

Introduction

Designing an intuitive and user-friendly Login and Sign-Up UI is crucial for any Flutter application. By creating a common structure, we can maintain consistency and reusability across both pages. Using widgets like `TextFormField` for user input, `ElevatedButton` for form submission, and layout elements such as `Column` and `Padding`, we can build a well-structured authentication interface.

1. Create a Common Structure

A common form structure for both the **Login** and **Sign-Up** pages using `TextFormField` widgets to capture user input and `ElevatedButton` for submitting the form.

2. Steps to Design Login & Sign-Up UI

2.1 Common Widgets

- `TextFormField`: To take user input like email and password.
- `ElevatedButton`: To submit the form.
- `Column`: To arrange widgets vertically.
- `Padding`: To add space around the widgets.

3. UI Elements Overview

3.1 `TextFormField` Widgets

- **Email**: A field for users to input their email.
- **Password**: A password input field where characters are hidden.

3.2 Validation

- The **Email** and **Password** fields have simple validation that checks if they are empty.
- The **Sign-Up** page has an additional password confirmation field and validates if both passwords match.

3.3 Buttons

- The **Login** and **Sign-Up** pages have a centered button that triggers form submission.

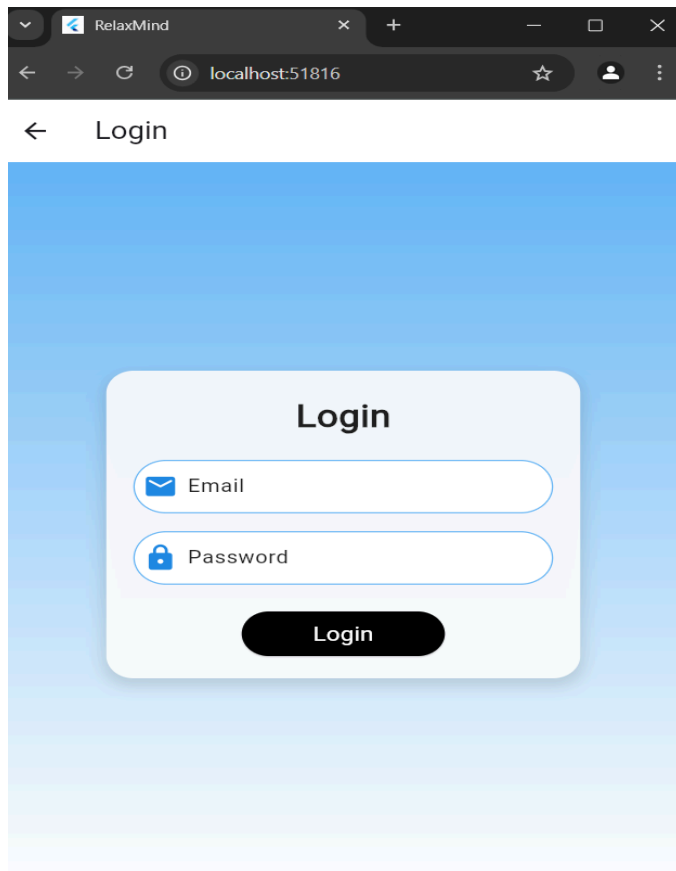
3.4 UI Improvements

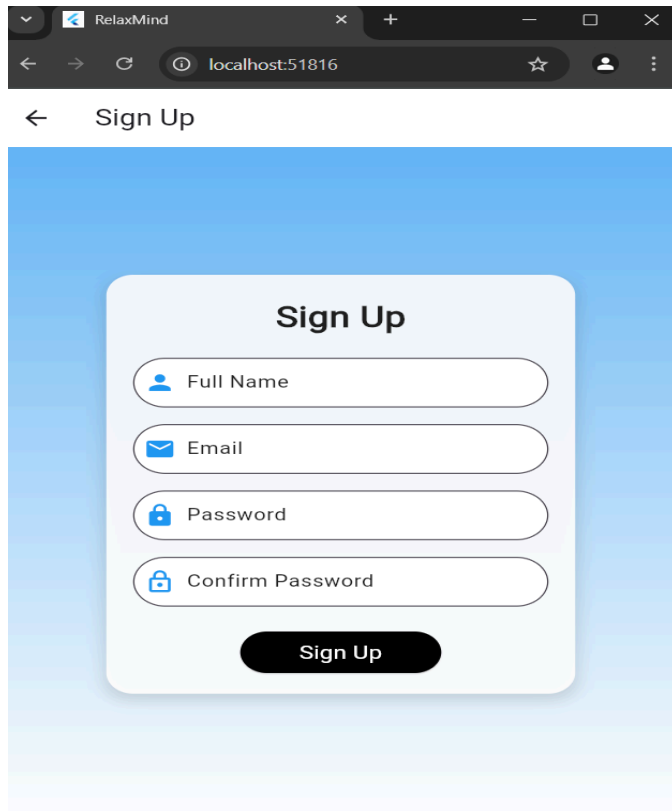
- **Icons**: Added icons to the input fields for a more intuitive UI.
- **Padding**: Used padding around the form elements for better spacing.

- **SizedBox**: Used to add space between form elements.

4. Navigation

To switch between the **Login** and **Sign-Up** pages used `Navigator.push()` for navigation.





Conclusion: A well-designed **Login and Sign-Up UI** enhances usability and improves user interaction in an application. By incorporating **form validation, proper UI spacing, icons for better accessibility, and navigation using `Navigator.push()`**, we create a seamless authentication process.