

Experiment No:- 04

Name:- Mayank Pahuja

Class:- D15-A RollNo:- 40

Theory: In Flutter, a "Form" is a widget that represents a container for a collection of form fields.

It helps manage the state of the form and facilitates the validation and submission of user input. Here are some key concepts and theories about forms in Flutter:

1. Widget Hierarchy:

Forms in Flutter are composed of the `Form` widget, which contains a list of `FormField` widgets. Each `FormField` represents an individual input field like text fields, checkboxes, or dropdowns.

2. Form State:

The `Form` widget maintains the state of the form, including the current values of the form fields and their validation statuses. The form state is automatically managed by Flutter.

3. Validation:

Forms provide built-in validation through the `validator` property of each `FormField`. Validators are functions that determine whether the input is valid. The form's overall validity is determined by the validity of all its fields.

4. Form Submission:

Form submission is typically triggered by a button press. The `onPressed` callback of the button can call the `FormState.save()` method, which invokes the `onSaved` callback for each form field and then calls the `onFormSaved` callback.

5. GlobalKey<FormState>:

To interact with the form state, a `GlobalKey<FormState>` is commonly used. This key allows access to the form state and is used to validate and save the form.

6. Auto-validation:

Flutter provides automatic validation by calling the `validator` function whenever the user input changes. This allows for real-time feedback to the user about the validity of their input.

7. Form Submission Lifecycle:

The form submission process involves validation, saving, and then handling the saved data. Developers can customize this process by providing their own logic within the `onSaved` and `onFormSaved` callbacks.

8. Focus Management:

Forms handle the focus of input fields, making it easy to navigate through the form using keyboard input or programmatically setting focus on specific fields.

Code & Implementation:-

Login_screen.dart & Otp_screen.dart file code & output:-

```

import
'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:get/get_core/get_core.dart';
import
'package:get/get_navigation/get_navigation.dart';
import
'package:rapido/screens/otp_screen.dart';

class PhoneHome extends StatefulWidget {
  const PhoneHome({Key? key}) : super(key:
key);
  @override
  State<PhoneHome> createState() =>
PhoneHomeState();
}
class PhoneHomeState extends
State<PhoneHome> {
  TextEditingController
phoneNumberController =
TextEditingController();
  final _formKey = GlobalKey<FormState>();
  String counterText = '0';

  sendCode() async {
    try {
      await
FirebaseAuth.instance.verifyPhoneNumber(
        phoneNumber:
'+91${phoneNumberController.text}',
        verificationCompleted:
(PHONE_AUTH_CREDENTIAL credential) {},
        verificationFailed:
(FirebaseAuthException e) {
          Get.snackbar('Error Occurred', e.code);
        },
        codeSent: (String vid, int? token) {
          Get.to(OtpPage(vid: vid));
        },
        codeAutoRetrievalTimeout: (vid) {},
      );
    } on FirebaseAuthException catch (e) {
      Get.snackbar('Error Occurred', e.code);
    } catch (e) {
      Get.snackbar('Error Occurred',
e.toString());
    }
  }
}

```

```

}
}
@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Padding(
      padding: const EdgeInsets.all(20.0),
      child: Form(
        key: _formKey,
        child: Column(
          crossAxisAlignment:
CrossAxisAlignment.start,
          children: [
            SizedBox(height: 40),
            Padding(
              padding: const EdgeInsets.only(left:
8.0),
              child: Image.asset(
                'assets/images/logo1.png',
                width: 90,
                height: 70,
              ),
            ),
            SizedBox(height: 12),
            Text(
              'What\'s your number?',
              style: TextStyle(fontSize: 20,
fontWeight: FontWeight.bold),
            ),
            SizedBox(height: 10),
            Text(
              'Enter your phone number to
proceed',
              style: TextStyle(color: Colors.grey),
            ),
            SizedBox(height: 10),
            TextFormField(
              maxLength: 10,
              keyboardType: TextInputType.phone,
              controller: phoneNumberController,
              decoration: InputDecoration(
                prefix: Text("+91 "),
                prefixStyle: TextStyle(fontSize: 15),
                prefixIcon: Icon(Icons.phone),
                counterText: '$counterText / 10',
                counterStyle: TextStyle(fontSize: 10),
                labelText: 'Number',
              ),
            ),
          ],
        ),
      ),
    ),
  );
}

```

```

        hintText: 'Enter your phone
number',
        hintStyle: TextStyle(fontSize: 10,
color: Colors.grey),
    ),
    onChanged: (value) {
        setState() {
            counterText =
value.length.toString();
        });
    },
    validator: (value) {
        if (value!.isEmpty) {
            return 'Please enter your phone
number';
        } else if (value.length != 10) {
            return 'Phone number must be 10
digits';
        }
        return null;
    },
),

```

```
SizedBox(height: 20),  
    Spacer(), // Move the button to the  
bottom of the screen  
    ElevatedButton(  
        onPressed: (counterText == '10')  
            ? () {  
                if  
(_formKey.currentState!.validate()) {  
                    sendCode();  
                }  
            }  
            : null, // Disable button if phone  
number is not 10 digits  
        child: Padding(  
            padding: const  
EdgeInsets.symmetric(horizontal: 90),  
            child: Text(  
                'Received OTP',  
                style: TextStyle(  
                    fontSize: 18.0,  
                    ),),),),),),)); }
```

#Otpscreen Code:-

```
// ignore_for_file: prefer_const_constructors

import
'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:get/get.dart';
import 'package:pininput/pininput.dart';
import 'package:rapido/wrapper.dart';
class OtpPage extends StatefulWidget {
  final String vid;
  const OtpPage({super.key, required this.vid});
  @override
  State<OtpPage> createState() =>
OtpPageState();
}
class OtpPageState extends State<OtpPage> {
  var code = "";
  signIn() async {
    PhoneAuthCredential credential =
PhoneAuthProvider.credential(
```

```

verificationId: widget.vid,
smsCode: code,
);
try {
  await
  FirebaseAuth.instance.signInWithCredential(cr
  edential).then((value) {
    Get.offAll(() => Wrapper());
  });
} on FirebaseAuthException catch (e) {
  Get.snackbar('Error Occurred', e.code);
} catch (e) {
  Get.snackbar('Error Occurred',
e.toString());
}
}
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    resizeToAvoidBottomInset: true,
    body: SingleChildScrollView(

```

```

        child: Column(
          crossAxisAlignment:
CrossAxisAlignment.center,
          children: [
            SizedBox(height: 40),
            Padding(
              padding: const EdgeInsets.only(left:
8.0),
              child: Image.asset(
                'assets/images/logo1.png',
                width: 90,
                height: 70,
              ),
            ),
            SizedBox(height: 20),
            Center(
              child: Text(
                "OTP Verification",
                style: TextStyle(fontSize: 30,
fontWeight: FontWeight.bold),
              ),
            ),
            Padding(
              padding: const
EdgeInsets.symmetric(horizontal: 25, vertical:
6),
              child: Text(
                "Enter OTP",
                textAlign: TextAlign.center,
                style: TextStyle(fontSize: 16),
              ),
            ),
            SizedBox(height: 20),
            textcode(),
            SizedBox(height: 40),
            button(),
          ],
        ),
      ),
    );
  }

  Widget button() {
    return Center(
      child: ElevatedButton(
        onPressed: () {
          signIn();
        },
        child: Padding(

```

```

padding: const
EdgeInsets.symmetric(horizontal: 80),
        child: Text(
          'Verify & Proceed',
          style: TextStyle(
            fontSize: 18,
          ),
        ),
      ),
    );
  }

  Widget textcode() {
    return Center(
      child: Padding(
        padding: const EdgeInsets.all(6.0),
        child: Pinput(
          length: 6,
          onChanged: (value) {
            setState() {
              code = value;
            }
          },
        ),
      ),
    );
  }
}

```

- Until User can't enter the 10 Digit Phone number the Otp Authentication & login to the app is not to be done.

