MAD and PWA Lab Mini Project

Name: Shubham Nakashe Class:D15A Roll no: 38

Aim: To create an interactive Form using form widget

Theory:

In Flutter, a Form widget is used to manage a group of form fields and handle form submission. It provides methods to validate and save form field values.

CODE:

```
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'package:twitch_clone/models/user.dart';
import 'package:twitch_clone/providers/user_provider.dart';
import 'package:twitch_clone/resources/auth_methods.dart';
import 'package:twitch_clone/responsive/responsive.dart';
import 'package:twitch_clone/screens/home_screen.dart';
import 'package:twitch_clone/screens/profile.dart';
import 'package:twitch_clone/widgets/custom_button.dart';
import 'package:twitch_clone/widgets/custom_text_field.dart';
import 'package:twitch_clone/widgets/loading_indicator.dart';
class SignupScreen extends StatefulWidget {
 static const String routeName = '/signup';
 const SignupScreen({super.key});
 @override
 State<SignupScreen> createState() => _SignupScreenState();
class _SignupScreenState extends State<SignupScreen> {
 final TextEditingController _emailController = TextEditingController();
 final TextEditingController _passwordController = TextEditingController();
 final TextEditingController _usernameController = TextEditingController();
 final AuthMethods _authMethods = AuthMethods();
 bool _isLoading = false;
 void signUpUser() async {
  void signUpUser() async {
   setState(() {
    _isLoading = true;
   });
   bool res = await _authMethods.signUpUser(context, _emailController.text,
```

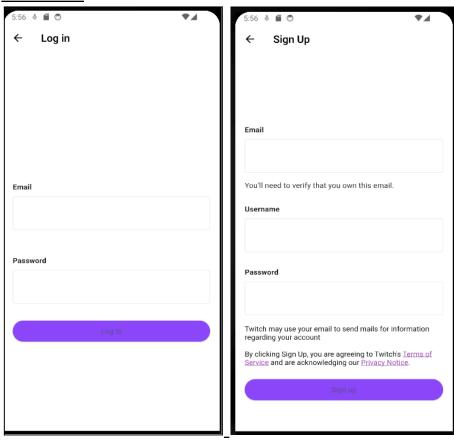
```
_usernameController.text, _passwordController.text);
  setState(() {
   _isLoading = false;
  });
  if (res) {
   final userProvider = Provider.of<UserProvider>(context, listen: false);
   userProvider.setUser(User(uid: ", username: _usernameController.text, email: "));
   Navigator.pushReplacementNamed(context,ProfileScreen.routeName);
 }
}
}
@override
void dispose() {
 _emailController.dispose();
 _passwordController.dispose();
 _usernameController.dispose();
 super.dispose();
}
@override
Widget build(BuildContext context) {
 final size = MediaQuery.of(context).size;
 return Scaffold(
  appBar: AppBar(
   title: const Text('Sign Up'),
  ),
  body: _isLoading
    ? const LoadingIndicator()
    : Responsive(
      child: SingleChildScrollView(
        child: Padding(
         padding: const EdgeInsets.symmetric(horizontal: 18.0),
         child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
           SizedBox(
            height: size.height * 0.1,
           const Text(
            "Email",
            style: TextStyle(
             fontWeight: FontWeight.bold,
            ),
           Padding(
```

```
child: CustomTextField(controller: _emailController),
             ),
             SizedBox(height: 6),
             Padding(
              padding: const EdgeInsets.only(bottom: 5.0),
              child: Text(
               'You\'ll need to verify that you own this email.',
               style: TextStyle(fontSize: 14),
              ),
             ),
             const SizedBox(
              height: 20,
             ),
             const Text(
              "Username",
              style: TextStyle(
               fontWeight: FontWeight.bold,
              ),
             ),
             Padding(
              padding: const EdgeInsets.symmetric(vertical: 8.0),
              child: CustomTextField(controller: _usernameController),
             ),
             const SizedBox(
              height: 20,
             ),
             const Text(
              "Password",
              style: TextStyle(
               fontWeight: FontWeight.bold,
              ),
             ),
             Padding(
              padding: const EdgeInsets.symmetric(vertical: 8.0),
              child: CustomTextField(controller: _passwordController),
             ),
             SizedBox(height: 10),
             RichText(
              text: const TextSpan(
               text:
                 'Twitch may use your email to send mails for information regarding your
account n n',
               style: TextStyle(color: Colors.black),
               children: [
                TextSpan(
                 text: 'By clicking Sign Up',
```

padding: const EdgeInsets.symmetric(vertical: 8.0),

```
style: TextStyle(color: Colors.black),
                ),
                TextSpan(
                 text: ', you are agreeing to Twitch\'s ',
                TextSpan(
                 text: 'Terms of Service',
                 style: TextStyle(
                   color: Color.fromARGB(255, 162, 72, 177),
                   decoration: TextDecoration.underline),
                ),
                TextSpan(
                 text: ' and are acknowledging our ',
                ),
                TextSpan(
                 text: 'Privacy Notice',
                 style: TextStyle(
                   color: Color.fromARGB(255, 162, 72, 177),
                   decoration: TextDecoration.underline),
                ),
                TextSpan(
                 text: '.',
                ),
               ],
             ),
            ),
            const SizedBox(
             height: 20,
            ),
            CustomButton(
             text: "Sign up",
             onTap: signUpUser,
            )
           ],
         ),
       ),
      ),
 );
}
```

OUTPUT:



LOGIN AND SIGNUP FORM