

Experiment - 5

Aim : To apply navigation, routing and gestures in Flutter App

Theory :

In the provided code for the `LoginScreen`, there are implementations related to navigation, routing, and gestures:

1. Navigation and Routing:

- The `LoginScreen` is a StatefulWidget that defines the UI for the login page.
- Navigation between screens is achieved using the `Navigator` class from the `flutter/material.dart` library.
- After successful login, the user is navigated to the `HomeScreen` using the `Navigator.push` method:

```
Navigator.push(context, MaterialPageRoute(builder: (context) => HomeScreen()));
```

This code pushes the `HomeScreen` onto the navigation stack, replacing the current screen.

- The `TextButton` in the "Already Have an Account?" section has an `onPressed` handler that navigates to the `SignUpPage`:

```
TextButton(  
  onPressed: () {  
    Navigator.push(context, MaterialPageRoute(builder: (context) => SignUpPage()));  
  },  
  child: Text("Sign Up", style: TextStyle(fontSize: 20, fontWeight: FontWeight.w300,  
color: Colors.yellowAccent)),  
)
```

This code navigates to the `SignUpPage` when the "Sign Up" text is pressed.

2. Gestures:

- Gestures are implemented for the "Login" button using the `CustomButton` widget. The `onPressed` callback of `CustomButton` is linked to the `login` method:

```
CustomButton(text: "Login", onPressed: () {  
  login(emailController.text.toString(), passwordController.text.toString());  
})
```

The `login` method is called when the "Login" button is pressed, and it handles the login process.

3. TextFields and UIhelper:

- The code uses `CustomTextField` from `UIhelper` to create text input fields for email and password. The `CustomTextField` widget abstracts the common structure for text input fields.

```
    Ulhelper.CustomTextField(emailController, "Email", Icons.mail, false),  
    Ulhelper.CustomTextField(passwordController, "Password", Icons.password, true),
```

These fields are created using the `CustomTextField` method, providing the necessary parameters such as the controller, hint text, icon, and whether it is a password field.

Code :

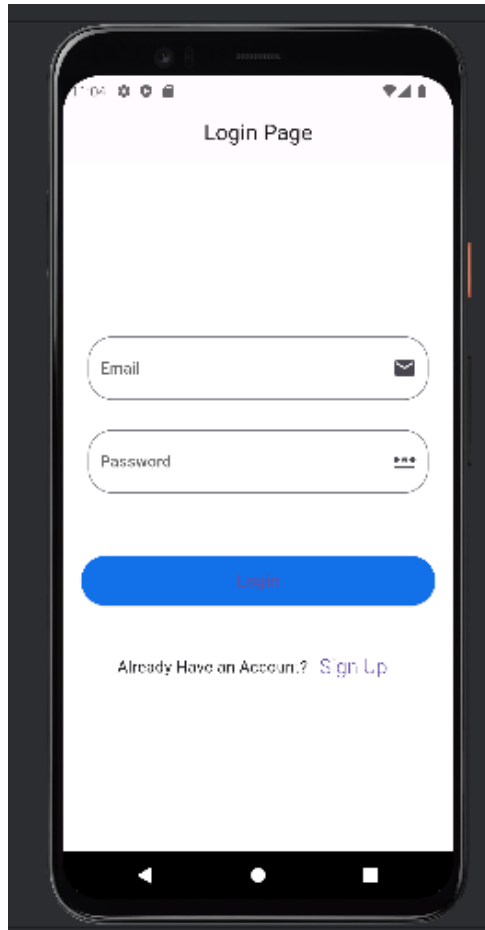
//login page

```
import 'package:firebase_auth/firebase_auth.dart';  
import 'package:flutter/material.dart';  
import 'package:gmeet_clone/Utils/Ulhelper.dart';  
import 'package:gmeet_clone/screens/Home_screen.dart';  
import 'package:gmeet_clone/screens/signup_screen.dart';  
import 'package:gmeet_clone/widgets/custom_button.dart';  
  
class LoginScreen extends StatefulWidget {  
  const LoginScreen({Key? key}) : super(key: key);  
  
  @override  
  State<LoginScreen> createState() => _LoginScreenState();  
}  
  
class _LoginScreenState extends State<LoginScreen> {  
  TextEditingController emailController=TextEditingController();  
  TextEditingController passwordController=TextEditingController();  
  
  login(String email, String password)async{  
    if(email=="" && password==""){  
      return Ulhelper.CustomAlertBox(context, "Enter Required Fields");  
    } else {  
      UserCredential? usercredential;  
      try{  
        usercredential=await FirebaseAuth.instance.signInWithEmailAndPassword(email: email,  
password: password).then((value){  
          Navigator.push(context, MaterialPageRoute(builder: (context)=>HomeScreen()));  
        });  
      } on FirebaseAuthException catch(ex) {  
        return Ulhelper.CustomAlertBox(context, ex.code.toString());  
      }  
    }  
  }  
}
```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text("Login Page"),
      centerTitle: true,
    ),
    body: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        UIhelper.CustomTextField(emailController, "Email", Icons.mail, false),
        UIhelper.CustomTextField(passwordController, "Password", Icons.password, true),
        const SizedBox(height: 30),
        CustomButton(text: "Login", onPressed: () {
          login(emailController.text.toString(), passwordController.text.toString());
        }),
        const SizedBox(height: 20),
        Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text("Already Have an Account?", style: TextStyle(fontSize: 16,color: Colors.white),),
            TextButton(onPressed: (){
              Navigator.push(context, MaterialPageRoute(builder: (context)=>SignUpPage()));
            }, child: Text("Sign Up",style: TextStyle(fontSize: 20, fontWeight:
FontWeight.w300,color: Colors.yellowAccent )))
          ],
        ),
      ],
    ),
  );
}

```



Conclusion : In summary, the code utilizes the `Navigator` class for navigation and routing between different screens (Login, SignUp, Home). Gestures are used for handling button presses, and `UIhelper` provides common UI elements like text input fields with consistent styling.