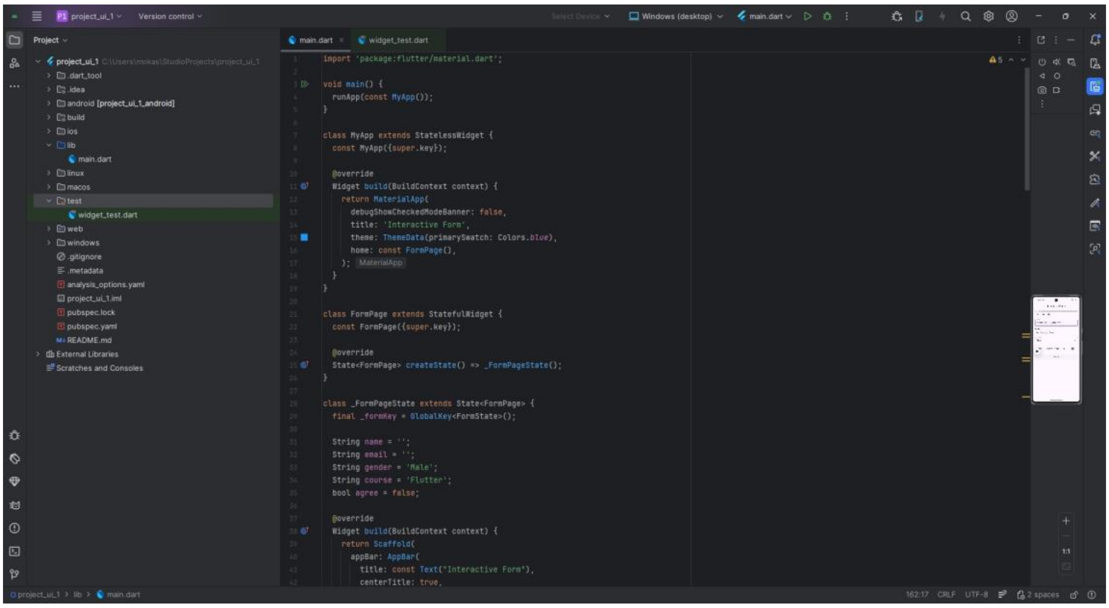
Features

- Smart coding assistance for Dart that includes code completion, formatting, navigation, intentions, refactorings, and more
- Integration with the Pub package manager and Dart Analysis Server
- On-the-fly problem detection with suggestions
- Built-in debugger for debugging Dart command line and web applications
- Running and debugging Dart tests
- Option in the *Welcome* screen to create new Dart projects

STEP 2: Create a New Flutter Project

1. Open Android Studio
2. Click New Flutter Project
3. Select Flutter Application
4. Choose Flutter SDK path (example):

   C:\flutter
5. Enter:
   - o Project name: project_ui_1
   - o Language: Dart
6. Click Finish



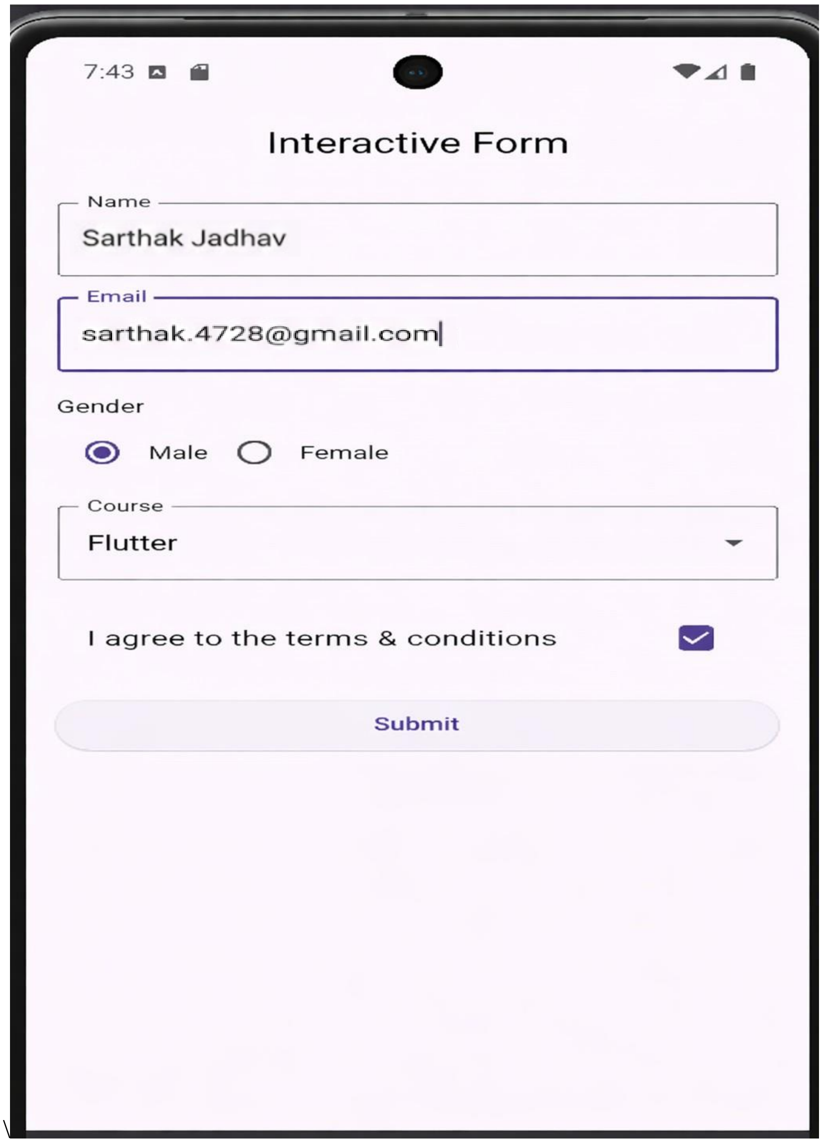
```dart
import 'package:flutter/material.dart';

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Interactive Form',
      theme: ThemeData(primarySwatch: Colors.blue),
      home: const FormPage(),
    ); MaterialApp
  }
}

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

  @override
  State<FormPage> createState() => _FormPageState();
}

class _FormPageState extends State<FormPage> {
  final _formKey = GlobalKey<FormState>();

  String name = '';
  String email = '';
  String gender = 'Male';
  String course = 'Flutter';
  bool agree = false;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Interactive Form'),
        centerTitle: true,
```

STEP 3: Code

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

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

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

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
    debugShowCheckedModeBanner: false,
    title: 'Interactive Form',
      theme: ThemeData(primarySwatch: Colors.blue),
      home: const FormPage(),
    );
  }
}

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

  @override
  State<FormPage> createState() => _FormPageState();
}

class _FormPageState extends State<FormPage> {
```

```
final _formKey = GlobalKey<FormState>();

String name = ";
String email = ";
String gender = 'Male';
String course = 'Flutter';
bool agree = false;

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: const Text("Interactive Form"), centerTitle: true),
    body: Padding(
      padding: const EdgeInsets.all(16.0),
      child: Form(
        key:  _formKey,
        child: ListView(
        children: [
            //              Name
            TextFormField(
              decoration: const InputDecoration(
                labelText: "Name",
                border: OutlineInputBorder(),
              ),
              validator: (value) {
                if (value == null || value.isEmpty) {
                  return "Enter your name";
                }
                return null;
  },
  onSaved: (value) => name = value!,
),

const SizedBox(height: 16),

// Email
TextFormField(
  decoration: const InputDecoration(
    labelText: "Email",
    border: OutlineInputBorder(),
  ),
  validator: (value) {
    if (value == null || !value.contains('@')) {
      return "Enter valid email";
    }
    return null;
  },
  onSaved: (value) => email = value!,
),

const SizedBox(height: 16),

// Gender (Radio Buttons)
const Text("Gender"),
Row(
  children: [
    Radio(
      value: "Male",
      groupValue: gender,
      onChanged: (value) {
        setState(() => gender = value.toString());
      },
    ),
    const Text("Male"),
    Radio(
      value: "Female",
      groupValue: gender,
      onChanged: (value) {
        setState(() => gender = value.toString());
      },
    ),
    const Text("Female"),
  ],
),

const SizedBox(height: 16),

// Dropdown
DropdownButtonFormField(
value: course,
  decoration: const InputDecoration(
    labelText: "Course",
    border: OutlineInputBorder(),
  ),
  items: const [
    DropdownMenuItem(value: "Flutter", child: Text("Flutter")),
    DropdownMenuItem(value: "React", child: Text("React")),
    DropdownMenuItem(value: "Angular", child: Text("Angular")),
  ],
  onChanged: (value) {
    setState(() => course = value.toString());
  },
),

const SizedBox(height: 16),

// Checkbox
CheckboxListTile(
  title: const Text("I agree to the terms & conditions"),
  value: agree,
  onChanged: (value) {
    setState(() => agree = value!);
  },
),

const SizedBox(height: 20),

// Submit Button
ElevatedButton(
onPressed: () {
    if (_formKey.currentState!.validate() && agree) {
      _formKey.currentState!.save();

      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
          content: Text(
                          "Submitted: $name, $email, $gender, $course",
                        ),
                      ),
                    );
                  } else if (!agree) {
                    ScaffoldMessenger.of(context).showSnackBar(
                    const SnackBar(
                        content: Text("Please accept terms & conditions"),
                      ),
                    );
                  }
                },
                child: const Text("Submit"),
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

STEP 6: Output



7:43

Interactive Form

Name: Sarthak Jadhav

Email: sarthak.4728@gmail.com

Gender

● Male ○ Female

Course: Flutter

I agree to the terms & conditions ☑

Conclusion:

Interactive forms in Flutter provide an efficient way to collect and validate user data using a rich set of widgets. By combining form widgets, validation techniques, and state management, Flutter enables developers to create responsive and user-friendly form interfaces for modern applications.