

Name: Varun Khubani
Division: D15A
Roll No: 30
Batch: B

MPL Experiment No 4

Aim: To implement an interactive form using the Form widget in Flutter for the Nike Shoes project.

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:

- In the Nike Shoes project, the Form widget will serve as the container for various form fields required for user interaction.
- Each form field, such as user registration details or adding items to the cart, will be represented by individual TextFormField widgets within the Form.

2. Form State

- The Form widget will manage the state of the form, including the current values of form fields and their validation statuses.
- Flutter's built-in mechanisms will handle the management of the form state automatically.

3. Validation:

- Forms will include built-in validation using the validator property of each TextFormField widget.
- Validators will be implemented to ensure the validity of user input, such as ensuring proper email format or password strength.

4. Form Submission:

- Form submission will be triggered by actions like user registration or adding items to the cart.
- The onPressed callback of relevant buttons will call the save() method of the FormState to save the form data.

5. GlobalKey<FormState>:

- Interaction with the form state will be facilitated using a GlobalKey<FormState>.
- This key will provide access to the form state, enabling tasks such as form validation or resetting.

6. Auto-validation:

- Automatic validation will be implemented by invoking the validator function whenever user input changes.
- This will provide real-time feedback to users regarding the validity of their input.

7. Form Submission Lifecycle:

- The form submission process will involve validation, saving, and handling the saved data.
- Custom logic will be implemented within the onSaved and onFormSaved callbacks to handle these stages effectively.

8. *Focus Management:

- Forms will handle focus management, allowing users to navigate through the form using keyboard input or programmatically setting focus on specific fields.

9. Form Persistence:

- Form data persistence across different screens or app sessions will be achieved by passing the data down the widget tree or utilizing state management solutions like Provider or Riverpod.

Code Snippets:

```
1  import 'package:firebase_auth/firebase_auth.dart';
2  import 'package:flutter/material.dart';
3  import 'package:kick_bazaar_app/main.dart';
4
5  class MySignup extends StatefulWidget {
6    const MySignup({Key? key}) : super(key: key);
7
8    @override
9    State<MySignup> createState() => _MySignupState();
10 }
11
12 class _MySignupState extends State<MySignup> {
13   TextEditingController emailController = TextEditingController();
14
15   // TextEditingController emailTextEditingController = TextEditingController();
16   TextEditingController passwordTextEditingController = TextEditingController();
17   TextEditingController phoneNoEditingController
18     TextEditingController phoneNoEditingController = TextEditingController();
19   TextEditingController
20     package:kick_bazaar_app/signup.dart ();
21   TextEditingController phoneNoEditingController = TextEditingController();
22   // TextEditingController cityTextEditingController = TextEditingController();
23   // TextEditingController countryTextEditingController = TextEditingController();
24   // TextEditingController profileHeadingTextEditingController =
25   //   TextEditingController();
26   //
27   // TextEditingController heightTextEditingController = TextEditingController();
28   // TextEditingController weightTextEditingController = TextEditingController();
29   // TextEditingController bodytypeTextEditingController = TextEditingController();
30   // TextEditingController drinkTextEditingController = TextEditingController();
31   // TextEditingController smokeTextEditingController = TextEditingController();
32   // TextEditingController haveChildrenTextEditingController =
33   //   TextEditingController();
34   // TextEditingController noOfChildrenTextEditingController =
35   //   TextEditingController();
36   // TextEditingController professionTextEditingController =
37   //   TextEditingController();
```

```

void createAccount() async {
  String email = _emailController.text.trim();
  String password = passwordTextEditingController.text.trim();
  String cpassword = confirmpasswordTextEditingController.text.trim();

  if (email == "" || password == "" ) {
    print('Please fill all details');
  } else {
    try {
      UserCredential userCredential = await FirebaseAuth.instance
        .createUserWithEmailAndPassword(email: email, password: password);

      if (userCredential.user != null) {
        Navigator.push(
          context,
          MaterialPageRoute(builder: (context) => NavigationPage()),
        );
      }

      print('Account Created Succesfully');
    } on FirebaseAuthException catch (ex) {
      print(ex.code.toString());
    }
  }
}

@override
Widget build(BuildContext context) {
  return Container(
    decoration: const BoxDecoration(
      image: DecorationImage(
        image: AssetImage(
          "/Users/varunkhubani/Downloads/Kick-Bazaar-App/assets/signup.png"), // AssetImage
        fit: BoxFit.cover), // DecorationImage // BoxDecoration
    child: Scaffold(
      appBar: AppBar(

```

```

Widget build(BuildContext context) {
  return Container(
    decoration: const BoxDecoration(
      image: DecorationImage(
        image: AssetImage(
          "/Users/varunkhubani/Downloads/Kick-Bazaar-App/assets/signup.png"), // AssetImage
        fit: BoxFit.cover), // DecorationImage // BoxDecoration
    child: Scaffold(
      appBar: AppBar(
        elevation: 0,
        backgroundColor: Colors.transparent,
      ), // AppBar
      backgroundColor: Colors.transparent,
      body: Stack(
        children: [
          Container(
            padding: const EdgeInsets.only(left: 35, top: 70),
            child: const Text(
              "Create Account",
              style: TextStyle(
                color: Colors.white,
                fontSize: 33.0,
              ), // TextStyle
            ), // Text
          ), // Container
          SingleChildScrollView(
            child: Container(
              padding: EdgeInsets.only(
                top: MediaQuery.of(context).size.height * 0.25,
                left: 30.0,
                right: 30.0), // EdgeInsets.only
              child: Column(
                children: <Widget>[
                  TextField(
                    controller: UsernameTextEditingController,

```

```
Widget build(BuildContext context) {
```

```
    TextField(  
      controller: UsernameTextEditingController,  
      decoration: InputDecoration(  
        fillColor: Colors.white10,  
        filled: true,  
        border: OutlineInputBorder(  
          borderRadius: BorderRadius.all(Radius.circular(30)),  
        ), // OutlineInputBorder  
        hintText: "Username",  
        hintStyle: TextStyle(fontSize: 18.0),  
      ), // InputDecoration  
    ), // TextField  
    const SizedBox(  
      height: 25.0,  
    ), // SizedBox  
    TextField(  
      controller: _emailController,  
      decoration: InputDecoration(  
        fillColor: Colors.white10,  
        filled: true,  
        border: OutlineInputBorder(  
          borderRadius: BorderRadius.all(Radius.circular(30)),  
        ), // OutlineInputBorder  
        hintText: "Email",  
        hintStyle: TextStyle(fontSize: 18.0),  
      ), // InputDecoration  
    ), // TextField  
    const SizedBox(  
      height: 25.0,  
    ), // SizedBox  
    TextField(  
      controller: passwordTextEditingController,  
      decoration: InputDecoration(  
        fillColor: Colors.white10,  
        filled: true,
```

```

class _MySignupState extends State<MySignup> {
  Widget build(BuildContext context) {
    // // Row
    // // CircleAvatar
    ],
  ), // Row
  const SizedBox(
    height: 25.0,
  ), // SizedBox
  Column(
    children: <Widget>[
      const Text("Already have an Account? ",
        style: TextStyle(
          fontWeight: FontWeight.bold, fontSize: 20.0)), // TextStyle // Text
      const SizedBox(
        height: 10,
      ), // SizedBox
      TextButton(
        style: TextButton.styleFrom(
          padding: const EdgeInsets.only(
            left: 20, top: 10, right: 20, bottom: 10), // EdgeInsets.only
          backgroundColor: Colors.black26),
        onPressed: () {
          Navigator.pop(context);
        },
        child: const Text(
          "Login",
          style: TextStyle(
            fontSize: 20.0,
            color: Colors.white,
            decoration: TextDecoration.underline), // TextStyle
        ), // Text
      ), // TextButton
    ], // <Widget>[]
  ), // Column
], // <Widget>[]
), // Column
), // Container

```

```

class ForgetPassword extends StatefulWidget {
  const ForgetPassword({Key? key}) : super(key: key);

  @override
  State<ForgetPassword> createState() => _ForgetPasswordState();
}

class _ForgetPasswordState extends State<ForgetPassword> {
  @override
  Widget build(BuildContext context) {
    return Container(
      decoration: const BoxDecoration(
        image: DecorationImage(
          image: AssetImage("/Users/varunkhubani/Downloads/Kick-Bazaar-App/assets/signup.png"), fit: BoxFit.cover)),
      child: Scaffold(
        appBar: AppBar(
          elevation: 0,
          backgroundColor: Colors.transparent,
        ), // AppBar
        backgroundColor: Colors.transparent,
        body: Stack(
          children: [
            Container(
              padding: const EdgeInsets.only(left: 35, top: 90),
              child: const Text(
                "Reset Password",
                style: TextStyle(
                  color: Colors.white,
                  fontSize: 33.0,
                ), // TextStyle
              ), // Text
            ), // Container
            Container(
              padding: const EdgeInsets.only(left: 35, top: 180),
              child: const Text(

```

```

Widget build(BuildContext context) {
  ), // Container
  Container(
    padding: const EdgeInsets.only(left: 35, top: 180),
    child: Text(
      "Enter the E-mail associated with your account and we will send you the instruction to reset your Password",
      style: TextStyle(
        color: Colors.black54,
        fontSize: 20.0,
        // fontStyle: FontStyle.italic,
        fontWeight: FontWeight.w400), // TextStyle
    ), // Text
  ), // Container
  SingleChildScrollView(
    child: Container(
      padding: EdgeInsets.only(
        top: MediaQuery.of(context).size.height * 0.4,
        left: 30.0,
        right: 30.0), // EdgeInsets.only
      child: Column(
        children: <Widget>[
          const TextField(
            decoration: InputDecoration(
              fillColor: Colors.white10,
              filled: true,
              border: OutlineInputBorder(
                borderRadius: BorderRadius.all(Radius.circular(30)),
              ), // OutlineInputBorder
              hintText: "Your Email",
              hintStyle: TextStyle(fontSize: 18.0),
            ), // InputDecoration
          ), // TextField
          const SizedBox(
            height: 25.0,
          ), // SizedBox
          Row(

```



```

class _ForgetPasswordState extends State<ForgetPassword> {
  Widget build(BuildContext context) {
    Row(
      mainAxisAlignment: MainAxisAlignment.end,
      children: [
        // const Text(
        //   "Signup ",
        //   style: TextStyle(
        //     fontSize: 25.0,
        //     color: Colors.black,
        //     fontWeight: FontWeight.w700),
        // ),
        CircleAvatar(
          backgroundColor: Colors.black45,
          radius: 25.0,
          child: IconButton(
            icon: const Icon(Icons.forward),
            color: Colors.white,
            onPressed: () {
              Navigator.pop(context);
            },
          ), // IconButton
        ), // CircleAvatar
      ],
    ), // Row
    const SizedBox(
      height: 25.0,
    ), // SizedBox
  ], // <Widget>[]
), // Column
), // Container
) // SingleChildScrollView
],
), // Stack
)); // Scaffold // Container
}

```

Conclusion:

Implementing an interactive form using the Form widget in Flutter for the Nike Shoes project will enhance user experience by providing structured and efficient interfaces for data collection and validation. By understanding the concepts associated with forms in Flutter and their practical application within the project, developers can create a seamless and intuitive user interaction flow.