PRANAV RAIKAR
D15A 47
Experiment 4

AIM: To create interactive form using form widget.

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.FocusManagement:
Formshandlethefocusof input fields,makingiteasytonavigatethroughtheformusing keyboardinputorprogrammaticallysettingfocusonspecificfields.

9.FormKeyandGlobalKey:
Usinga`GlobalKey<FormState>`allowsformorecontrolovertheform,suchastriggering formvalidationorresettingtheform. It isusuallydefinedasaglobalkeyinthewidget tree.

10.FormPersistence:
Formdatacanbepersistedacrossdifferentscreensorappsessionsbypassingthedata downthewidget treeorusingstatemanagementsolutionslikeProviderorRiverpod.
Formsplayacrucial roleinuserinteraction,datacollection,andvalidationinFlutter

applications,providingastructuredandefficientwaytohandleuserinput.

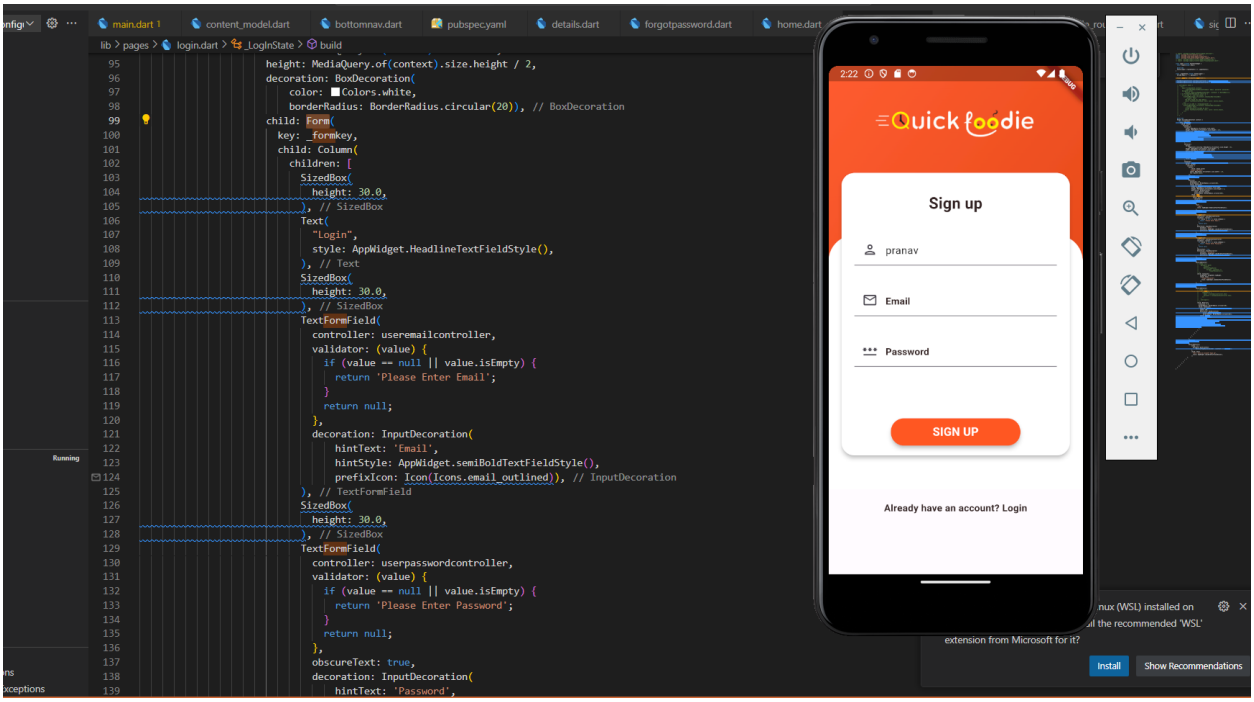
Code:

```
// import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:food_panda/pages/signup.dart';
import 'package:food_panda/widget/widget_support.dart';
// import 'package:sample_flutter/pages/bottomnav.dart';
// import 'package:sample_flutter/pages/forgotpassword.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: 60.0, 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: 50.0,
      ),
      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: 30.0,
                ),
                Text(
                  "Login",
                  style: AppWidget.HeadlineTextFieldStyle(),
                ),
                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.semiBoldTextFieldStyle(),
      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.semiBoldTextFieldStyle(),
      prefixIcon: Icon(Icons.password_outlined)),
),
SizedBox(
  height: 20.0,
),
GestureDetector(
  // onTap: () {
  //   Navigator.push(
  //       context,
  //       MaterialPageRoute(
  //           builder: (context) =>
  //               ForgotPassword()));
  // },
  child: Container(
      alignment: Alignment.topRight,
      child: Text(
        "Forgot Password?",
        style: AppWidget.semiBoldTextFieldStyle(),
      )),
),
SizedBox(
  height: 80.0,
),
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: 70.0,
        ),
        GestureDetector(
            onTap: () {
              Navigator.push(context,
                  MaterialPageRoute(builder: (context) => SignUp()));
            },
            child: Text(
              "Don't have an account? Sign up",
              style: AppWidget.semiBoldTextFieldStyle(),
            ))
      ],
    ),
  )
],
),
),
);
}
}
```



Conclusion :Thus I learnt to create interactive form widgets