

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

Experiment No 5

Aim: To apply routing in Flutter Application

Theory: In Flutter, routing refers to the navigation system that allows users to move between different screens or pages within an app. Flutter uses a widget-based approach for navigation, where each screen is represented by a widget. The `Navigator` class manages the stack of routes and facilitates transitions between them.

Routes are typically defined using the `MaterialPageRoute` class, providing a seamless and platform-aware transition between screens. Developers can use the `Navigator` to push new routes onto the stack or pop existing routes off it. Named routes help in easily identifying and navigating to specific screens.

Flutter also supports route arguments, allowing developers to pass data between screens. Additionally, the `Navigator` provides a flexible set of transitions, such as slide, fade, or custom animations, enhancing the user experience during navigation.

Overall, Flutter's routing system provides a structured and intuitive way to handle navigation within mobile and web applications.

Code:

login_Screen.dart

```
import
'package:facebook_clone/features/auth/presentation/screens/create_account_screene.dart';
import
'package:facebook_clone/features/auth/providers/auth_provider.dart';
import 'package:flutter/material.dart';
import 'package:flutter_riverpod/flutter_riverpod.dart';

import '/core/constants/constants.dart';
import '/core/widgets/round_button.dart';
import '/core/widgets/round_text_field.dart';
import '/features/auth/utils/utils.dart';

final _formKey = GlobalKey<FormState>();
```

```

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

  @override
  ConsumerState<LoginScreen> createState() => _LoginScreenState();
}

class _LoginScreenState extends ConsumerState<LoginScreen> {
  late final TextEditingController _emailController;
  late final TextEditingController _passwordController;

  bool isLoading = false;

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

  @override
  void dispose() {
    _emailController.dispose();
    _passwordController.dispose();
    super.dispose();
  }

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

  @override

```

```

Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(),
    body: Padding(
      padding: Constants.defaultPadding,
      child: Column(
        mainAxisAlignment: MainAxisAlignment.spaceAround,
        children: [
          Image.asset(
            'assets/icons/fb_logo.png',
            width: 60,
          ),
          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',
                  keyboardType: TextInputType.visiblePassword,
                  textInputAction: TextInputAction.done,
                  isPassword: true,
                  validator: validatePassword,
                ),
                const SizedBox(height: 15),
                RoundButton(onPressed: login, label: 'Login'),
                const SizedBox(height: 15),
                const Text(
                  'Forget Password',
                  style: TextStyle(fontSize: 18),
                ),
              ],
            ),
          ),
        ],
      ),
    ),
  );
}

```

```

        ],
      ),
    ),
    Column(
      children: [
        RoundButton(
          onPressed: () {
            Navigator.of(context).pushNamed(
              CreateAccountScreen.routeName,
            );
          },
          label: 'Create new account',
          color: Colors.transparent,
        ),
        Image.asset(
          'assets/icons/meta.png',
          height: 50,
        ),
      ],
    ),
  ],
),
],
),
),
);
}
}

```

Create_account_screen.dart

```

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.d
art';
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;

  // controllers
  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,
                ),
              ),
            ],
          ),
        ],
      ),
    ),
  ),
)

```

```

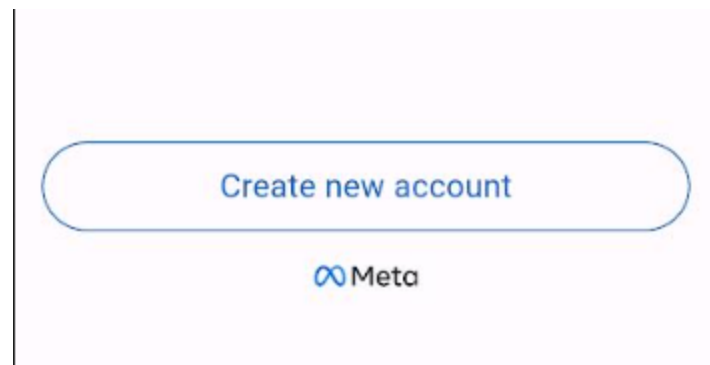
const SizedBox(height: 20),
BirthdayPicker(
  dateTime: birthday ?? DateTime.now(),
  onPressed: () async {
    birthday = await pickSimpleDate(
      context: context,
      date: birthday,
    );
    setState(() {});
  },
),
const SizedBox(height: 20),
GenderPicker(
  gender: gender,
  onChanged: (value) {
    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:





First name

Last name

Birthday (0 years old)

Wed, Jan 31, 2024

Male



Female



Email

Password

Create Account