**Name:**Mohd Sufiyan Siddiqui
**Class :** D15A
**Roll NO :** 55

# Lab - 04

**Aim:To create an interactive Form using form widget**

## Theory:

Creating an interactive form using the form widget in Flutter involves several steps. Here's a theoretical outline to guide you through the process:

1. Import Required Libraries : Begin by importing the necessary libraries in your Flutter project. Typically, you'll need the `flutter/material.dart` library for UI components and `flutter/widgets.dart` for widgets.

2. Design UI Layout : Plan the layout of your form UI. Decide on the fields you want to include (text fields, checkboxes, dropdowns, etc.) and how you want them arranged on the screen.

3. Implement Form Widget : Use the `Form` widget to wrap your form fields. The `Form` widget automatically handles form validation and submission.

4. Add Form Fields : Inside the `Form` widget, add form fields using widgets like `TextFormField`, `Checkbox`, `DropdownButton`, etc., depending on your requirements.

5. Validation : Implement validation for the form fields to ensure that users enter data correctly. You can use the `validator` property of `TextFormField` or custom validation logic.

6. Handle Form Submission : Implement logic to handle form submission. This typically involves attaching a callback function to the form's `onSubmitted` or `onSaved` property.

7. Display Validation Errors : If form validation fails, display error messages to guide users on what needs to be corrected. You can use the `errorText` property of form fields for this purpose.

8. Testing : Thoroughly test your interactive form to ensure that it behaves as expected. Test different scenarios, such as valid and invalid input, to verify that your validation logic works correctly.

9. Refinement and Optimization : Refine your form based on user feedback and optimize its performance if necessary. Consider factors like usability, accessibility, and responsiveness.

10. Documentation and Maintenance : Document your code to make it easier for others (and yourself) to understand. Also, be prepared to maintain and update your form as needed, especially if the underlying Flutter framework evolves.

Remember that this is a high-level overview, and the actual implementation details may vary depending on your specific requirements and preferences. Additionally, you can refer to the Flutter documentation and community resources for more detailed guidance and examples.

**Code:**

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:flutter_application_1/pages/bottomnav.dart';
import 'package:flutter_application_1/pages/forgotpassword.dart';
import 'package:flutter_application_1/pages/home.dart';
import 'package:flutter_application_1/pages/signup.dart';
import 'package:flutter_application_1/widget/widget_support.dart';

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

  @override
  State<LogIn> createState() => _LogInState();
}

class _LogInState extends State<LogIn> {
  String email = "", password = "";

  final _formkey = GlobalKey<FormState>();

  TextEditingController useremailcontroller = new TextEditingController();
  TextEditingController userpasswordcontroller = new TextEditingController();

  userLogin() async {
    try {
      await FirebaseAuth.instance
          .signInWithEmailAndPassword(email: email, password: password);
      Navigator.push(
          context, MaterialPageRoute(builder: (context) => BottomNav()));
    } on FirebaseAuthException catch (e) {
      if (e.code == 'user-not-found') {
        ScaffoldMessenger.of(context).showSnackBar(SnackBar(
            content: Text(
          "No User Found for that Email",
          style: TextStyle(fontSize: 18.0, color: Colors.black),
```

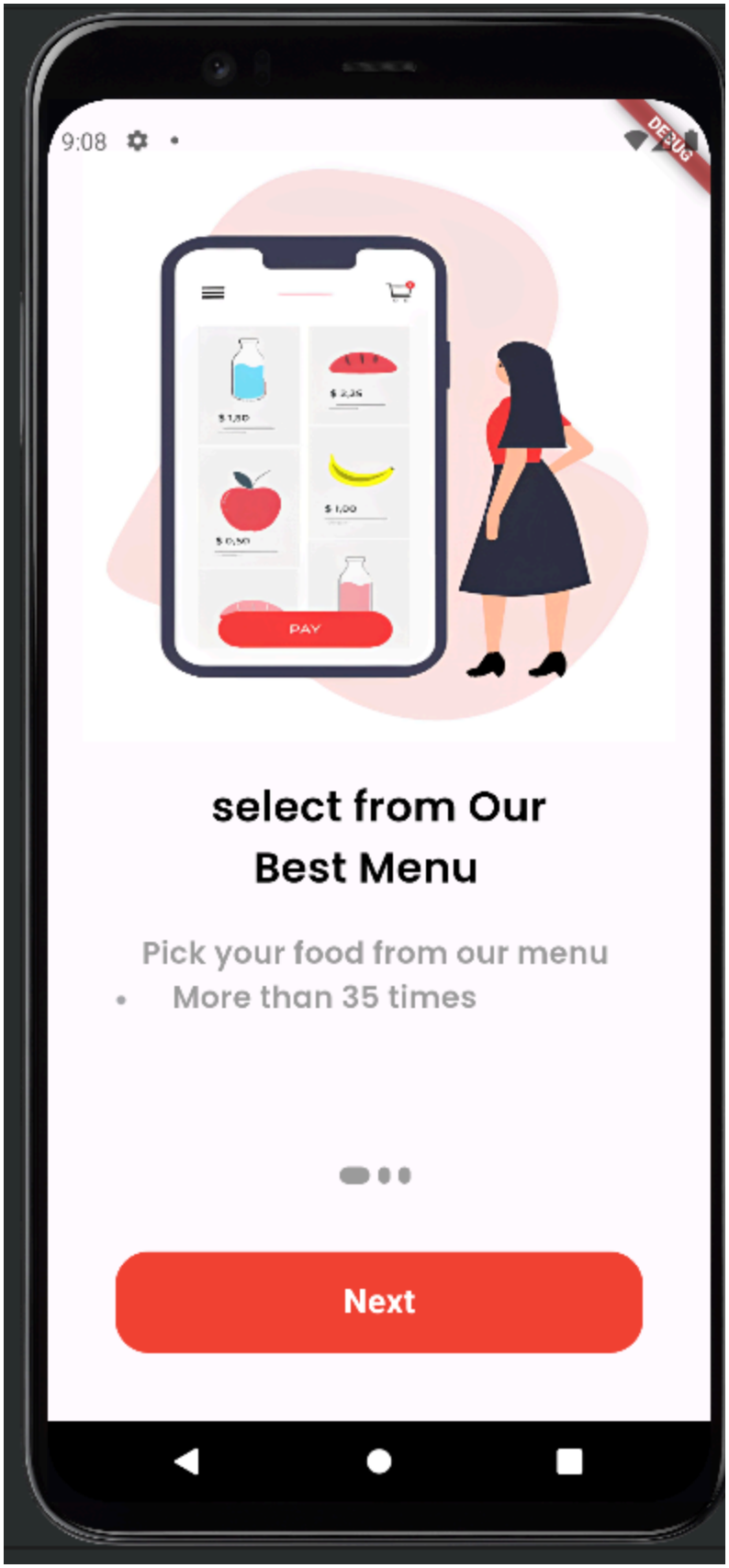
```
      )));
    } else if (e.code == 'wrong-password') {
      ScaffoldMessenger.of(context).showSnackBar(SnackBar(
          content: Text(
        "Wrong Password Provided by User",
        style: TextStyle(fontSize: 18.0, color: Colors.black),
      )));
    }
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: Container(
      child: Stack(
        children: [
          Container(
            width: MediaQuery.of(context).size.width,
            height: MediaQuery.of(context).size.height / 2.5,
            decoration: BoxDecoration(
                gradient: LinearGradient(
                    begin: Alignment.topLeft,
                    end: Alignment.bottomRight,
                    colors: [
                  Color(0xFFff5c30),
                  Color(0xFFe74b1a),
                ])),
          ),
          Container(
            margin:
                EdgeInsets.only(top: MediaQuery.of(context).size.height / 3),
            height: MediaQuery.of(context).size.height / 2,
            width: MediaQuery.of(context).size.width,
            decoration: BoxDecoration(
                color: Colors.white,
                borderRadius: BorderRadius.only(
                    topLeft: Radius.circular(40),
                    topRight: Radius.circular(40))),
            child: Text(""),
          ),
          Container(
            margin: EdgeInsets.only(top: 10.0, left: 20.0, right: 20.0),
            child: Column(
```

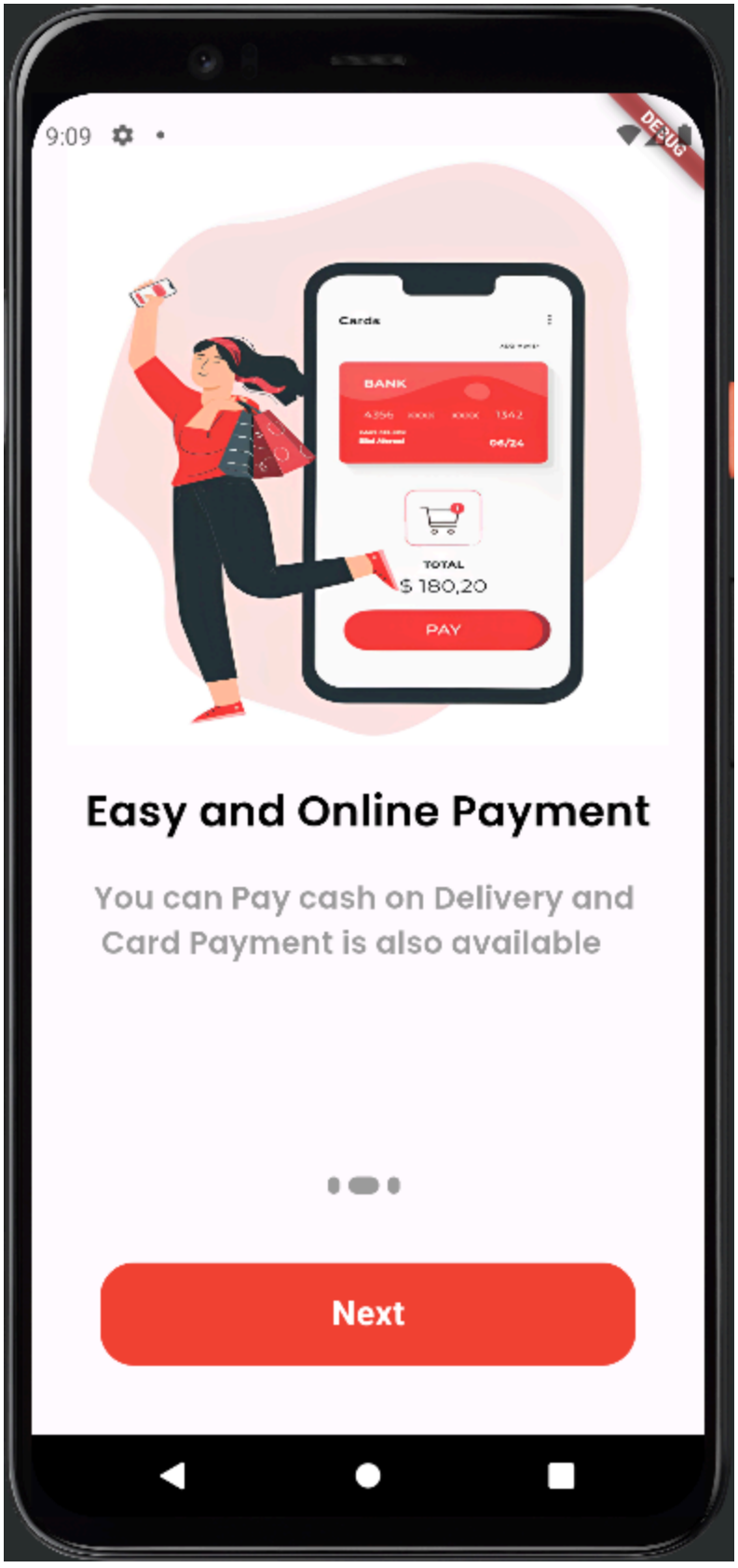
```
children: [
  Center(
     child: Image.asset(
   "images/logo.png",
   width: MediaQuery.of(context).size.width / 1.5,
   fit: BoxFit.cover,
  )),
  SizedBox(
   height: 30.0,
  ),
  Material(
   elevation: 5.0,
   borderRadius: BorderRadius.circular(20),
   child: Container(
    padding: EdgeInsets.only(left: 20.0, right: 20.0),
    width: MediaQuery.of(context).size.width,
    height: MediaQuery.of(context).size.height / 2,
    decoration: BoxDecoration(
       color: Colors.white,
       borderRadius: BorderRadius.circular(20)),
    child: Form(
     key: _formkey,
     child: Column(
      children: [
       SizedBox(
        height: 10.0,
       ),
       Text(
        "Login",
        style: AppWidget.HeadlineTextFeildStyle(),
       ),
       SizedBox(
        height: 30.0,
       ),
       TextFormField(
        controller: useremailcontroller,
        validator: (value) {
         if (value == null || value.isEmpty) {
          return 'Please Enter Email';
         }
         return null;
        },
        decoration: InputDecoration(
           hintText: 'Email',
```

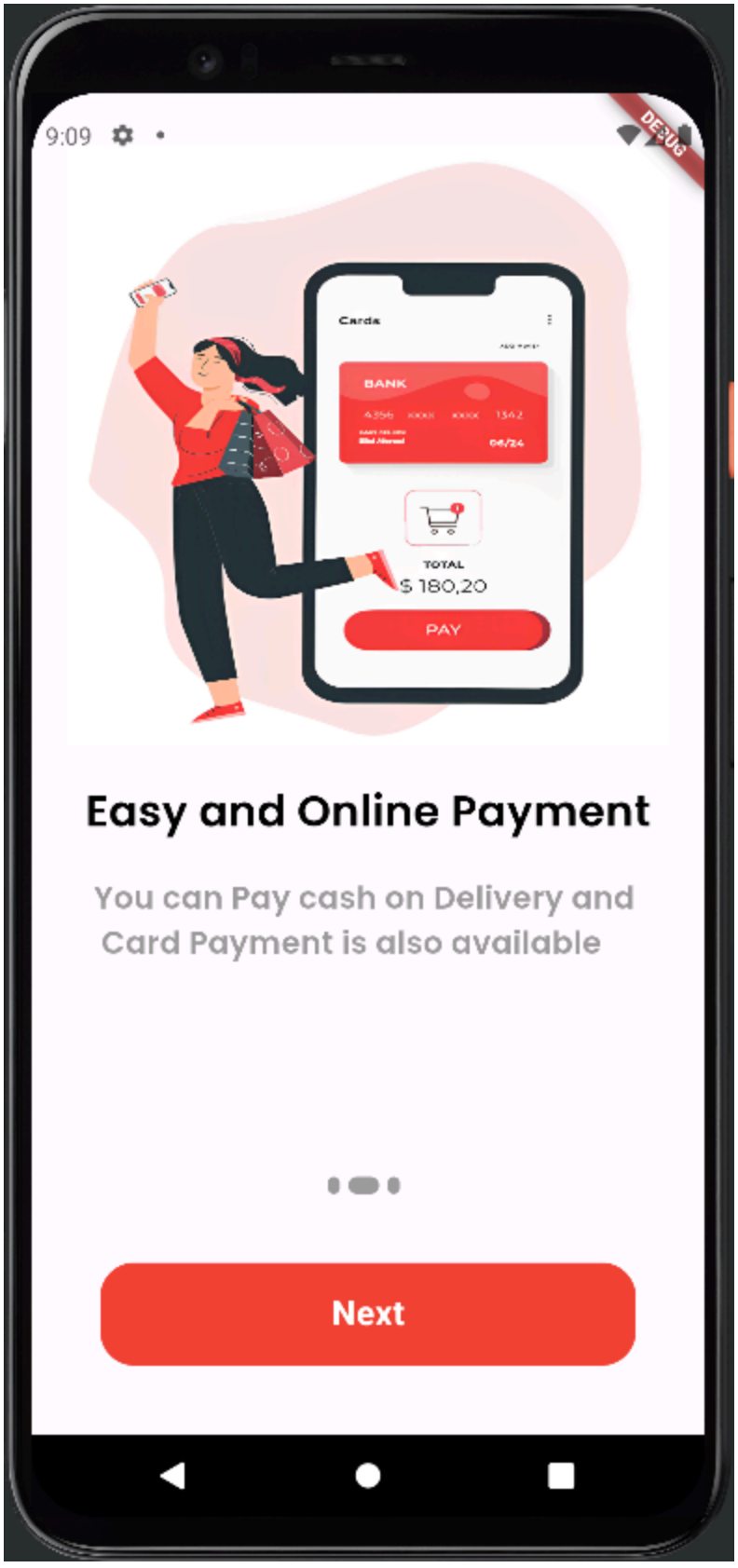
```
      hintStyle: AppWidget.semiBoldTextFeildStyle(),
      prefixIcon: Icon(Icons.email_outlined)),
),
SizedBox(
  height: 30.0,
),
TextFormField(
  controller: userpasswordcontroller,
  validator: (value) {
    if (value == null || value.isEmpty) {
      return 'Please Enter Password';
    }
    return null;
  },
  obscureText: true,
  decoration: InputDecoration(
      hintText: 'Password',
      hintStyle: AppWidget.semiBoldTextFeildStyle(),
      prefixIcon: Icon(Icons.password_outlined)),
),
SizedBox(
  height: 20.0,
),
GestureDetector(
  onTap: () {
    Navigator.push(
        context,
        MaterialPageRoute(
            builder: (context) =>
                ForgotPassword()));
  },
  child: Container(
      // alignment: Alignment.topRight,
      alignment: Alignment.center,
      child: Text(
        "Forgot Password?",
        style: AppWidget.semiBoldTextFeildStyle(),
      )),
),
SizedBox(
  height: 30.0,
),
GestureDetector(
  onTap: () {
```

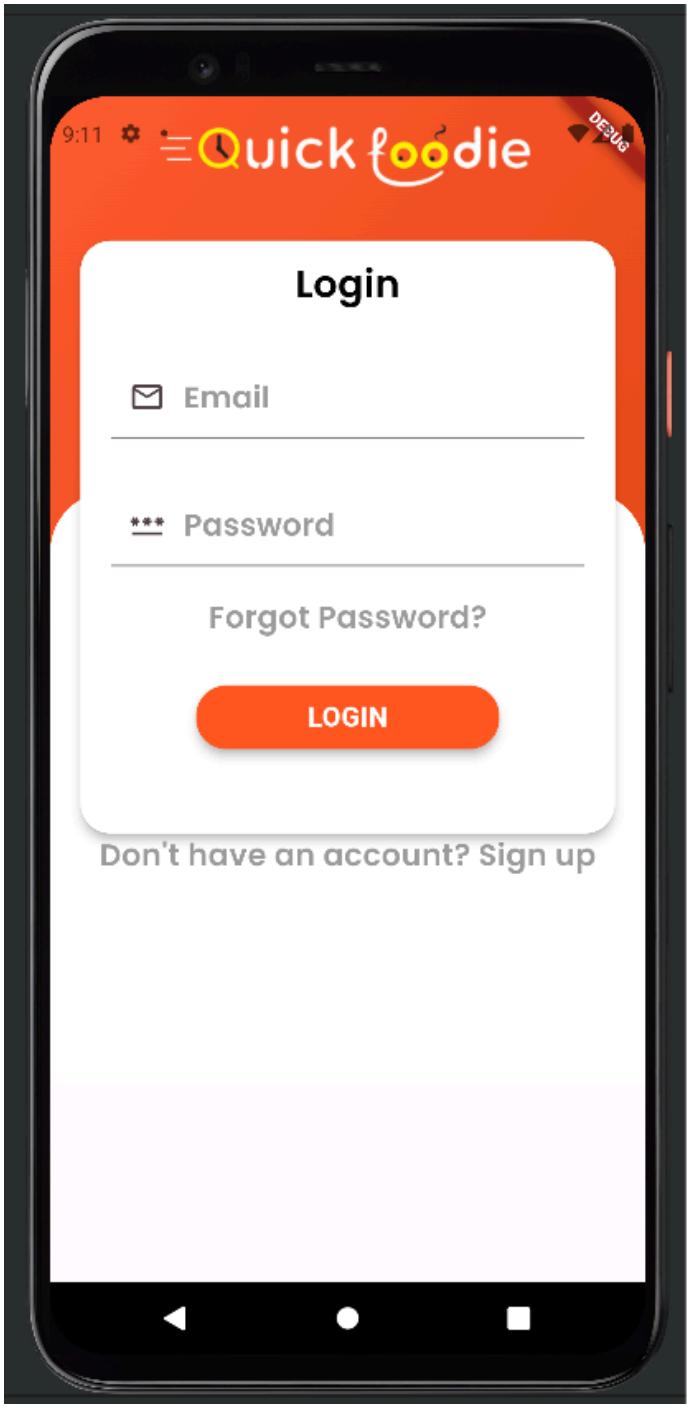
```
            if (_formkey.currentState!.validate()) {
              setState(() {
                email = useremailcontroller.text;
                password = userpasswordcontroller.text;
              });
            }
            userLogin();
          },
          child: Material(
            elevation: 5.0,
            borderRadius: BorderRadius.circular(20),
            child: Container(
              padding: EdgeInsets.symmetric(vertical: 8.0),
              width: 200,
              decoration: BoxDecoration(
                  color: Color(0Xffff5722),
                  borderRadius: BorderRadius.circular(20)),
              child: Center(
                  child: Text(
                "LOGIN",
                style: TextStyle(
                    color: Colors.white,
                    fontSize: 18.0,
                    fontFamily: 'Poppins1',
                    fontWeight: FontWeight.bold),
              )),
            ),
          ),
        ),
      ],
    ),
  ),
),
SizedBox(
  height: 0.0,
),
GestureDetector(
    onTap: () {
      Navigator.push(context,
          MaterialPageRoute(builder: (context) => SignUp()));
    },
    child: Text(
      "Don't have an account? Sign up",
```

```
                  style: AppWidget.semiBoldTextFeildStyle(),
                ))
            ],
          ),
        )
      ],
    ),
  ),
);
  }
}
```


9:08
DEBUG
PAY
select from Our
Best Menu
Pick your food from our menu
More than 35 times
Next


9:09
DEBUG
Cards
BANK
TOTAL
$ 180,20
PAY
Easy and Online Payment
You can Pay cash on Delivery and
Card Payment is also available
Next


9:09
DEBUG
Cards
BANK
TOTAL
$ 180,20
PAY
Easy and Online Payment
You can Pay cash on Delivery and
Card Payment is also available
Next


9:11
Quick foodie
DEBUG
Login
Email
Password
Forgot Password?
LOGIN
Don't have an account? Sign up