

Name: Varun Khubani

Roll no :30

Div:D15A

MPL Experiment 3

## 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:

```

class MyLogin extends StatefulWidget {
  const MyLogin({Key? key}) : super(key: key);

  @override
  State<MyLogin> createState() => _MyLoginState();
}

class _MyLoginState extends State<MyLogin> {
  TextEditingController emailTextEditingController = TextEditingController();
  TextEditingController passwordTextEditingController = TextEditingController();
  bool showProgressBar = false;
  // final FirebaseFirestore _firestore = FirebaseFirestore.instance;

  //function for login with email and password
  void login() async {
    String email = emailTextEditingController.text.trim();
    String password = passwordTextEditingController.text.trim();

    if (email == "" || password == "") {
      print('please fill all details');
    } else {
      try {
        UserCredential userCredential = await FirebaseAuth.instance
          .signInWithEmailAndPassword(email: email, password: password);

        if (userCredential != null) {
          Navigator.push(
            context,
            MaterialPageRoute(builder: (context) => NavigationPage()),
          );
        }
      } catch (e) {
        print(e);
      }
    }
  }
}

```

```

void login() async {
  } else {
    try {
      UserCredential userCredential = await FirebaseAuth.instance
        .signInWithEmailAndPassword(email: email, password: password);

      if (userCredential != null) {
        Navigator.push(
          context,
          MaterialPageRoute(builder: (context) => NavigationPage()),
        );
      }
    } on FirebaseAuthException catch (ex) {
      print(ex.code.toString());
    }
  }
}

@override
Widget build(BuildContext context) {
  return Container(
    decoration: const BoxDecoration(
      image: DecorationImage(
        image: AssetImage(
          "/Users/varunkhubani/Downloads/Kick-Bazaar-App/assets/login.png"),
        fit: BoxFit.cover)), // DecorationImage // BoxDecoration
    child: Scaffold(
      backgroundColor: Colors.transparent,
      body: Stack(

```

```

class _MyLoginState extends State<MyLogin> {
  Widget build(BuildContext context) {
    children: [
      Container(
        padding: const EdgeInsets.only(left: 35, top: 180),
        child: const Text(
          "Welcome Back!",
          style: TextStyle(
            color: Colors.white,
            fontSize: 33.0,
          ), // TextStyle
        ), // Text
      ), // Container
      SingleChildScrollView(
        child: Container(
          padding: EdgeInsets.only(
            top: MediaQuery.of(context).size.height * 0.5,
            left: 30.0,
            right: 30.0), // EdgeInsets.only
          child: Column(
            children: [
              TextField(
                controller: emailTextEditingController,
                decoration: InputDecoration(
                  fillColor: Colors.black12,
                  filled: true,
                  hintText: "Your Email",
                  hintStyle: const TextStyle(fontSize: 18.0),
                  border: OutlineInputBorder(
                    borderSide:

```

```
Widget build(BuildContext context) {  
  height: 30.0,  
), // SizedBox  
Row(  
  mainAxisAlignment: MainAxisAlignment.end,  
  children: [  
    // const Text(  
    //   "Sign In",  
    //   style: TextStyle(  
    //     fontSize: 25, fontWeight: FontWeight.w700),  
    // ),  
    CircleAvatar(  
      backgroundColor: Colors.black45,  
      radius: 25.0,  
      child: IconButton(  
        icon: const Icon(Icons.login),  
        color: Colors.white,  
        onPressed: () {  
          // Navigator.push(  
          //   context,  
          //   MaterialPageRoute(  
          //     builder: (context) => const NavigationPage()  
          login();  
        },  
      ), // IconButton  
    ), // CircleAvatar  
  ],  
), // Row
```

Output



