



# Login Screen Flutter Viva

## Overview

You're building a **Login UI** using Flutter's widget tree.

Flutter apps are made by **nesting widgets** — meaning widgets are children of other widgets. You use these to define UI components like buttons, textboxes, etc.



## `main()` function

```
void main() {  
  runApp(MaterialApp(  
    home: Demo(),  
    debugShowCheckedModeBanner: false,  
  ));  
}
```



## What is it doing?

- `main()` is the entry point of your Flutter app.
- `runApp()` launches your app and tells Flutter what widget to load first.
- `MaterialApp` is the **root widget** that gives your app Material Design look (buttons, text fields, etc.).
- `home: Demo()` loads the `Demo` class as the first screen.
- `debugShowCheckedModeBanner: false` removes the red **DEBUG** label.



## Who is parent and child?

- `runApp()` takes a child = `MaterialApp`
- `MaterialApp` has a child = `Demo()` (your main screen widget)



## `class Demo extends StatelessWidget`

This is your **screen widget**.

```
class Demo extends StatelessWidget {
```

- `StatelessWidget` means the UI won't change unless the entire widget is rebuilt.
- If we needed dynamic change (like form state), we'd use `StatefulWidget`.



## Controllers

```
final TextEditingController userNameController = TextEditingController();  
final TextEditingController passwordController = TextEditingController();
```

- These **controllers** are used to **read the input** entered in the TextFormFields.



## Scaffold

```
return Scaffold(  
  
```

- `Scaffold` gives you the basic page layout — app bar, body, floating buttons, etc.



## AppBar

```
appBar: AppBar(  
  title: Text("Login Page"),  
),
```

- `AppBar` is a **child** of `Scaffold`.
- Inside `AppBar`, `Text("Login Page")` is its **child** — the title shown at the top of the screen.



## Padding Widget

```
body: Padding(  
  padding: const EdgeInsets.all(16.0),  
  
```

```
child: Column(
```

- `Padding` adds space **around** its child widget.
- `child:` is a **Column**.
- `EdgeInsets.all(16.0)` gives 16 pixels space from all 4 sides.

## Column

```
child: Column(  
  children: [
```

- `Column` arranges widgets **vertically**, one below another.
- Its children are stored in a `children: []` list.
- All widgets inside the list are children of `Column`.

## TextFormField (Username)

```
TextFormField(  
  controller: userNameController,  
  decoration: InputDecoration(  
    labelText: "Enter Username",  
    prefixIcon: Icon(Icons.person),  
    border: OutlineInputBorder(),  
  ),  
)
```

## What it does?

- It's an **input field** for the username.
- `controller` connects this field with `userNameController` so we can access its text.
- `InputDecoration` is used to style the field.

## Widget Hierarchy

- `TextFormField` is a child of `Column`.

- Inside it, `InputDecoration` is not a widget, but a parameter used to decorate the `TextFormField`.

---

## TextFormField (Password)

Same as above but:

```
obscureText: true,
```

- This hides the input using dots for passwords.

---

## SizedBox

```
SizedBox(height: 16),
```

- Adds vertical space between widgets.
- It's a **child** of `Column`.

---

## ElevatedButton

```
ElevatedButton(  
  onPressed: () {  
    // logic  
  },  
  child: Text("Login"),  
),
```

### What it does?

- It's a clickable button.
- `onPressed` is the function that runs when the button is tapped.

### Widget Hierarchy

- It is a **child** of `Column`.
- Inside it, `Text("Login")` is the **child** of `ElevatedButton`.

## Logic Inside **onPressed**

```
String username = userNameController.text;
String password = passwordController.text;
```

- Get input text using the controller.
- Simple condition to check:

```
if (username == "admin" && password == "1234") {
  // show success
} else {
  // show failure
}
```

## **SnackBar**

```
ScaffoldMessenger.of(context).showSnackBar(
  SnackBar(content: Text("Login Successful")),
);
```

- Shows a small message popup at the bottom of the screen.
- `ScaffoldMessenger.of(context)` is required to show it within this screen.

## **Final Widget Tree Summary**

```
runApp
├─ MaterialApp
│   └─ Demo (StatelessWidget)
│       └─ Scaffold
│           ├── AppBar
│           │   └─ Text
│           └─ Padding
│               └─ Column
│                   ├── TextFormField (username)
│                   └─ SizedBox
```

```
└─ TextFormField (password)
└─ SizedBox
└─ ElevatedButton
  └─ Text
```

## How to Explain in Viva:

You can say:

"I used a Scaffold to give the page structure. Inside the body, I added a Padding widget so that the content doesn't touch the screen edge. Then I used a Column to arrange the username field, password field, and login button vertically. Each field is a TextFormField with a controller to access the typed data. The login button uses an ElevatedButton and shows a SnackBar if the login is successful or not. All widgets are kept basic without any external libraries or styling."

Let me know if you want:

- **Register** page
- **Forgot password?** text
- **Show/hide password** toggle
- Navigation to another page after login

All basic – no fancy Flutter packages needed.