

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:

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.

9.FormKey and GlobalKey:

Using a `GlobalKey<FormState>` allows for more control over the form, such as triggering form validation or resetting the form. It is usually defined as a global key in the widget tree.

10.Form Persistence:

Form data can be persisted across different screens or app sessions by passing the data down the widget tree or using state management solutions like `Provider` or `Riverpod`. Forms play a crucial role in user interaction, data collection, and validation in Flutter.

applications, providing a structured and efficient way to handle user input.

Code:

Login

```
// import 'package:firebase_auth/firebase_auth.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:food_panda/pages/bottomnav.dart';
import 'package:food_panda/pages/forgotpassword.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(0xFFf5c30),
                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,

```

```

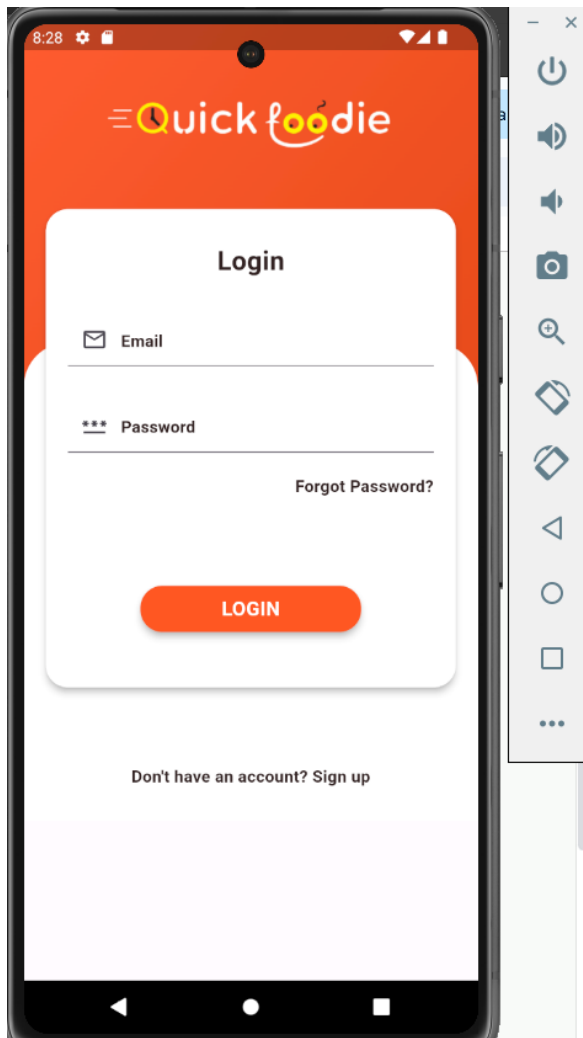
        child: Text(
          "Forgot Password?",
          style: AppWidget.semiBoldTextFieldStyle(),
        )),
      ),
      SizedBox(
        height: 80.0,
      ),
      GestureDetector(
        onTap: () {
          if (_formkey.currentState!.validate()) {
            setState(() {
              email = 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(0Xfff5722),
              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(),
      ),
    ],
  ),
],
),
),
);
}
}

```



Signup:

Code:

```

// import 'package:firebase_auth/firebase_auth.dart';

import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:food_panda/pages/bottomnav.dart';
import 'package:food_panda/pages/login.dart';

```

```

import 'package:food_panda/service/database.dart';
import 'package:food_panda/service/shared_pref.dart';
import 'package:food_panda/widget/widget_support.dart';

import 'package:random_string/random_string.dart';
// import 'package:sample_flutter/pages/bottomnav.dart';

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

  @override
  State<SignUp> createState() => _SignUpState();
}

class _SignUpState extends State<SignUp> {
  String email = "", password = "", name = "";

  TextEditingController namecontroller = new TextEditingController();

  TextEditingController passwordcontroller = new TextEditingController();

  TextEditingController mailcontroller = new TextEditingController();

  final _formkey = GlobalKey<FormState>();

  registration() async {
    if (password != null) {
      try {
        UserCredential userCredential = await FirebaseAuth.instance
          .createUserWithEmailAndPassword(email: email, password: password);

        ScaffoldMessenger.of(context).showSnackBar((SnackBar(
          backgroundColor: Colors.redAccent,
          content: Text(
            "Registered Successfully",
            style: TextStyle(fontSize: 20.0),
          )),);
        String Id = randomAlphaNumeric(10);
        Map<String, dynamic> addUserInfo = {
          "Name": namecontroller.text,
          "Email": mailcontroller.text,
          "Wallet": "0",
          "Id": Id,
        };
        await DatabaseMethods().addUserDetail(addUserInfo, Id);
        await SharedPreferencesHelper().saveUserName(namecontroller.text);
        await SharedPreferencesHelper().saveUserEmail(mailcontroller.text);
        await SharedPreferencesHelper().saveUserWallet('0');
        await SharedPreferencesHelper().saveUserId(Id);

        // ignore: use_build_context_synchronously
        Navigator.pushReplacement(
          context, MaterialPageRoute(builder: (context) => BottomNav()));
      } on FirebaseException catch (e) {

```

```

    if (e.code == 'weak-password') {
      ScaffoldMessenger.of(context).showSnackBar(SnackBar(
        backgroundColor: Colors.orangeAccent,
        content: Text(
          "Password Provided is too Weak",
          style: TextStyle(fontSize: 18.0),
        )),
    );
    } else if (e.code == "email-already-in-use") {
      ScaffoldMessenger.of(context).showSnackBar(SnackBar(
        backgroundColor: Colors.orangeAccent,
        content: Text(
          "Account Already exists",
          style: TextStyle(fontSize: 18.0),
        )),
    );
  }
}
}

@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 / 1.8,
          decoration: BoxDecoration(
            color: Colors.white,
            borderRadius: BorderRadius.circular(20)),
        child: Form(
          key: _formkey,
          child: Column(
            children: [
              SizedBox(
                height: 30.0,
              ),
              Text(
                "Sign up",
                style: AppWidget.HeadlineTextFieldStyle(),
              ),
              SizedBox(
                height: 30.0,
              ),
              TextFormField(
                controller: namecontroller,
                validator: (value) {
                  if (value == null || value.isEmpty) {
                    return 'Please Enter Name';
                  }
                  return null;
                },
                decoration: InputDecoration(
                  hintText: 'Name',
                  hintStyle: AppWidget.semiBoldTextFieldStyle(),
                  prefixIcon: Icon(Icons.person_outlined)),
              ),
              SizedBox(
                height: 30.0,
              ),
              TextFormField(
                controller: mailcontroller,
                validator: (value) {
                  if (value == null || value.isEmpty) {
                    return 'Please Enter E-mail';
                  }
                }
              )
            ],
          ),
        ),
      ),
    ),
  ),
);

```

```


        return null;
    },
    decoration: InputDecoration(
        hintText: 'Email',
        hintStyle: AppWidget.semiBoldTextFieldStyle(),
        prefixIcon: Icon(Icons.email_outlined)),
    ),
    SizedBox(
        height: 30.0,
    ),
    TextFormField(
        controller: passwordcontroller,
        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: 80.0,
    ),
    GestureDetector(
        onTap: () async {
            if (_formkey.currentState!.validate()) {
                setState(() {
                    email = mailcontroller.text;
                    name = namecontroller.text;
                    password = passwordcontroller.text;
                });
            }
            registration();
        },
        child: Material(
            elevation: 5.0,
            borderRadius: BorderRadius.circular(20),
            child: Container(
                padding: EdgeInsets.symmetric(vertical: 8.0),
                width: 200,
                decoration: BoxDecoration(
                    color: Color(0Xfff5722),
                    borderRadius: BorderRadius.circular(20)),
                child: Center(
                    child: Text(
                        "SIGN UP",
                        style: TextStyle(
                            color: Colors.white,
                            fontSize: 18.0,
                            fontFamily: 'Poppins1',


```


8:31

Quick Foodie

Sign up

 Name

 Email

 Password

SIGN UP

Already have an account? [Login](#)

AddItem

Code:-

```
import 'dart:io';

import 'package:firebase_storage/firebase_storage.dart';
import 'package:flutter/material.dart';
import 'package:food_panda/service/database.dart';
import 'package:food_panda/widget/widget_support.dart';
import 'package:image_picker/image_picker.dart';
import 'package:random_string/random_string.dart';
// import 'package:sample_flutter/widget/widget_support.dart';

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

  @override
  State<AddFood> createState() => _AddFoodState();
}

class _AddFoodState extends State<AddFood> {
  final List<String> fooditems = ['Ice-cream', 'Burger', 'Salad', 'Pizza'];
  String? value;
  TextEditingController namecontroller = new TextEditingController();
  TextEditingController pricecontroller = new TextEditingController();
  TextEditingController detailcontroller = new TextEditingController();
  final ImagePicker _picker = ImagePicker();
  File? selectedImage;

  Future getImage() async {
    var image = await _picker.pickImage(source: ImageSource.gallery);
    selectedImage = File(image!.path);
    setState(() {});
  }

  uploadItem() async {
    if (selectedImage != null &&
        namecontroller.text != "" &&
        pricecontroller.text != "" &&
        detailcontroller.text != "") {
      String addId = randomAlphaNumeric(10);
      Reference firebaseStorageRef =
        FirebaseStorage.instance.ref().child("blogImages").child(addId);
      final UploadTask task = firebaseStorageRef.putFile(selectedImage!);

      var downloadUrl = await (await task).ref.getDownloadURL();

      Map<String, dynamic> addItem = {
        "Image": downloadUrl,
        "Name": namecontroller.text,
        "Price": pricecontroller.text,
        "Detail": detailcontroller.text
      };
      await DatabaseMethods().addFoodItem(addItem, value!).then((value) {
        ScaffoldMessenger.of(context).showSnackBar((SnackBar(
```

```

        backgroundColor: Colors.redAccent,
        content: Text(
          "Food item added successfully",
          style: TextStyle(fontSize: 20.0),
        )),
      ));
    }
  }
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      leading: GestureDetector(
        onTap: () {
          Navigator.pop(context);
        },
        child: Icon(
          Icons.arrow_back_ios_new_outlined,
          color: Color(0xFF373866),
        )),
      centerTitle: true,
      title: Text(
        "Add Item",
        style: AppWidget.HeadlineTextFieldStyle(),
      ),
    ),
    body: SingleChildScrollView(
      child: Container(
        margin:
          EdgeInsets.only(left: 20.0, right: 20.0, top: 20.0, bottom: 50.0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Text(
              "Upload the Item Picture",
              style: AppWidget.semiBoldTextFieldStyle(),
            ),
            SizedBox(
              height: 20.0,
            ),
            selectedImage == null
              ? GestureDetector(
                  onTap: () {
                    getImage();
                  },
                  child: Center(
                    child: Material(
                      elevation: 4.0,
                      borderRadius: BorderRadius.circular(20),
                      child: Container(
                        width: 150,
                        height: 150,
                        decoration: BoxDecoration(

```



```

    SizedBox(
      height: 30.0,
    ),
    Text(
      "Item Price",
      style: AppWidget.semiBoldTextFieldStyle(),
    ),
    SizedBox(
      height: 10.0,
    ),
    Container(
      padding: EdgeInsets.symmetric(horizontal: 20.0),
      width: MediaQuery.of(context).size.width,
      decoration: BoxDecoration(
        color: Color(0xFFececfc8),
        borderRadius: BorderRadius.circular(10)),
      child: TextField(
        controller: pricecontroller,
        decoration: InputDecoration(
          border: InputBorder.none,
          hintText: "Enter Item Price",
          hintStyle: AppWidget.LightTextFieldStyle(),
        ),
      ),
    ),
    SizedBox(
      height: 30.0,
    ),
    Text(
      "Item Detail",
      style: AppWidget.semiBoldTextFieldStyle(),
    ),
    SizedBox(
      height: 10.0,
    ),
    Container(
      padding: EdgeInsets.symmetric(horizontal: 20.0),
      width: MediaQuery.of(context).size.width,
      decoration: BoxDecoration(
        color: Color(0xFFececfc8),
        borderRadius: BorderRadius.circular(10)),
      child: TextField(
        maxLines: 6,
        controller: detailcontroller,
        decoration: InputDecoration(
          border: InputBorder.none,
          hintText: "Enter Item Detail",
          hintStyle: AppWidget.LightTextFieldStyle(),
        ),
      ),
    ),
    SizedBox(
      height: 20.0,
    ),
    Text(
      "Select Category",

```

```

        style: AppWidget.semiBoldTextFieldStyle(),
      ),
      SizedBox(
        height: 20.0,
      ),
      Container(
        padding: EdgeInsets.symmetric(horizontal: 10.0),
        width: MediaQuery.of(context).size.width,
        decoration: BoxDecoration(
          color: Color(0xFFececfc8),
          borderRadius: BorderRadius.circular(10)),
        child: DropdownButtonHideUnderline(
          child: DropdownButton<String>(
            items: fooditems
              .map((item) => DropdownMenuItem<String>(
                value: item,
                child: Text(
                  item,
                  style:
                    TextStyle(fontSize: 18.0, color: Colors.black),
                )),
              .toList(),
            onChanged: ((value) => setState(() {
              this.value = value;
            })),
            dropdownColor: Colors.white,
            hint: Text("Select Category"),
            iconSize: 36,
            icon: Icon(
              Icons.arrow_drop_down,
              color: Colors.black,
            ),
            value: value,
          )),
      ),
      SizedBox(
        height: 30.0,
      ),
      GestureDetector(
        onTap: () {
          uploadItem();
        },
        child: Center(
          child: Material(
            elevation: 5.0,
            borderRadius: BorderRadius.circular(10),
            child: Container(
              padding: EdgeInsets.symmetric(vertical: 5.0),
              width: 150,
              decoration: BoxDecoration(
                color: Colors.black,
                borderRadius: BorderRadius.circular(10)),
              child: Center(
                child: Text(

```


The screenshot shows the 'Add Item' screen of the 'Inventory Management' app. The interface is clean and modern, with a white background and light blue accents. The title 'Add Item' is prominently displayed at the top. Below it, there's a section for uploading an item picture, represented by a large square with a camera icon. The form consists of four main sections: 'Item Name' with a text input field, 'Item Price' with a text input field, 'Item Detail' with a larger text area, and 'Select Category' with a dropdown menu. The bottom of the screen shows the standard Android navigation bar. On the right side of the image, there's a vertical toolbar with various icons for app navigation and system functions.

The screenshot shows the 'Add Item' screen of the 'Inventory Management System' app. The interface is clean and modern, with a white background and light blue accents. The title 'Add Item' is prominently displayed at the top. Below it, there's a section for uploading an item picture, represented by a large square with a camera icon. The form consists of four main sections: 'Item Name' with a text input field, 'Item Price' with a text input field, 'Item Detail' with a larger text area, and 'Select Category' with a dropdown menu. The bottom of the screen shows the standard Android navigation bar. On the right side, there's a vertical toolbar with various icons for system functions like power, volume, camera, search, and window management.