

---

# Flutter Forms & Validation: Building Reliable Input Screens

**Subtitle:** Learn how to create input forms with validation to capture user data accurately.

**Website Name:** haas.dev

**Website Link:** <https://dev-roast-app.vercel.app>

---

## Introduction

Forms are essential in apps for **login, signup, or data collection**. Proper validation ensures **accurate user input** and improves UX. This guide teaches Flutter's Form widget, TextFormField, and validation techniques.

---

## Step 1: Create a Form

```
final _formKey = GlobalKey<FormState>();

Form(
  key: _formKey,
  child: Column(
    children: [
      TextFormField(
        decoration: InputDecoration(labelText: 'Email'),
      ),
      ElevatedButton(
        onPressed: () {},
        child: Text('Submit'),
      ),
    ],
  ),
)
```

**Exercise:** Create a simple form with an email field.

---

## Step 2: Add Validators

```
TextFormField(
  decoration: InputDecoration(labelText: 'Email'),
  validator: (value) {
```

```

    if (value == null || value.isEmpty) {
        return 'Please enter your email';
    } else if (!value.contains('@')) {
        return 'Enter a valid email';
    }
    return null;
},
)

```

**Exercise:** Add a password field with minimum 6 characters validation.

---

### Step 3: Submit Form

```

ElevatedButton(
  onPressed: () {
    if (_formKey.currentState!.validate()) {
      print("Form is valid");
    } else {
      print("Form is invalid");
    }
  },
  child: Text('Submit'),
)

```

**Exercise:** Print email and password if form is valid.

---

### Step 4: Save & Retrieve Form Values

```

String email = '';
String password = '';

TextFormField(
  onSave: (value) => email = value!,
)

```

Call `_formKey.currentState!.save()` after validation to save values.

**Exercise:** Save form values and display them in a dialog.

---

### Step 5: Real-Time Validation

```

TextFormField(
  autovalidateMode: AutovalidateMode.onUserInteraction
)

```

```
autovalidateMode: AutovalidateMode.onUserInteraction,  
validator: (value) {  
  return value!.isEmpty ? 'Cannot be empty' : null;  
},  
)
```

**Exercise:** Enable real-time validation for both email and password fields.

---

## Step 6: Mini Project

Build a **Signup Form App**:

- Email, password, and confirm password fields
  - Real-time validation for each field
  - Submit button prints values if valid
  - Show error messages inline
- 

## Key Takeaways

- Use `Form` and `TextFormField` for structured input
  - Always validate user input to prevent errors
  - `autovalidateMode` improves UX
  - Save input values using `onSaved`
  - Proper forms make your app **professional and reliable**
- 

Skipping form validation leads to **bugs and poor UX**, which is critical in real apps.

Visit **haas.dev** for more Flutter guides, tutorials, and complete project resources.

Website Name: haas.dev

Website Link: <https://dev-roast-app.vercel.app>

---