Atharva Mulam Batch B
D15A Roll No: 37

Exp No: 4

Aim: To create an interactive Form using form widget.

Theory:
In Flutter, forms are used to collect user input and interact with users. Flutter provides a set of widgets and classes to create forms efficiently.

1) **Form Widget:**

The Form widget is a container for form fields and manages the form's state. It's typically used as the root widget of a form and helps with validation and submission.
The Form widget doesn't display anything by itself but provides utilities to interact with its child form fields.

2) **FormField Widget:**

The FormField widget represents a single form field within a Form.
It's a generic class that's extended by various field-specific widgets like TextFormField, DropdownButtonFormField, CheckboxFormField, etc.
FormField widgets handle user input, validation, and error messages associated with the field.

3) **TextFormField:**

TextFormField is a commonly used form field widget for collecting text input from users.
It provides features like keyboard input, text editing, validation, error handling, and input formatting.
You can customize its appearance, input type, validator, controller, and more.

4) **Form Validation:**

Flutter provides built-in support for form validation, which ensures that user input meets specific criteria before submission.
You can define validation logic using validators like required, minLength, maxLength, email, numeric, etc., or create custom validators.
Form validation is typically performed within the validator parameter of form field widgets or by implementing the FormFieldValidator function.

5) Form Submission:

After validating user input, you can handle form submission using callbacks like onSaved or onFieldSubmitted provided by form field widgets.

Alternatively, you can use the FormState object to access and process the form data when the form is submitted.

Form submission involves processing the input data, performing additional actions, and updating the UI accordingly.

6) GlobalKey<FormState>:

To interact with a Form and its state, you typically use a GlobalKey<FormState> object.

The FormState object contains methods to validate, reset, and save form fields.

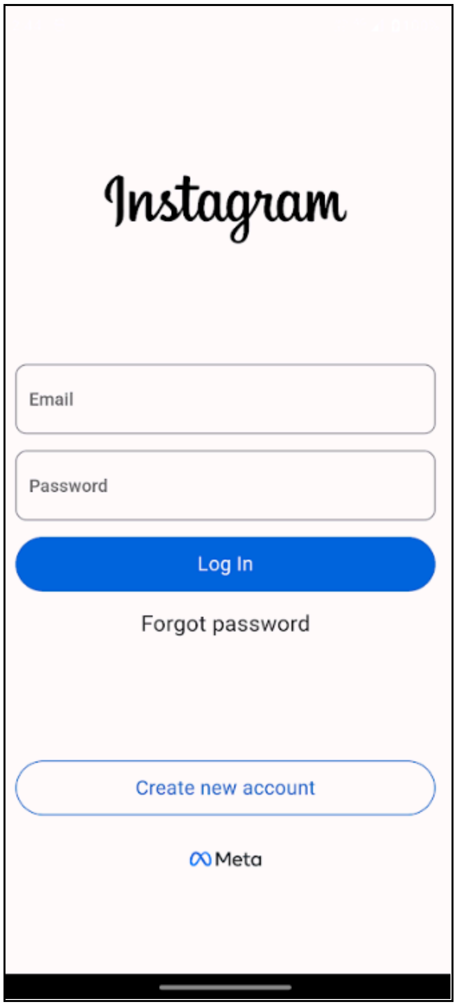
It's essential for accessing and manipulating the state of the form, especially when performing actions like validation and submission.

Code:

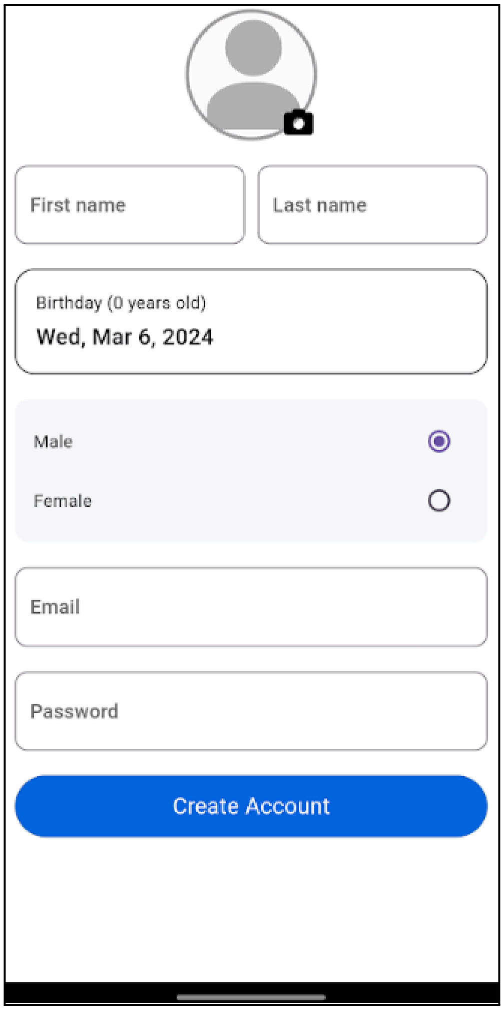
```
import 'package:flutter/material.dart';
import 'package:flutter_riverpod/flutter_riverpod.dart';
import 'package:ig/core/constants/constants.dart';
import 'package:ig/core/widgets/round_button.dart';
import 'package:ig/core/widgets/round_text_field.dart';
import 'package:ig/features/auth/presentation/screens/create_account_screen.dart';
import 'package:ig/features/auth/providers/auth_provider.dart';
import 'package:ig/features/auth/utils/utils.dart';

final _formKey = GlobalKey<FormState>();

class LoginScreen extends ConsumerStatefulWidget {
  const LoginScreen({super.key});

  @override
  ConsumerState<LoginScreen> createState() => _LoginScreenState();
  static const String _routeName = '/login';
  static String get route => _routeName;
}

class _LoginScreenState extends ConsumerState<LoginScreen> {
  late final TextEditingController _emailController;
  late final TextEditingController _passwordController;
  bool isLoading = false;

  Future<void> login() async{
    if(_formKey.currentState!.validate()){
      setState(() => isLoading = true);
      ref.read(authProvider).signIn(email: _emailController.text,password:
_passwordController.text,);
      setState(() => isLoading = false);
    }
  }

  @override
  void initState(){
    _emailController = TextEditingController();
    _passwordController = TextEditingController();
  super.initState();
  }

  @override
  void dispose(){
    _emailController.dispose();
    _passwordController.dispose();
    super.dispose();
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(),
      body: Padding(
        padding: Constants.defaultPadding,
        child: Column(
          mainAxisSize: MainAxisSize.max,
          mainAxisAlignment: MainAxisAlignment.spaceAround,
          children: [
            Image.asset('assets/icons/igword.png',
      width: 220,),
      Form(
       key: _formKey,
       child: Column(
        children: [
         RoundTextField(
          controller: _emailController,
          hintText: 'Email',
          keyboardType: TextInputType.emailAddress,
          textInputAction: TextInputAction.next,
          validator: validateEmail,
          ),
          const SizedBox(height: 15,),
          RoundTextField(
          controller: _passwordController,
          hintText: 'Password',
          textInputAction: TextInputAction.next,
          keyboardType: TextInputType.visiblePassword,
          validator: validatePassword,
          isPassword: true,
          ),
          const SizedBox(height: 15,),
          RoundButton(onPressed: login, label: 'Log In'),
          const SizedBox(height: 15,),
          const Text('Forgot password',style: TextStyle(fontSize: 20),),
        ]),
      ),
      Column(
       children: [
         RoundButton(
          onPressed: (){

Navigator.of(context).pushNamed(CreateAccountScreen.routeName);
          },
          label: 'Create new account',
          color: Colors.transparent,),
          Image.asset('assets/icons/meta-logo.png',height: 80,),
       ],
      )
     ],

    ) ),
  );
 }
}
```

Output:

Login



Sign Up



Birthday (0 years old)
Wed, Mar 6, 2024

Male ●
Female ○