

Name: Vedang Khandagale
Division: D15A
Roll No: 29
Batch: B

Experiment No 4

Aim: To create interactive form using form widget

Code:

```
import 'dart:io';
import 'package:facebook_clone/core/constants/app_colors.dart';
import 'package:facebook_clone/core/constants/constants.dart';
import 'package:facebook_clone/core/utils/utils.dart';
import 'package:facebook_clone/core/widgets/pick_image_widget.dart';
import 'package:facebook_clone/core/widgets/round_button.dart';
import 'package:facebook_clone/core/widgets/round_text_field.dart';
import package:facebook_clone/features/auth/presentation/widgets/birthday_picker.dart';
import package:facebook_clone/features/auth/presentation/widgets/gender_picker.dart';
import
'package:facebook_clone/features/auth/providers/auth_provider.dart';
import 'package:facebook_clone/features/auth/utils/utils.dart';
import 'package:flutter/material.dart';
import 'package:flutter_riverpod/flutter_riverpod.dart';
final _formKey = GlobalKey<FormState>();
class CreateAccountScreen extends ConsumerStatefulWidget {
  const CreateAccountScreen({super.key});
  static const routeName = '/create-account';
  @override
  ConsumerState<CreateAccountScreen> createState() =>
    _CreateAccountScreenState();
}
class _CreateAccountScreenState extends ConsumerState<CreateAccountScreen> {
  File? image;
  DateTime? birthday;
  String gender = 'male';
  bool isLoading = false;
  late final TextEditingController _fNameController;
  late final TextEditingController _lNameController;
```

```

late final TextEditingController _emailController;
late final TextEditingController _passwordController;
@override
void initState() {
  _fNameController = TextEditingController();
  _lNameController = TextEditingController();
  _emailController = TextEditingController();
  _passwordController = TextEditingController();
  super.initState();
}
@override
void dispose() {
  _fNameController.dispose();
  _lNameController.dispose();
  _emailController.dispose();
  _passwordController.dispose();
  super.dispose();
}
Future<void> createAccount() async {
  if (_formKey.currentState!.validate()) {
    _formKey.currentState!.save();
    setState(() => isLoading = true);
    await ref
      .read(authProvider)
      .createAccount(
        fullName: '${_fNameController.text} ${_lNameController.text}',
        birthday: birthday ?? DateTime.now(),
        gender: gender,
        email: _emailController.text,
        password: _passwordController.text,
        image: image,
      )
      .then((credential) {
        if (!credential!.user!.emailVerified) {
          Navigator.pop(context);
        }
      }).catchError((_) {
        setState(() => isLoading = false);
      });
    setState(() => isLoading = false);
  }
}

```

```

    }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: AppColors.realWhiteColor,
    appBar: AppBar(),
    body: SingleChildScrollView(
      child: Padding(
        padding: Constants.defaultPadding,
        child: Form(
          key: _formKey,
          child: Column(
            children: [
              GestureDetector(
                onTap: () async {
                  image = await pickImage();
                  setState(() {});
                },
                child: PickImageWidget(image: image),
              ),
              const SizedBox(height: 20),
              Row(
                children: [
                  // First Name Text Field
                  Expanded(
                    child: RoundTextField(
                      controller: _fNameController,
                      hintText: 'First name',
                      textInputAction: TextInputAction.next,
                      validator: validateName,
                    ),
                  ),
                  const SizedBox(width: 10),
                  // Last Name Text Field
                  Expanded(
                    child: RoundTextField(
                      controller: _lNameController,
                      hintText: 'Last name',

```

```

        textInputAction: TextInputAction.next,
        validator: validateName,
    ),
),
],
),
const SizedBox(height: 20),
BirthdayPicker(
    dateTime: birthday ?? DateTime.now(),
    onPressed: () async {
        birthday = await pickSimpleDate(
            context: context,
            date: birthday,
        );
        setState(() {});
    },
),
const SizedBox(height: 20),
GenderPicker(
    gender: gender,
    onChanged: (value) {
        gender = value ?? 'male';
        setState(() {});
    },
),
const SizedBox(height: 20),
// Phone number / email text field
RoundTextField(
    controller: _emailController,
    hintText: 'Email',
    textInputAction: TextInputAction.next,
    keyboardType: TextInputType.emailAddress,
    validator: validateEmail,
),
const SizedBox(height: 20),
// Password Text Field
RoundTextField(
    controller: _passwordController,
    hintText: 'Password',
    textInputAction: TextInputAction.done,

```

```

        keyboardType: TextInputType.visiblePassword,
        validator: validatePassword,
        isPassword: true,
      ),
      const SizedBox(height: 20),
      isLoading
        ? const Center(child: CircularProgressIndicator())
        : RoundButton(
            onPressed: createAccount,
            label: 'Create Account',
          ),
    ],
  ),
),
),
),
);
}

```

Output: