
*Flutter*

# Training Assignments

# Forms

## Overview

What you'll learn

- Build a form with validation.
- Create and style a text field.
- Retrieve the value of a text field.

## Tasks

1. Prepare

```dart
class Intro extends StatelessWidget {
  const Intro({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: buildHomePage(),
    );
  }

  Widget buildHomePage() {
    return Scaffold(
      appBar: AppBar(title: const Text('Introduction Forms'),),
      body: const MyCustomForm(),
    );  // Scaffold
  }
}

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

  @override
  State<MyCustomForm> createState() => _MyCustomFormState();
}

class _MyCustomFormState extends State<MyCustomForm> {
  @override
  Widget build(BuildContext context) {
    return Container();
  }
}
```

2. Build a form with validation

+ Update **_MyCustomFormState class** follow code

```dart
class _MyCustomFormState extends State<MyCustomForm> {
  final _formKey = GlobalKey<FormState>();

  @override
  Widget build(BuildContext context) {
    return Form(
      key: _formKey,
      child: Column(
        children: [
          TextFormField(
            validator: (value) {
              if (value == null || value.isEmpty) {
                return 'Please enter some text';
              }
              return null;
            },
          ),  // TextFormField
          ElevatedButton(
            onPressed: () {
              if (_formKey.currentState!.validate()) {
                ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
                  content: Text('Processing Data'),
                ));  // SnackBar
              }
            },
            child: const Text('Submit'))  // ElevatedButton
        ],
      ),  // Column
    );  // Form
  }
}
```

3. Create and style a text field

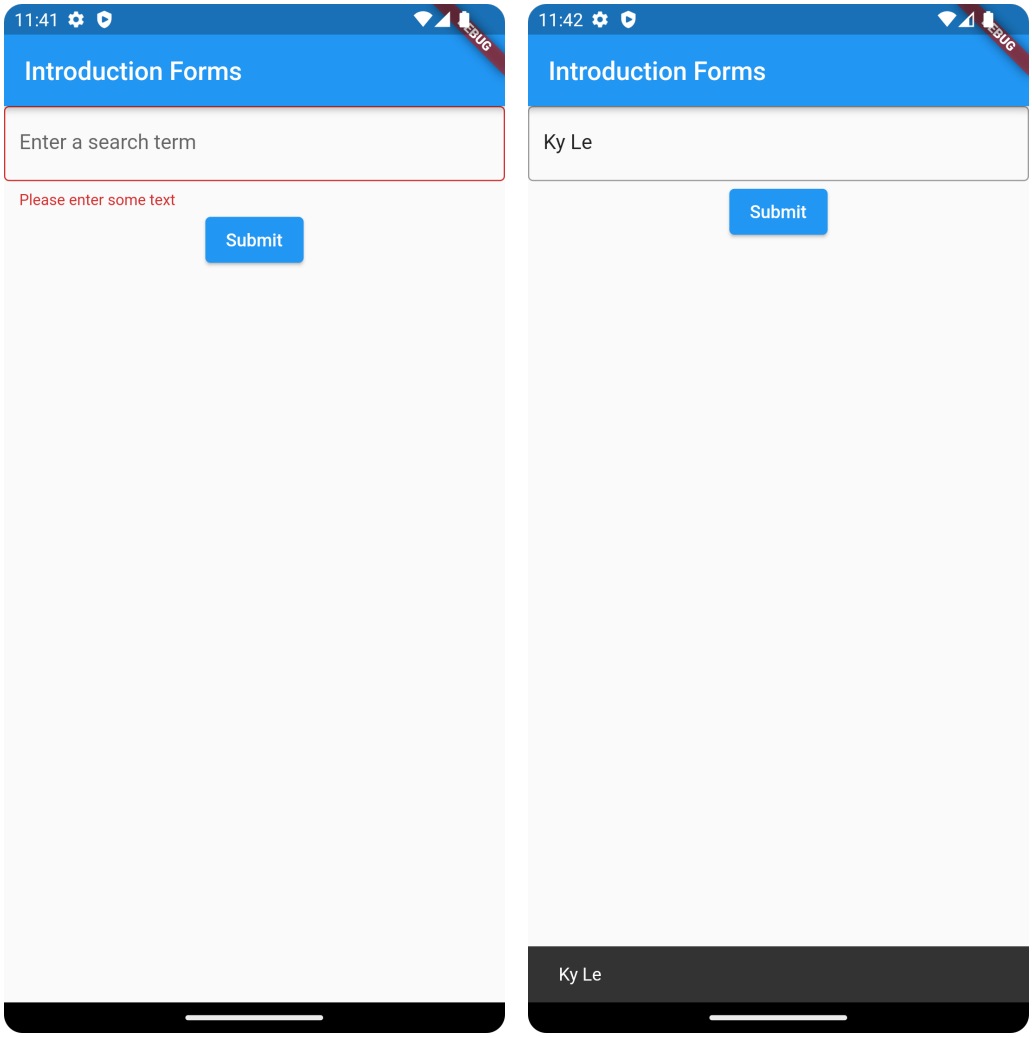
+ Update TextFormField widget

```dart
TextFormField(
  decoration: const InputDecoration(
    border: OutlineInputBorder(),
    hintText: 'Enter a search term',
  ),  // InputDecoration
  validator: (value) {
    if (value == null || value.isEmpty) {
      return 'Please enter some text';
    }
    return null;
  },
),  // TextFormField
```

4. Retrieve the value of a text field

```dart
class _MyCustomFormState extends State<MyCustomForm> {
  final _formKey = GlobalKey<FormState>();
  final searchTerm = TextEditingController();

  @override
  void dispose() {
    super.dispose();
    // Clean up the controller when the widget is disposed.
    searchTerm.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Form(
      key: _formKey,
      child: Column(
        children: [
          TextFormField(
            controller: searchTerm,
            decoration: const InputDecoration(
              border: OutlineInputBorder(),
              hintText: 'Enter a search term',
            ),  // InputDecoration
            validator: (value) {
              if (value == null || value.isEmpty) {
                return 'Please enter some text';
              }
              return null;
            },
          ),  // TextFormField
          ElevatedButton(
            onPressed: () {
              if (_formKey.currentState!.validate()) {
                ScaffoldMessenger.of(context).showSnackBar(SnackBar(
                  content: Text(searchTerm.text),
```

5. Run the app



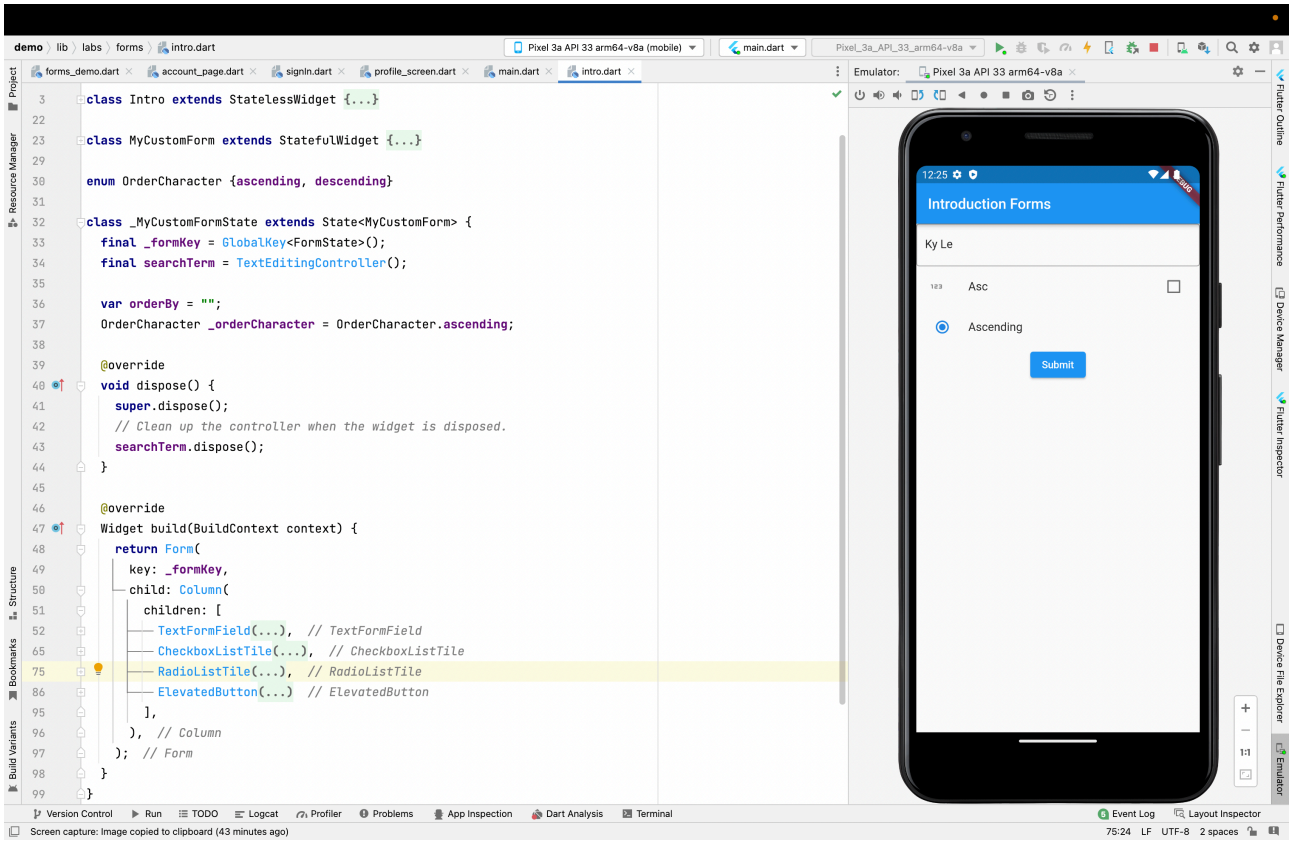

6. Add **Checkbox** into form

```dart
CheckboxListTile(
  title: const Text('Asc'),
  secondary: const Icon(Icons.onetwothree_rounded),
  value: false,
  onChanged: (value) {
    setState(() {

    });
  }
), // CheckboxListTile
```

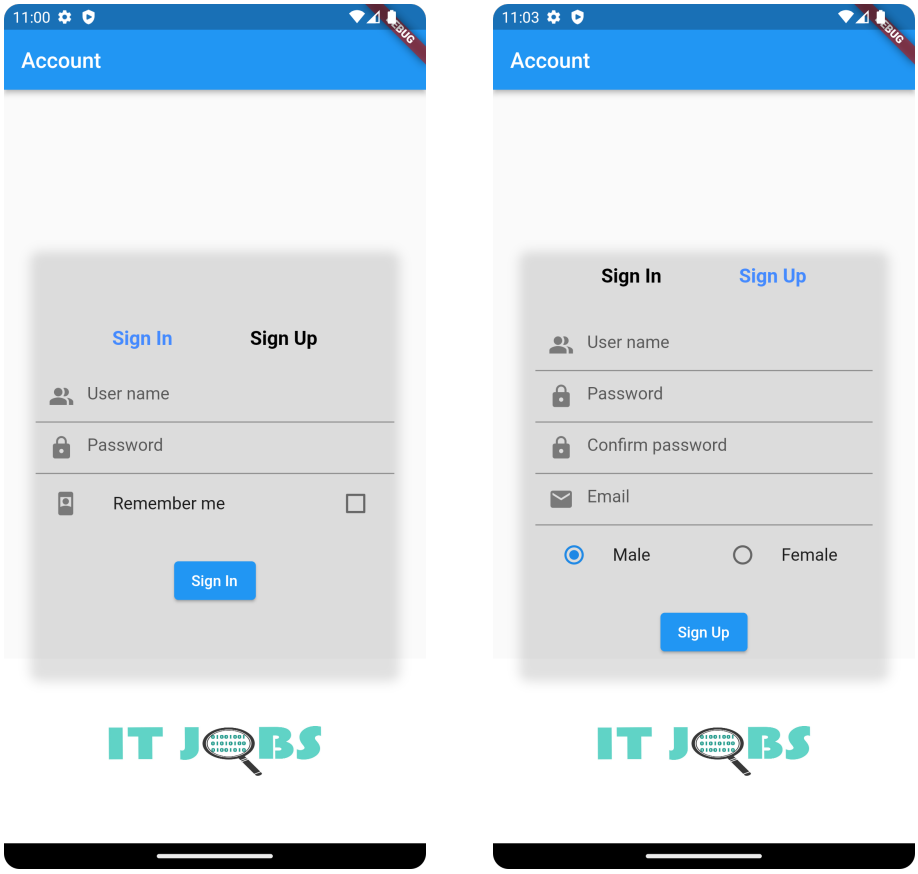
## 7. Add **Radio button** into form

```dart
enum OrderCharacter {ascending, descending}

class _MyCustomFormState extends State<MyCustomForm> {
  final _formKey = GlobalKey<FormState>();
  final searchTerm = TextEditingController();

  var orderBy = "";
  OrderCharacter _orderCharacter = OrderCharacter.ascending;

  @override
  void dispose() {
    super.dispose();
    // Clean up the controller when the widget is disposed.
    searchTerm.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return Form(
      key: _formKey,
      child: Column(
        children: [
          TextFormField(...),  // TextFormField
          CheckboxListTile(...),  // CheckboxListTile
          RadioListTile(
            title: const Text('Ascending'),
            value: OrderCharacter.ascending,
            groupValue: _orderCharacter,
            onChanged: (OrderCharacter? value) {
              setState(() {
                _orderCharacter = value!;
                orderBy = OrderCharacter.ascending.name;
              });
            }
          ),  // RadioListTile
```

Final code



## Extra tasks




Tips:

- Using Stack class: https://api.flutter.dev/flutter/widgets/Stack-class.html
- Using GestureDetector class
- Using Container class

```
child: Container(
  width: size.width * 0.85,
  height: size.height * 0.5,
  decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(10),
      boxShadow: const [
        BoxShadow(color: Colors.black12, blurRadius: 8, spreadRadius: 5)
      ]),  // BoxDecoration
```

--THE END—