Experiment 4

Name: Sonali Makhijani
Div: D15A
Roll no: 34

Aim: To create an interactive form using the form widget
Theory:

1. Form Widget:
   - The Form widget is a container used to group multiple form fields together.
   - It helps manage the state of the form, including validation, submission, and resetting.
   - The Form widget maintains a FormState object that holds the current state of the form fields.
   - Form widgets facilitate form submission, validation, and error handling.

2. FormField Widget:
   - A FormField widget represents a single form field within a Form.
   - Flutter provides various subclasses of the FormField widget for different types of input fields, such as TextFormField, CheckboxFormField, RadioFormField, DropdownButtonFormField, etc.
   - Each form field widget encapsulates the logic for validating user input and managing its state.
   - Form fields can be customized with properties to specify validation rules, error messages, initial values, input formatting, and more.

3. Validation:
   - Flutter's form widgets include built-in support for validation to ensure that user input meets specific criteria.
   - Form fields can be configured with validation functions or validators to check the correctness of user input.
   - Validators can be synchronous or asynchronous functions that return error messages if the input is invalid.

4. Submission:
   - The Form widget provides a mechanism to submit the form data once it's been filled out by the user.
   - Developers can define an onSaved callback for each form field to specify how the field's value should be processed when the form is submitted.
   - When the form is submitted, the onSaved callbacks for all form fields are invoked, allowing developers to collect, process, and submit the form data to a backend server or perform other actions.

signup_page.dart

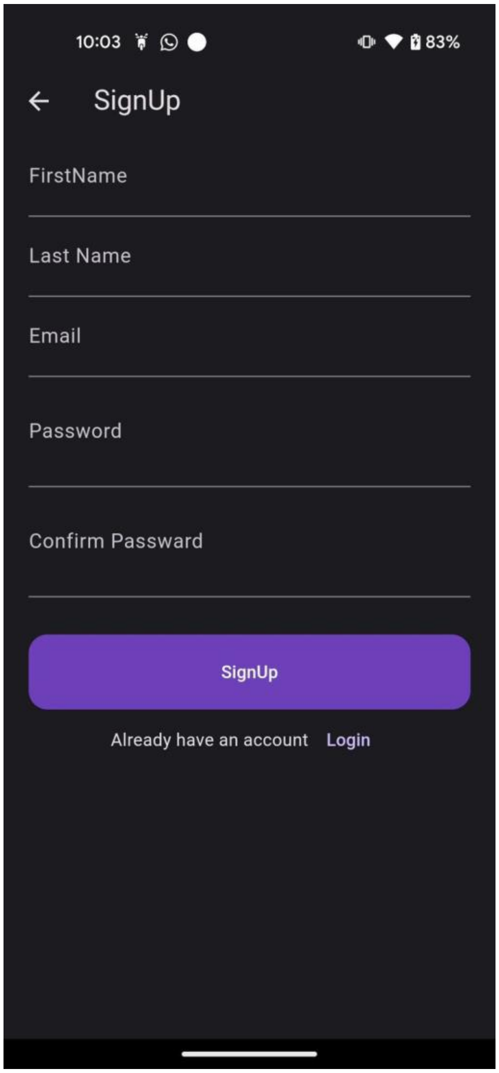
```
import 'package:flutter/material.dart';
import 'package:flutter_bloc/flutter_bloc.dart';

import '/data/models/user_model.dart';
import '/domain/bloc/auth_bloc/auth_bloc.dart';
import '/core/utils/input_validation.dart';
import '/presentation/screens/splash/splash.dart';
import '/presentation/widgets/loader.dart';

part 'components/signup_text_fields.dart';

class Signup extends StatefulWidget {
  const Signup({super.key});

  @override
  State<Signup> createState() => _SignupState();
}

class _SignupState extends State<Signup> {
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  final TextEditingController _confirmPasswordController =
      TextEditingController();
  final TextEditingController _firstNameController = TextEditingController();
  final TextEditingController _lastNameController = TextEditingController();
  final _formKey = GlobalKey<FormState>();

  @override
  Widget build(BuildContext context) {
    final colorScheme = Theme.of(context).colorScheme;
    return Scaffold(
      appBar: AppBar(
        title: const Text('SignUp'),
        backgroundColor: colorScheme.background,
      ),
      body: BlocConsumer<AuthBloc, AuthState>(
        listener: (context, state) {
          if (state is AuthStateCreatingUserSuccess) {
            Navigator.pop(context);
            Navigator.push(
              context,
              MaterialPageRoute(
                builder: (context) => const SplashScreen(),
              ),
            );
          }
        },
        builder: (context, state) {
          if (state is AuthStateCreatingUser) {
            return const Loader();
          } else {
            return ListView(
              padding: const EdgeInsets.symmetric(horizontal: 20),
              children: [
                SignupTextFields(
                  emailController: _emailController,
                  passwordController: _passwordController,
                  confirmPasswordController: _confirmPasswordController,
                  firstNameController: _firstNameController,
                  lastNameController: _lastNameController,
                  formKey: _formKey,
                ),
                const SizedBox(height: 30),
                ElevatedButton(
                  onPressed: () {
                    FocusScope.of(context).unfocus();
                    if (_formKey.currentState?.validate() ?? false) {
                      final UserModel user = UserModel()
                        ..email = _emailController.text
                        ..password = _passwordController.text
                        ..firstName = _firstNameController.text
                        ..lastName = _lastNameController.text;
                      context
                          .read<AuthBloc>()
                          .add(AuthEventCreateNewUser(user));
                    }
                  },
                  child: const Text('SignUp'),
                ),
                Row(
                  mainAxisAlignment: MainAxisAlignment.center,
                  children: [
                    const Text('Already have an account'),
                    TextButton(
                      onPressed: () => Navigator.pop(context),
                      child: const Text('Login'),
                    ),
                  ],
                )
              ],
            );
          }
        },
      ),
    );
  }
}
```

login_page.dart

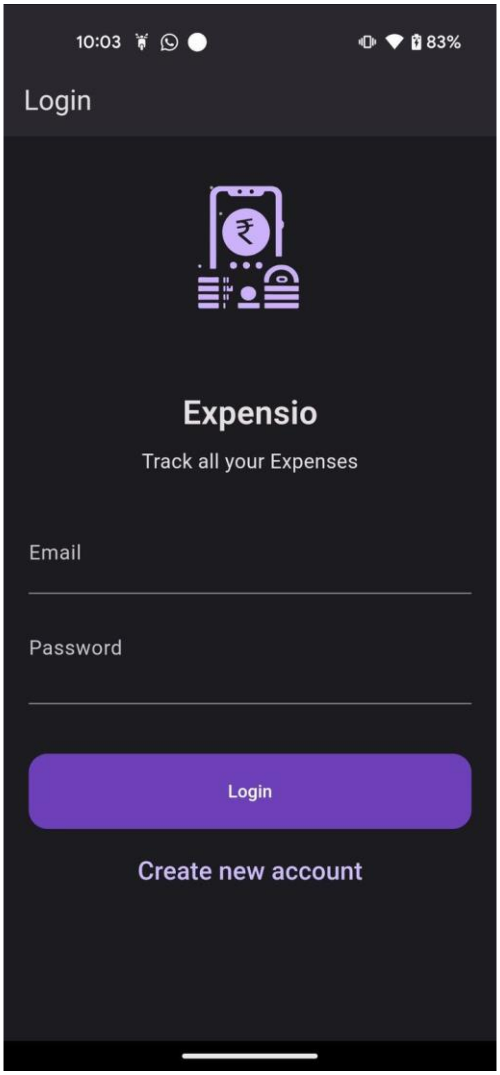
```
import 'package:flutter/material.dart';
import 'package:flutter_bloc/flutter_bloc.dart';
import 'package:lottie/lottie.dart';

import '/domain/bloc/auth_bloc/auth_bloc.dart';
import '/presentation/widgets/loader.dart';
import '/presentation/screens/authentication/signup.dart';
import '/presentation/screens/homescreen/homescreen.dart';
import '/core/constants/app_assets.dart';
import '/core/utils/input_validation.dart';
```

```
part 'components/login_icon_texts.dart';
part 'components/login_bottom_buttons.dart';

class Login extends StatefulWidget {
  const Login({super.key});

  @override
  State<Login> createState() => _LoginState();
}

class _LoginState extends State<Login> {
  final TextEditingController _emailController = TextEditingController();
  final TextEditingController _passwordController = TextEditingController();
  final _formKey = GlobalKey<FormState>();
  bool _isPasswordVisible = true;

  @override
  Widget build(BuildContext context) {
    final colorScheme = Theme.of(context).colorScheme;
    return BlocConsumer<AuthBloc, AuthState>(
      listener: (context, state) {
        if (state is AuthStateLoginSuccess) {
          Navigator.pushReplacement(
            context,
            MaterialPageRoute(
              builder: (context) => const HomeScreen(),
            ),
          );
        }
      },
      builder: (context, state) {
        return Scaffold(
          appBar: AppBar(
            title: const Text("Login"),
            backgroundColor: colorScheme.background,
          ),
          body: state is AuthStateLoggingIn
              ? const Loader()
              : GestureDetector(
                  onTap: () => FocusScope.of(context).unfocus(),
                  child: Form(
                    key: _formKey,
                    child: ListView(
                      physics: const BouncingScrollPhysics(),
                      padding: const EdgeInsets.symmetric(horizontal: 20),
                      children: [
                        const LoginIconAndTexts(),
                        TextFormField(
                          controller: _emailController,
                          validator: InputValidator.email,
                          decoration: const InputDecoration(
                            label: Text('Email'),
                            hintText: 'Enter your Email',
                          ),
                        ),
                          TextFormField(
                            controller: _passwordController,
                            obscureText: _isPasswordVisible,
                            validator: InputValidator.password,
                            decoration: InputDecoration(
                              suffix: IconButton(
                                icon: Icon(
                                  _isPasswordVisible
                                      ? Icons.visibility
                                      : Icons.visibility_off,
                                ),
                                onPressed: () => setState(
                                  () => _isPasswordVisible = !_isPasswordVisible,
                                ),
                              ),
                              label: const Text('Password'),
                              hintText: 'Enter your password',
                            ),
                          ),
                          LoginBottomButtons(
                            onLoginPressed: () {
                              FocusScope.of(context).unfocus();
                              if (_formKey.currentState!.validate()) {
                                context.read<AuthBloc>().add(
                                      AuthEventLoginWithEmailAndPassword(
                                        email: _emailController.text,
                                        password: _passwordController.text,
                                      ),
                                    );
                                _emailController.clear();
                                _passwordController.clear();
                              }
                            },
                            onCreateNewAccountPressed: () {
                              Navigator.push(
                                context,
                                MaterialPageRoute(
                                  builder: (_) => const Signup(),
                                ),
                              );
                            },
                          ),
                          const SizedBox(height: 20)
                        ],
                      ),
                    ),
                  ),
          );
        },
      );
    }
}
```

App UI:



Signup page



Login page

Widgets used: Form Widget, Form Widget Fields

Conclusion: Therefore understood the use of form widget in Flutter. Implemented signup and login page using form widget in my Flutter application.