Name:Deversh Wadhwa
Roll no: 68

EXPERIMENT NO: 04

Aim: To create an interactive Form using form widget

Theory :

Creating forms in Flutter involves several key steps, and understanding the theory behind it can help you build robust and user-friendly interfaces. Let's break it down into key components:

1. **State Management**:
   - Forms often require managing the state of input fields. In Flutter, you can manage state using StatefulWidget or state management solutions like Provider, Bloc, Riverpod, etc.
   - State management ensures that as users interact with the form (typing, selecting options, etc.), the UI updates to reflect the changes.
2. **Input Widgets**:
   - Flutter provides various input widgets like TextField, TextFormField, DropdownButton, Checkbox, Radio, etc., to collect different types of user input.
   - Each input widget has properties to customize its appearance, behavior, validation, and error handling.
3. **Validation**:
   - Validating user input is crucial for data integrity and user experience. Flutter provides built-in validators like required, minLength, maxLength, email, etc., but you can also define custom validators.
   - Validation ensures that users enter the correct format of data before submitting the form.
4. **Handling User Input**:
   - As users interact with the form, you need to handle their input. You can do this by listening to events like onChanged, onSubmitted, onTap, etc., depending on the input widget.
   - Update the state of the form based on user input to reflect changes in the UI.
5. **Submission**:
   - After users fill out the form correctly, they usually submit it to perform some action (e.g., saving data, sending a request to a server, etc.).
   - You can use buttons like RaisedButton, ElevatedButton, or even InkWell to handle form submission.
6. **Layout**:
   - Proper layout and organization of form elements enhance usability. You can use Flutter's layout widgets like Column, Row, ListView, etc., along with containers like Padding, SizedBox, etc., to structure your form.
   - Group related fields together, use labels and hints effectively, and consider the accessibility aspect of your form layout.
7. **Accessibility**:
   - Make your form accessible to all users, including those with disabilities. Use semantic labels, provide textual alternatives to non-text elements, ensure proper tab navigation, etc.
   - Accessibility features not only improve user experience but also help your app comply with accessibility standards and regulations.

CODE:

LoginScreen

```
import 'package:emart_app/consts/consts.dart';
import 'package:emart_app/consts/lists.dart';
import 'package:emart_app/views/auth_screen/signup_screen.dart';
import 'package:emart_app/widgets_common/applogo_widget.dart';
import 'package:emart_app/widgets_common/bg_widget.dart';
import 'package:emart_app/widgets_common/custom_textfield.dart';
import 'package:emart_app/widgets_common/our_button.dart';
import 'package:get/get.dart';
class LoginScreen extends StatelessWidget
  { const LoginScreen({super.key});

  @override
  Widget build(BuildContext context)
    { return bgWidget(
       child:Scaffold( resizeToAvoidBottomInse
         t: false,
     body:
       Center( child:
       Column( childre
       n: [
           (context.screenHeight*0.1).heightBox,
           applogoWidget(),
           10.heightBox,
           "Log in to $appname".text.fontFamily(bold).white.size(18).make(),
           15.heightBox,
          Column( child
          ren: [
              customTextField(hint:emailHint,title:email),
             customTextField(hint:passwordHint,title:password),
             Align(
               alignment: Alignment.centerRight,

             child: TextButton(onPressed: () {}, child: forgetPass.text.make())),
              5.heightBox,
              //ourButton().box.width(context.screenWidth -50).make(),
              ourButton(color: buttonColor,title: login,textColor: whiteColor,onPress:
(){}).box.width(context.screenWidth -50).make(),
              5.heightBox,
              createNewAccount.text.color(fontGrey).make(),
              5.heightBox,
              ourButton(color: bodyColor2,title: signup,textColor: whiteColor,onPress:
```

```
(){
                Get.to(()=>SignupScreen());
              }).box.width(context.screenWidth -50).make(),
              10.heightBox,
              loginWith.text.color(fontGrey).make(),
              5.heightBox,
              Row(
                mainAxisAlignment: MainAxisAlignment.center,
                children: List.generate(3, (index) =>
                    Padding(
                      padding: const EdgeInsets.all(8.0),
                      child: CircleAvatar(
                                      backgroundColor: lightGrey,
                                      radius: 25,
                                      child: Image.asset(socialIconList[index],
                                      width: 30,),
                                    ),
                    )),
              )
            ],
          ).box.white.rounded.padding(const EdgeInsets.all(16)).
          width(context.screenWidth -70).
              shadowSm.
          make(),

          ],
        ),
      ),
    ));
  }
}
```

Sign up Screen

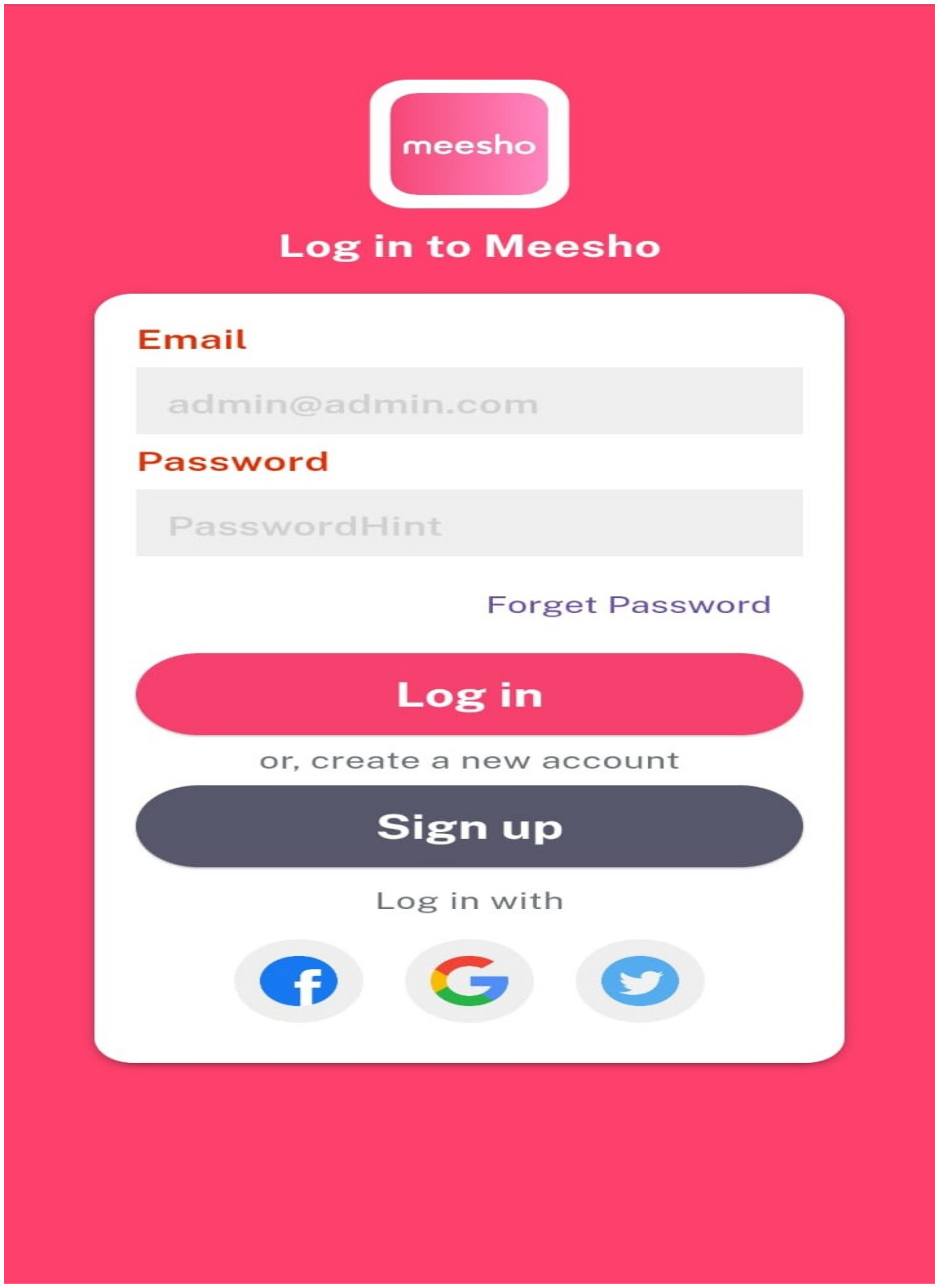
```
import 'package:emart_app/consts/consts.dart';
import 'package:emart_app/consts/lists.dart';
import 'package:emart_app/widgets_common/applogo_widget.dart';
import 'package:emart_app/widgets_common/bg_widget.dart';
import 'package:emart_app/widgets_common/custom_textfield.dart';
import 'package:emart_app/widgets_common/our_button.dart';
import 'package:get/get.dart';


class SignupScreen extends StatelessWidget
  { const SignupScreen({super.key});

  @override
  @override
  Widget build(BuildContext context)
    { return bgWidget(
        child:Scaffold( resizeToAvoidBottom
          Inset: false, body: Center(
            child:
              Column( childr
              en: [
                (context.screenHeight*0.1).heightBox,
                applogoWidget(),
                10.heightBox,
                "Sign Up to $appname".text.fontFamily(bold).white.size(18).make(),
                15.heightBox,
                Column( chil
                dren: [
                    customTextField(hint:nameHint,title:name),
                    customTextField(hint:emailHint,title:email),
                    customTextField(hint:passwordHint,title:password),
                    customTextField(hint:passwordHint,title:confirmPassword),
                    Align(
                        alignment: Alignment.centerRight,

                        child: TextButton(onPressed: () {}, child: forgetPass.text.make())),

                    Row(
                      children:
                        [ Checkbo
                        x(
                          checkColor: buttonColor,
```

value: false,

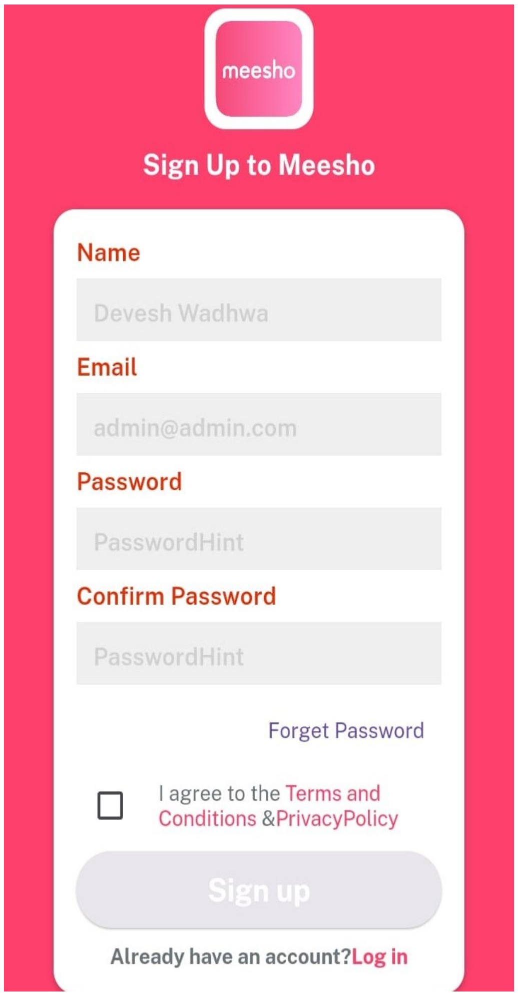
```
      onChanged: (newValue) {},
    ),
    10.widthBox,
    Expanded(
      child: RichText(text: const
        TextSpan( children: [
          TextSpan(
            text: "I agree to the ",
            style:
            TextStyle( fontFamily:
            bold, color: fontGrey,
            )
          ),
          TextSpan(
              text: termsAndCond,
              style:
              TextStyle( fontFamil
              y: bold, color:
              buttonColor,
              )
          ),

          TextSpan( tex
              t: " &",
              style:
                TextStyle( fontF
                amily: bold,
                color: fontGrey,
              )
          ),
          TextSpan(
              text: privacyPolicy,
              style:
              TextStyle( fontFamil
              y: bold, color:
              buttonColor,
              )
          ),
        ]
      )),
    )

  ],
),
5.heightBox,
//ourButton().box.width(context.screenWidth -50).make(),
ourButton(color: buttonColor,title: signup,textColor:
```

```
whiteColor,onPress: (){}).box.width(context.screenWidth -50).make(),
          10.heightBox,
          RichText(text:
          TextSpan( children: [
            TextSpan(
             text: alreadyHaveAccount,
             style: TextStyle(fontFamily: bold,color: fontGrey),

            ),
            TextSpan( te
             xt: login,
             style: TextStyle(fontFamily: bold,color:buttonColor),

            )
           ]
          ),
          ).onTap(()
           { Get.back
           ();
          }),



         ],
        ).box.white.rounded.padding(const EdgeInsets.all(16)).
        width(context.screenWidth -70).
        shadowSm.
        make(),

       ],
      ),
     ),
    ));
  }
}
```

OUTPUT: Device**(Emulator)**



Login Screen


Sign Up to Meesho

Name

Devesh Wadhwa

Email

admin@admin.com

Password

PasswordHint

Confirm Password

PasswordHint

Forget Password

☐ I agree to the Terms and Conditions &PrivacyPolicy

Sign Up screen

Conclusion: By understanding and implementing these theoretical concepts, you can create robust and user-friendly forms in Flutter that provide a seamless experience for your users.