

Name:Komal Singh
Div:D15B
Roll No:56

MPL Practical 4

Aim:

To create an interactive form in a Flutter application using the **Form** widget, allowing user input validation and data submission.

Theory:

Forms are essential UI components used for collecting user input in applications. In Flutter, forms are built using the **Form** widget, which manages multiple input fields together.

Key Components of a Form in Flutter:

1. **Form Widget** – A container that holds form fields and a global key for validation.
2. **TextFormField Widget** – Used for user input, supporting validation and text handling.
3. **DropdownButtonFormField Widget** – Allows users to select values from a dropdown list.
4. **Checkbox, Radio, and Switch Widgets** – Used for boolean input selection.
5. **ElevatedButton** – Triggers form submission or reset actions.

Form Validation in Flutter:

- The **GlobalKey<FormState>** is used to manage form validation and state.
- The **validator** property of **TextFormField** ensures valid input by checking conditions like emptiness, email format, or password strength.
- The **ScaffoldMessenger** can be used to display messages based on form validation results.

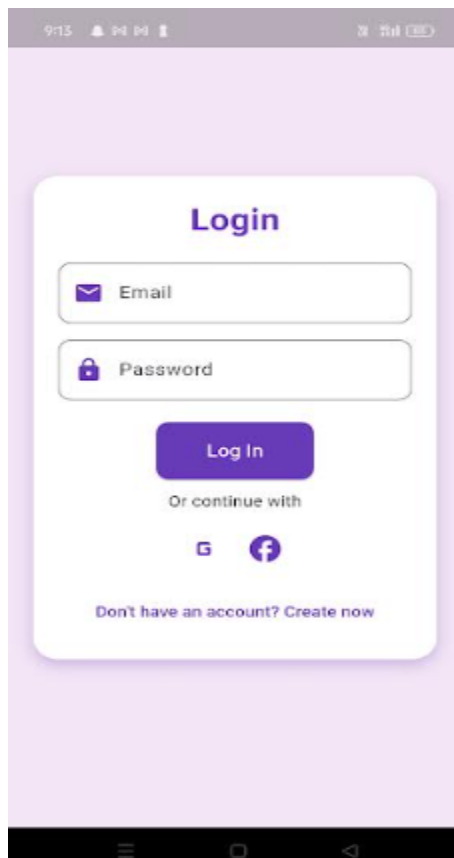
Using these components, Flutter enables efficient user input handling and validation in forms, ensuring a seamless experience.

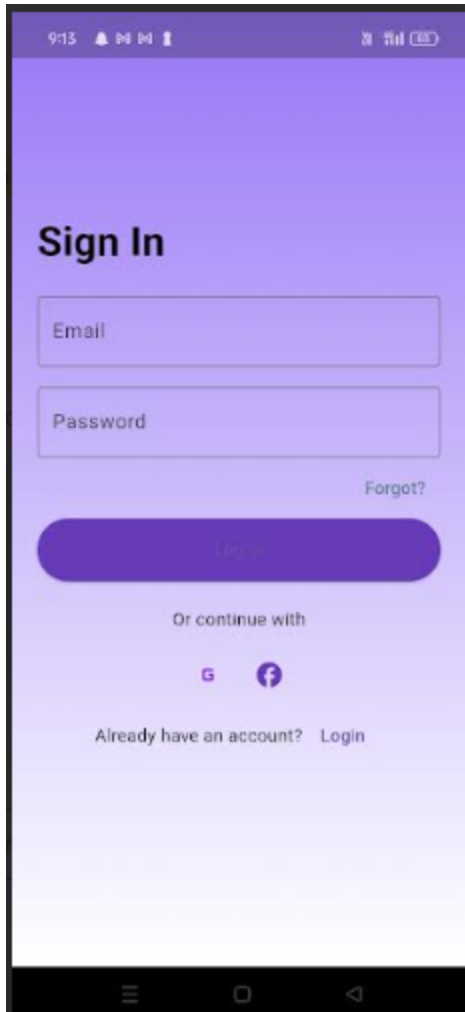
Steps to Create an Interactive Login/Signup Form in Flutter

1. **Create a Flutter Project**
 - Start a new Flutter project using **flutter create project_name**.
2. **Set Up the Form Widget**
 - Use the **Form** widget to manage user input.

- Assign a `GlobalKey<FormState>` to validate and save form data.
- 3. **Add Input Fields**
 - Use `TextFormField` for email and password fields.
 - Implement `validator` functions to check input validity.
- 4. **Handle User Input**
 - Use `TextEditingController` to capture user input.
 - Validate input when the user submits the form.
- 5. **Create Login and Signup Buttons**
 - Use `ElevatedButton` to trigger form submission.
 - Implement separate functions for login and signup logic.
- 6. **Navigation Between Screens**
 - Use `Navigator.push()` to move between login and signup screens.
- 7. **Integrate Authentication (Optional)**
 - Connect the form to Firebase or an API for user authentication.
- 8. **Run the App**
 - Use `flutter run` to test the login/signup functionality.

Output:





Conclusion:

Creating an interactive form using the **Form** widget in Flutter ensures proper user input handling, validation, and data submission. It helps build user-friendly applications by providing a structured way to collect and process user data efficiently.