

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

## Experiment No 3

Aim: To include images and fonts in flutter app

Theory: Certainly! Here's a simplified process for adding images in Flutter:

### 1. Import Libraries:

Ensure that you have the necessary libraries imported in your Dart file. For images, you'll typically use ``dart:ui`` and other relevant Flutter packages.

### 2. Adding Local Images:

- Place your local images in the ``assets`` folder.
- Declare the images in the ``pubspec.yaml`` file.

### 3. Adding Network Images:

- Use the ``Image.network`` widget for displaying images from the internet.

### 4. Image Widget:

- Create an ``Image`` widget and provide it with an ``ImageProvider``.
- Use ``AssetImage`` for local images and ``NetworkImage`` for network images.

### 5. ImageProvider:

- Understand that ``AssetImage`` and ``NetworkImage`` are subclasses of the ``ImageProvider`` class.
- You can create custom ``ImageProvider`` if needed.

### 6. CachedNetworkImage (Optional):

- If you want to cache network images, consider using the ``cached_network_image`` package.

### 7. Image Loading and Error Handling:

- Customize the loading and error behavior using ``loadingBuilder`` and ``errorBuilder`` properties of the ``Image`` widget or other relevant widgets.

Remember, the actual implementation details might vary based on your specific use case and the packages you choose to use. The key is to understand the concepts of working with local and network images and the various widgets and packages available in Flutter for handling images.

Code:

```
import
'package:facebook_clone/features/auth/presentation/screens/create_account_
screee.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.max,
        mainAxisSize: MainAxisSize.max,
        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,

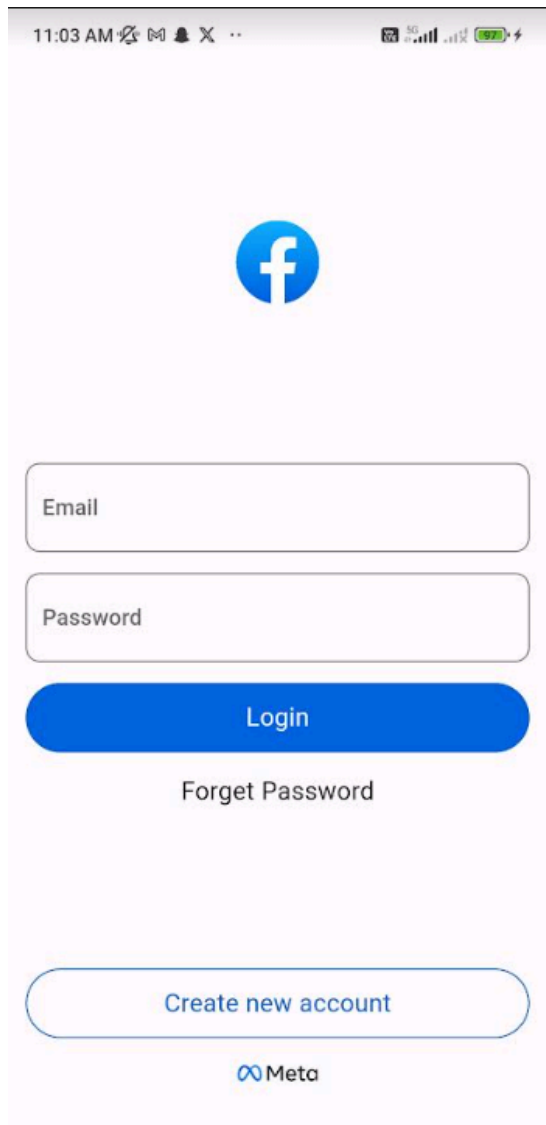
```

```

        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,
        ),
    ],
),
],
),
),
);
}
}

```

Output:



The image shows a mobile application interface for logging into Facebook. At the top, there is a status bar with the time 11:03 AM, various notification icons, and a battery level of 97%. Below the status bar is the Facebook logo, a blue circle with a white 'f'. Underneath the logo are two input fields: one labeled 'Email' and one labeled 'Password'. Below these fields is a blue button labeled 'Login'. Under the 'Login' button is a link labeled 'Forget Password'. At the bottom of the screen is a button labeled 'Create new account' and the Meta logo.

11:03 AM

97%

f

Email

Password

Login

Forget Password

Create new account

Meta