

## **EXPERIMENT NO: - 04**

**Name:-** Ronak Katariya

**Class:-** D15A

**Roll:No: - 23**

**AIM: -** To create an interactive Form using form widget.

---

### **Theory: -**

A form in Flutter is a structured container that collects user input through various fields like text fields, dropdowns, checkboxes, and buttons. It plays a crucial role in applications that require user data entry, such as login pages, registration forms, and feedback submissions. Flutter provides the Form widget, which works alongside TextFormField and other input elements to manage validation, state handling, and error messages efficiently. By using form validation techniques, developers can ensure data accuracy and enhance user experience.

When you create a form, it is necessary to provide the GlobalKey. This key uniquely identifies the form and allows you to do any validation in the form fields. The form widget uses child widget TextFormField to provide the users to enter the text field. This widget renders a material design text field and also allows us to display validation errors when they occur.

### **Creation of a Form**

- While creating a form in Flutter, the **Form widget** is essential as it acts as a container for grouping multiple form fields and managing validation.
- A **GlobalKey<FormState>** is required to uniquely identify the form and enable validation or data retrieval from the form fields.
- The **TextFormField widget** is used to provide input fields where users can enter data such as names, phone numbers, or email addresses.
- To enhance the appearance and usability of input fields, **InputDecoration** is used, allowing customization of labels, icons, borders, and hint text.
- Validation plays a crucial role in forms, and the **validator property** within **TextFormField** ensures user input meets specific criteria before submission.

- Different types of input require appropriate **keyboard types**, such as `TextInputType.number` for numeric fields or `TextInputType.emailAddress` for email fields.
- Proper **state management** is needed to store and retrieve user input, ensuring the form data is processed correctly.
- A **submit button** is necessary to trigger form validation and submit the collected data for further processing.

### Some Properties of Form Widget

- **key:** A `GlobalKey` that uniquely identifies the Form. You can use this key to interact with the form, such as validating, resetting, or saving its state.
- **child:** The child widget that contains the form fields. Typically, this is a `Column`, `ListView`, or another widget that allows you to arrange the form fields vertically.
- **autovalidateMode:** An enum that specifies when the form should automatically validate its fields.

### Some Methods of Form Widget

- **validate():** This method is used to trigger the validation of all the form fields within the Form. It returns `true` if all fields are valid, otherwise `false`. You can use it to check the overall validity of the form before submitting it.
- **save():** This method is used to save the current values of all form fields. It invokes the `onSaved` callback for each field. Typically, this method is called after validation succeeds.
- **reset():** Resets the form to its initial state, clearing any user-entered data.
- **currentState:** A getter that returns the current `FormState` associated with the Form.

## Code: -

### login\_screen.dart

```
import 'package:flutter/material.dart';
import 'signup_screen.dart';
import 'landing_page.dart'; // Import the
LandingPage

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

  @override
  State<LoginScreen> createState() =>
    _LoginScreenState();
}

class _LoginScreenState extends
State<LoginScreen> {
  final TextEditingController _emailController =
TextEditingController();
  final TextEditingController _passwordController =
TextEditingController();

  @override
  void dispose() {
    _emailController.dispose();
    _passwordController.dispose();
    super.dispose();
  }

  void _login(BuildContext context) {
    String email = _emailController.text;
    String password = _passwordController.text;

    if (email == 'abc@gmail.com' && password ==
'abc') {
      // Simulate successful login
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(content: Text('Login
Successful!')),
      );
      // Navigate to the LandingPage
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => const
LandingPage()),
      );
    } else {
      // Show error message for invalid credentials
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(content: Text('Invalid email or
password.')),
      );
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        decoration: const BoxDecoration(
          gradient: LinearGradient(
            begin: Alignment.topCenter,
            end: Alignment.bottomCenter,
            colors: [Color(0xFF4A00E0),
Color(0xFF8E2DE2)],
          ),
        ),
      child: Center(
        child: Padding(
          padding: const EdgeInsets.all(20.0),
          child: Column(
            mainAxisAlignment:
MainAxisAlignment.center,
            children: [
              const Text(
                "Welcome Back",
                style: TextStyle(fontSize: 28, fontWeight:
FontWeight.bold, color: Colors.white),
              ),
              const SizedBox(height: 10),
              const Text(
                "Enter your credentials to login",
                style: TextStyle(fontSize: 16, color:
Colors.white70),
              ),
              const SizedBox(height: 30),
              _buildTextField(Icons.email, "Email",
controller: _emailController),
              const SizedBox(height: 15),
              _buildTextField(Icons.lock, "Password",
isPassword: true, controller: _passwordController),
              const SizedBox(height: 20),
              ElevatedButton(
                onPressed: () {
                  _login(context);
                },
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

```

style: ElevatedButton.styleFrom(
  backgroundColor: Colors.white,
  foregroundColor: Colors.deepPurple,
  minimumSize: const
Size(double.infinity, 50),
  shape:
RoundedRectangleBorder(borderRadius:
BorderRadius.circular(30)),
),
  child: const Text("Login", style:
TextStyle(fontSize: 18)),
),
  const SizedBox(height: 10),
  TextButton(
    onPressed: () {},
    child: const Text("Forgot password?",
style: TextStyle(color: Colors.white)),
  ),
  const SizedBox(height: 10),
  Row(
    mainAxisAlignment:
MainAxisAlignment.center,
    children: [
      const Text("Don't have an account?",
style: TextStyle(color: Colors.white)),
      TextButton(
        onPressed: () {
          Navigator.push(context,
MaterialPageRoute(builder: (_) => const
SignUpScreen()));
        },
        child: const Text("Sign Up", style:
TextStyle(fontWeight: FontWeight.bold, color:
Colors.white)),
      ),
    ],
  ),
),
],
),
),
),
),
),
),
);
}

```

```

Colors.white70),
  filled: true,
  fillColor: Colors.white.withOpacity(0.2),
  border: OutlineInputBorder(borderRadius:
BorderRadius.circular(30), borderSide:
BorderSide.none),
),
);
}
}

```

```

Widget _buildTextField(IconData icon, String
hint, {bool isPassword = false, required
TextEditingController controller}) {
  return TextField(
    controller: controller,
    obscureText: isPassword,
    style: const TextStyle(color: Colors.white),
    decoration: InputDecoration(
      prefixIcon: Icon(icon, color: Colors.white),
      hintText: hint,
      hintStyle: const TextStyle(color:

```

### Code:Signup screen.dart

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

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(
        decoration: const BoxDecoration(
          gradient: LinearGradient(
            begin: Alignment.topCenter,
            end: Alignment.bottomCenter,
            colors: [Color(0xFF4A00E0),
Color(0xFF8E2DE2)],
          ),
        ),
        child: Center(
          child: Padding(
            padding: const EdgeInsets.all(20.0),
            child: Column(
              mainAxisAlignment:
MainAxisAlignment.center,
              children: [
                const Text(
                  "Sign Up",
                  style: TextStyle(fontSize: 28,
fontWeight: FontWeight.bold, color: Colors.white),
                ),
                const SizedBox(height: 10),
                const Text(
                  "Create your account",
                  style: TextStyle(fontSize: 16, color:
Colors.white70),
                ),
                const SizedBox(height: 30),
                _buildTextField(Icons.email, "Email"),
                const SizedBox(height: 15),
                _buildTextField(Icons.lock, "Password",
isPassword: true),
                const SizedBox(height: 20),
                ElevatedButton(
                  onPressed: () {}, // Add Firebase signup
logic
                  style: ElevatedButton.styleFrom(
                    backgroundColor: Colors.white,
                    foregroundColor: Colors.deepPurple,
                    minimumSize: const
Size(double.infinity, 50),
                    shape:
```

```
RoundedRectangleBorder(borderRadius:
BorderRadius.circular(30)),
                ),
                child: const Text("Sign Up", style: TextStyle(fontSize:
18)),
              ),
              const SizedBox(height: 20),
              _googleSignInButton(),
              const SizedBox(height: 10),
              Row(
                mainAxisAlignment:
MainAxisAlignment.center,
                children: [
                  const Text("Already have an account?",
style: TextStyle(color: Colors.white)),
                  TextButton(
                    onPressed: () {
                      Navigator.push(context,
MaterialPageRoute(builder: (_) => const
LoginScreen()));
                    },
                    child: const Text("Login", style:
TextStyle(fontWeight: FontWeight.bold, color:
Colors.white)),
                  ),
                ],
              ),
            ),
          ),
        ),
      );
    }
  }
}
```

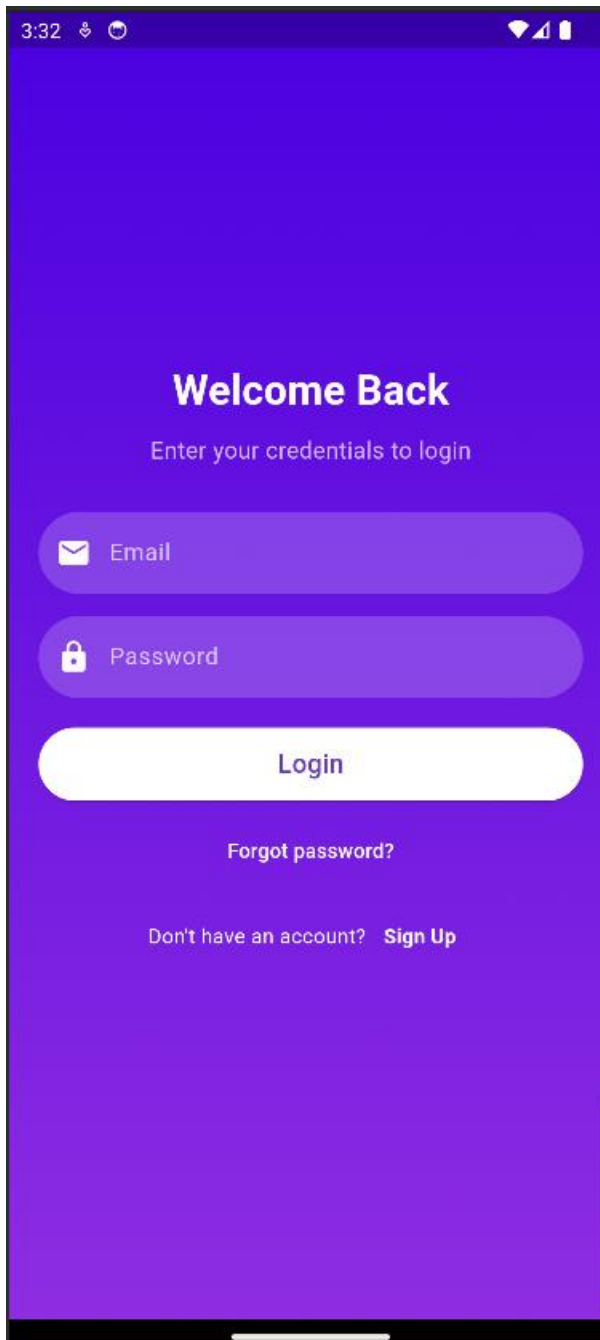
```
Widget _buildTextField(IconData icon, String hint,
{bool isPassword = false}) {
  return TextField(
    obscureText: isPassword,
    style: const TextStyle(color: Colors.white),
    decoration: InputDecoration(
      prefixIcon: Icon(icon, color: Colors.white),
      hintText: hint,
      hintStyle: const TextStyle(color: Colors.white70),
      filled: true,
      fillColor: Colors.white.withOpacity(0.2),
      border: OutlineInputBorder(borderRadius:
BorderRadius.circular(30), borderSide:
```

```
BorderSide.none),
    ),
  );
}

Widget _googleSignInButton() {
  return ElevatedButton.icon(
    onPressed: () {}, // Implement Google Sign-In
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.white,
      foregroundColor: Colors.black,
    )
  );
}
```

```
    minimumSize: const Size(double.infinity, 50),
    shape: RoundedRectangleBorder(borderRadius:
BorderRadius.circular(30)),
  ),
  icon: Image.asset("assets/images/google.png",
height: 24), // Add Google logo in assets
  label: const Text("Sign In with Google", style:
TextStyle(fontSize: 16)),
);
}
}
```

## OUTPUT: -



3:32

# Welcome Back

Enter your credentials to login

Email

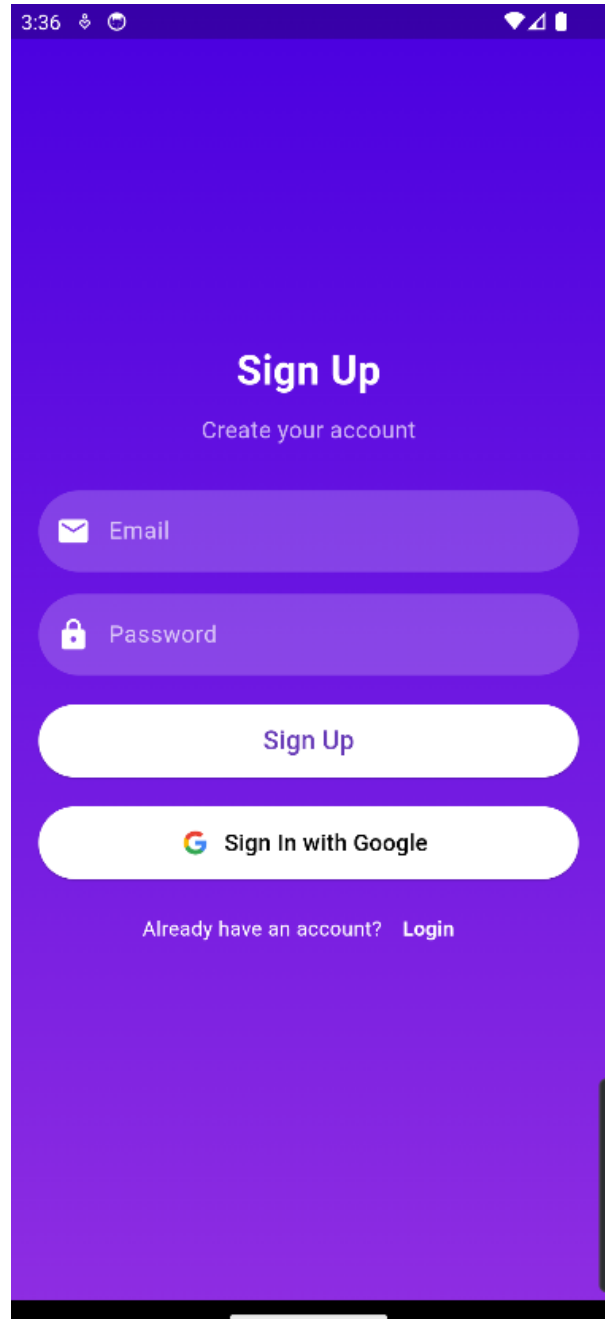
Password

Login

[Forgot password?](#)

Don't have an account? [Sign Up](#)

This is a mobile app login screen with a purple background. It features a status bar at the top showing the time 3:32 and connectivity icons. The main heading is 'Welcome Back' in white, followed by the instruction 'Enter your credentials to login'. There are two input fields for 'Email' and 'Password', each with a corresponding icon. A white 'Login' button is centered below the fields. At the bottom, there are links for 'Forgot password?' and 'Don't have an account? Sign Up'.



3:36


# Sign Up

Create your account

Email

Password

Sign Up

 Sign In with Google

Already have an account? [Login](#)

This is a mobile app sign-up screen with a purple background. It features a status bar at the top showing the time 3:36 and connectivity icons. The main heading is 'Sign Up' in white, followed by the instruction 'Create your account'. There are two input fields for 'Email' and 'Password', each with a corresponding icon. A white 'Sign Up' button is centered below the fields. Below that is a button for 'Sign In with Google' featuring the Google logo. At the bottom, there is a link for 'Already have an account? Login'.

