

### **Aim:**

To learn how to create an interactive form using the Form widget in Flutter to collect and validate user input.

### **Theory:**

Flutter's Form widget is used to manage form elements, including text fields, checkboxes, and other input widgets. It allows for form validation, which ensures that user inputs are correct before submission. The TextFormField widget is commonly used for text input in forms. The FormState object helps in managing the state of the form, including validation and resetting the form.

### **Code:**

```
import 'package:cloud_firestore/cloud_firestore.dart';
import "package:flutter/material.dart";
import "package:nutrilab/authservice.dart";
import 'dart:developer';
import "package:nutrilab/navbarwidget.dart";
import "./deco.dart";
class GoToRegisterPage extends StatefulWidget {
  const GoToRegisterPage({super.key});
  @override
  State<GoToRegisterPage> createState() => _GoToRegisterPageState();
}
class _GoToRegisterPageState extends State<GoToRegisterPage> {
  final _auth = AuthService();
  final _name = TextEditingController();
  final _email = TextEditingController();
  final _password = TextEditingController();
  final _mobile = TextEditingController();
  @override
  void dispose() {
    super.dispose();
    _email.dispose();
    _password.dispose();
    _name.dispose();
    _mobile.dispose();
  }
  Future addUser(String name, email, int mobile) async {
    await FirebaseFirestore.instance.collection('users').doc(email).set({
      'name': name,
      'email': email,
      'mobile': mobile,
      'liked': [],
    });
  }
}
```

```

    'cart': {},
    'address': {},
  });
}
@override
Widget build(BuildContext context) {
  final screenWidth = MediaQuery.of(context).size.width;
  // Base font sizes
  final double baseFontSize = screenWidth * 0.05;
  final double smallFontSize = baseFontSize * 0.65;
  final double mediumFontSize = baseFontSize * 0.8;
  final double largeFontSize = baseFontSize * 1.4;
  return Scaffold(
    backgroundColor: Color.fromARGB(255, 225, 226, 209),
    body: Center(
      child: Container(
        width: MediaQuery.of(context).size.width - 40,
        child: Column(
          children: [
            Spacer(flex: 16),
            Text(
              "Start an account today!",
              style: TextStyle(
                fontFamily: 'Gayathri',
                fontWeight: FontWeight.w700,
                fontSize: mediumFontSize,
              ),
            ),
            Spacer(flex: 4),
            TextFormField(
              controller: _name,
              cursorColor: Color.fromARGB(255, 24, 79, 87),
              style: TextStyle(
                color: const Color.fromARGB(255, 0, 0, 0),
                fontFamily: 'Gayathri',
                fontWeight: FontWeight.w700,
                fontSize: mediumFontSize,
              ),
              decoration: myDecorationField.copyWith(hintText: 'Name'),
              keyboardType: TextInputType.emailAddress,
            ),
            Spacer(),
            TextFormField(
              controller: _email,
              cursorColor: Color.fromARGB(255, 24, 79, 87),
              style: TextStyle(
                color: const Color.fromARGB(255, 0, 0, 0),

```

```

        fontFamily: 'Gayathri',
        fontWeight: FontWeight.w700,
        fontSize: mediumFontSize,
      ),
      decoration: myDecorationField.copyWith(hintText: 'Email'),
      keyboardType: TextInputType.emailAddress,
    ),
    Spacer(),
    TextFormField(
      obscureText: true,
      controller: _password,
      cursorColor: Color.fromARGB(255, 24, 79, 87),
      style: TextStyle(
        color: const Color.fromARGB(255, 0, 0, 0),
        fontFamily: 'Gayathri',
        fontWeight: FontWeight.w700,
        fontSize: mediumFontSize,
      ),
      decoration: myDecorationField.copyWith(hintText: 'Password'),
    ),
    Spacer(),
    TextFormField(
      controller: _mobile,
      cursorColor: Color.fromARGB(255, 24, 79, 87),
      style: TextStyle(
        color: const Color.fromARGB(255, 0, 0, 0),
        fontFamily: 'Gayathri',
        fontWeight: FontWeight.w700,
        fontSize: mediumFontSize,
      ),
      decoration: myDecorationField.copyWith(hintText: 'Mobile'),
      keyboardType: TextInputType.number,
    ),
    Spacer(),
    SizedBox(
      width: MediaQuery.of(context).size.width - 40,
      child: ElevatedButton(
        onPressed: _signup,
        style: ElevatedButton.styleFrom(
          padding: EdgeInsets.fromLTRB(100, 5, 100, 15),
          backgroundColor: Color.fromARGB(255, 24, 79, 87),
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(20),
          ),
        ),
        child: Text(
          "SIGN UP",

```

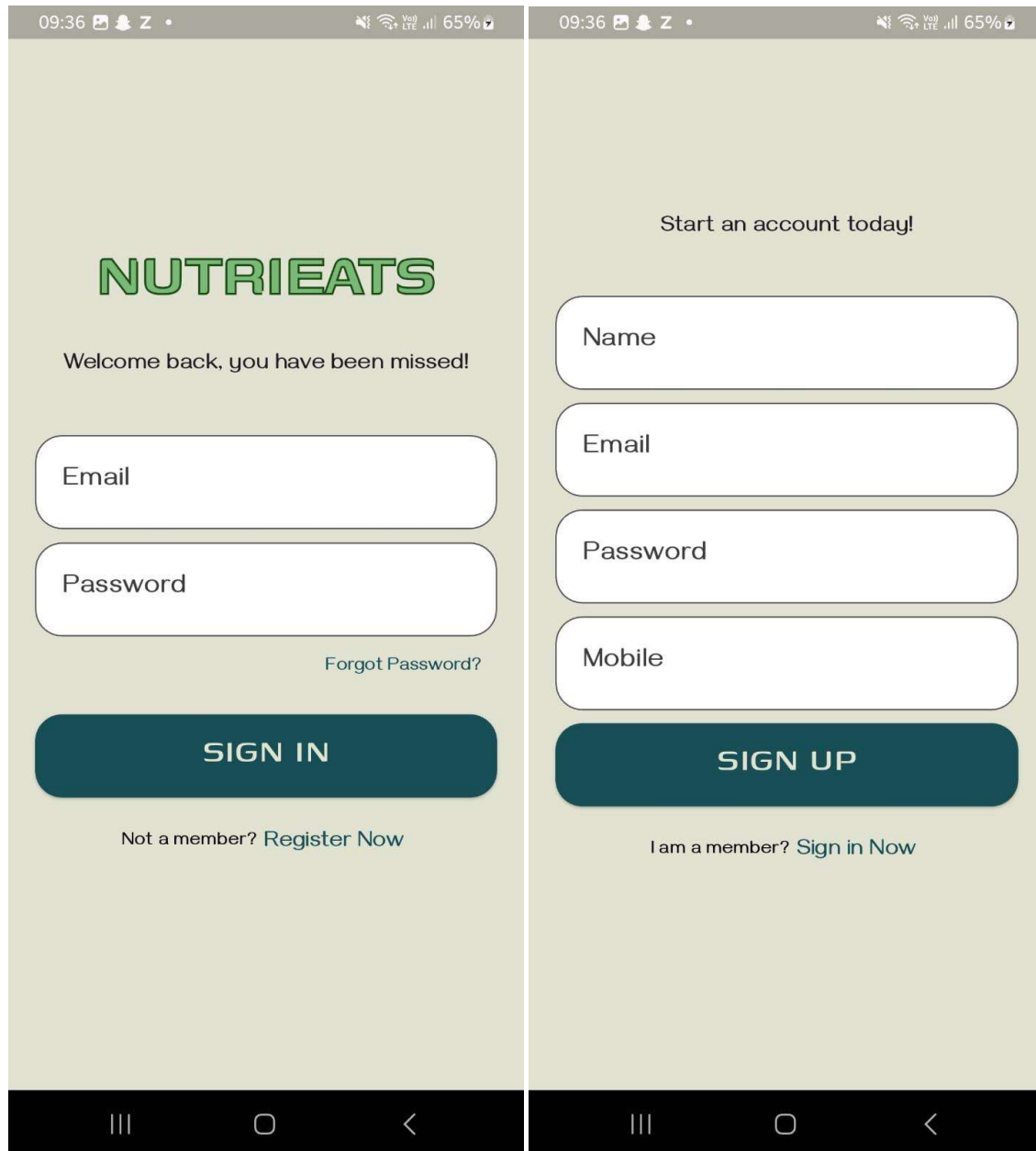


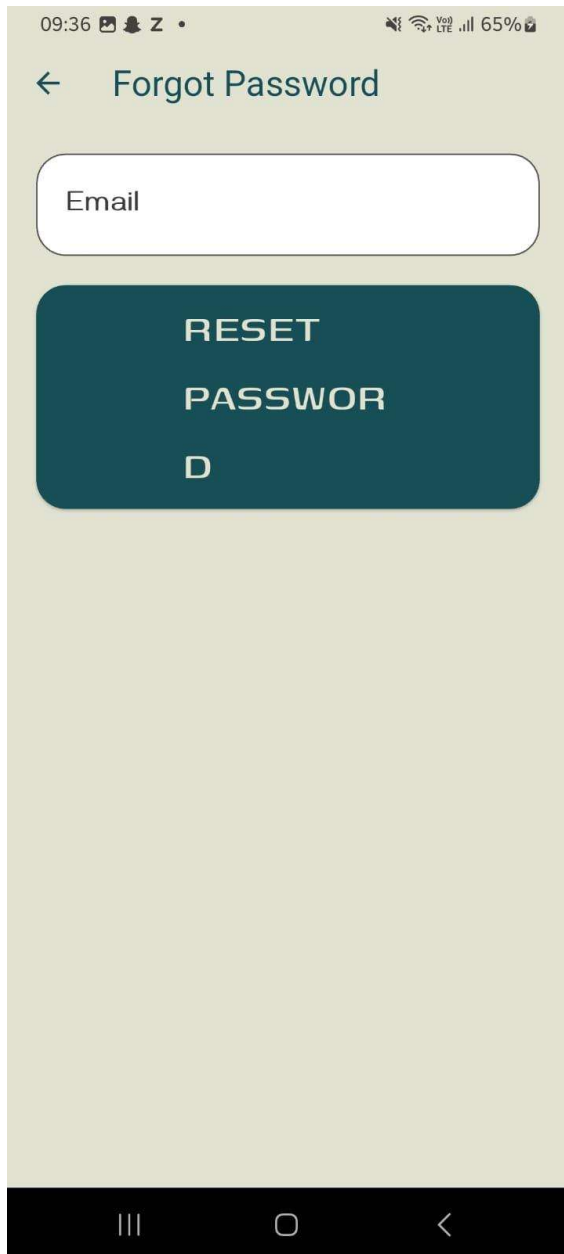
```

    );
  }
  goToHome(BuildContext context) => Navigator.push(
    context,
    MaterialPageRoute(
      builder: (context) => BottomNav(cart: {}), // Pass an empty cart map
    ),
  );
  _signup() async {
    final user = await _auth.createUserWithEmailAndPassword(
      context, _email.text, _password.text);
    if (user != null) {
      log("User Created Successfully");
      addUser(_name.text, _email.text, int.parse(_mobile.text));
      goToHome(context);
    }
  }
}

```

**Output:**





**Conclusion:** This experiment demonstrated how to create an interactive form in Flutter using the Form widget, allowing for efficient input collection and validation. It highlighted the importance of form management and validation in building user-friendly and error-free applications.