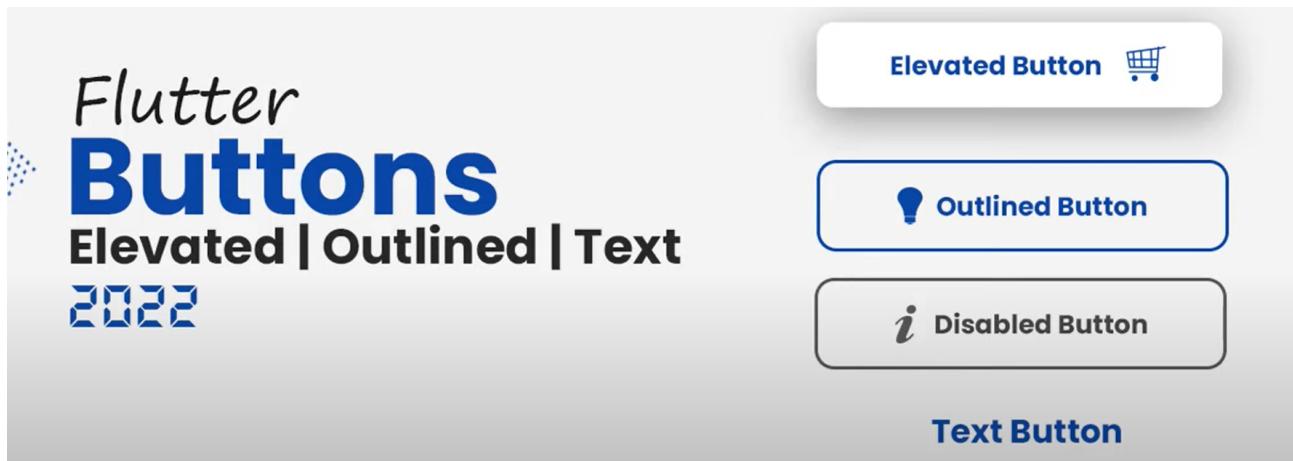


Flutter UI Design



<https://docs.flutter.dev/release/breaking-changes/buttons>

main.dart

A screenshot of Visual Studio Code showing the 'main.dart' file. The code defines a simple Flutter application:

```
1 import 'package:flutter/material.dart';
2 import 'Home.dart';
3
4 void main() {
5   runApp(const MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   const MyApp({Key? key}) : super(key: key);
10
11   // This widget is the root of your application.
12   @override
13   Widget build(BuildContext context) {
14     return MaterialApp(
15       debugShowCheckedModeBanner: false,
16       title: 'Flutter Demo',
17       themeMode: ThemeMode.system,
18       home: Home()); // MaterialApp
19   }
20 }
```

The 'lib' folder contains 'main.dart', 'Dashboard.dart', 'Home.dart', and 'main copy.dart'. The 'FLUTTER_ADDITEM' folder contains '.dart_tool', '.idea', 'android', 'build', 'ios', 'lib', 'main.dart', 'main copy.dart', 'test', 'web', and '.gitignore'. The 'OUTLINE' and 'DEPENDENCIES' sections are also visible.

Home.dart

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Tab Bar:** Dashboard.dart, Home.dart (active), main.dart, main copy.dart.
- Left Sidebar:** Includes icons for Explorer, Search, Problems, and others, with a notification count of 1.
- Code Editor:** Displays the `Home.dart` file content. The code defines a `Home` StatelessWidget that returns a `Scaffold` with a central `Column` containing three buttons: `Elevated`, `Outlined`, and `Text`.
- Preview Area:** On the right, there is a mobile phone preview showing the three buttons labeled "Elevated", "Outlined", and "Text".

```
1 import 'package:flutter/cupertino.dart';
2 import 'package:flutter/material.dart';
3
4 class Home extends StatelessWidget {
5   const Home({Key? key}) : super(key: key);
6
7   @override
8   Widget build(BuildContext context) {
9     return Scaffold(
10       body: Column(
11         children: [
12           Padding(
13             padding: const EdgeInsets.all(8.0),
14             child: Center(
15               child: Column(
16                 children: [
17                   ElevatedButton(
18                     onPressed: () {},
19                     child: const Text('Elevated'),
20                     style: ElevatedButton.styleFrom(
21                       fixedSize: Size(300, 80),
22                     ),
23                   ), // ElevatedButton
24                   OutlinedButton(
25                     onPressed: () {},
26                     child: const Text('Outlined'),
27                     style: OutlinedButton.styleFrom(
28                       fixedSize: Size(300, 80),
29                     ),
30                   ), // OutlinedButton
31                   TextButton(
32                     onPressed: () {},
33                     child: const Text('Text'),
34                     style: TextButton.styleFrom(
35                       fixedSize: Size(300, 80),
36                     ),
37                   ), // TextButton
38                 ],
39               ), // Column
40             ), // Center
41           ), // Padding
42           ],
43         ), // Column
44       ); // Scaffold
45     }
46   }
47 }
```

Modify for Design ElevatedButton

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Explorer:** Shows the project structure with files like Dashboard.dart, Home.dart lib, main.dart lib, and main copy.dart lib.
- Open Editors:** Home.dart is the active editor.
- Code:** The code defines a StatelessWidget named Home. It contains a Scaffold with a Center body, which in turn contains an ElevatedButton with the text "Let's Begin".
- Preview:** An iPhone X emulator on the right shows the button with a blue background and white text.

```
lib > Home.dart > Home
1 import 'package:flutter/material.dart';
2
3 class Home extends StatelessWidget {
4     const Home({Key? key}) : super(key: key);
5
6     @override
7     Widget build(BuildContext context) {
8         return Scaffold(
9             body: Center(
10                 child: ElevatedButton(
11                     child: Text("Let's Begin"),
12                     onPressed: () {},
13                 ), // ElevatedButton
14             ), // Center
15         ); // Scaffold
16     }
17 }
18 }
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Bar:** Go, Run, Terminal, Help.
- Explorer:** Shows the project structure with files like Dashboard.dart, Home.dart, main.dart, and main copy.dart.
- Code:** The code defines a StatelessWidget named Home. It uses the Cupertino ElevatedButton style from the flutter/cupertino package. The button has a yellow background, black text, and a black border.
- Preview:** An iPhone X emulator on the right shows the button with a yellow background and black text.

```
lib > Home.dart > ...
1 import 'package:flutter/cupertino.dart';
2 import 'package:flutter/material.dart';
3
4 class Home extends StatelessWidget {
5     const Home({Key? key}) : super(key: key);
6
7     @override
8     Widget build(BuildContext context) {
9         return Scaffold(
10             body: Center(
11                 child: ElevatedButton(
12                     child: Text("Let's Begin"),
13                     onPressed: () {},
14                     style: ElevatedButton.styleFrom(
15                         padding: EdgeInsets.all(20.0),
16                         fixedSize: Size(300, 80),
17                         textStyle: TextStyle(fontSize: 25, fontWeight: FontWeight.bold),
18                         primary: Colors.yellow,
19                         onPrimary: Colors.black,
20                         elevation: 15,
21                         shadowColor: Colors.yellow,
22                         side: BorderSide(color: Colors.black87, width: 2),
23                         //alignment: Alignment.topLeft,
24                         //shape: CircleBorder(),
25                     ),
26                 ), // ElevatedButton
27             ), // Center
28         ); // Scaffold
29     }
30 }
```

Add Icon to Button

The screenshot shows the Visual Studio Code interface with the file `Home.dart` open. The code defines a `ElevatedButton` with an `Icon` and a `Text` label. The mobile app preview on the right shows a yellow button with the text "Let's Begin" and a shopping cart icon.

```
lib > Home.dart > ...  
1 import 'package:flutter/cupertino.dart';  
2 import 'package:flutter/material.dart';  
3  
4 class Home extends StatelessWidget {  
5   const Home({Key? key}) : super(key: key);  
6  
7   @override  
8   Widget build(BuildContext context) {  
9     return Scaffold(  
10       body: Center(  
11         child: ElevatedButton.icon(  
12           icon: Icon(Icons.add_shopping_cart_outlined),  
13           label: Text("Let's Begin"),  
14           onPressed: () {},  
15           style: ElevatedButton.styleFrom(  
16             padding: EdgeInsets.all(20.0),  
17             fixedSize: Size(300, 80),  
18             textStyle: TextStyle(fontSize: 25, fontWeight: FontWeight.bold),  
19             primary: Colors.yellow,  
20             onPrimary: Colors.black,  
21             elevation: 15,  
22             shadowColor: Colors.yellow,  
23             side: BorderSide(color: Colors.black87, width: 2),  
24             //alignment: Alignment.topLeft,  
25             shape: StadiumBorder(),  
26           ), // ElevatedButton.icon  
27         ), // Center  
28       ); // Scaffold  
29     }  
30   }  
31 }
```

Change to Add Row in Button

The screenshot shows the Visual Studio Code interface with the file `Home.dart` open. The code has been modified to use a `Row` widget inside the `ElevatedButton`. The mobile app preview on the right shows a yellow button with the text "Let's Begin" and a shopping cart icon.

```
lib > Home.dart > ⚙️ Home > ⚙️ build  
1 import 'package:flutter/cupertino.dart';  
2 import 'package:flutter/material.dart';  
3  
4 class Home extends StatelessWidget {  
5   const Home({Key? key}) : super(key: key);  
6  
7   @override  
8   Widget build(BuildContext context) {  
9     return Scaffold(  
10       body: Center(  
11         child: ElevatedButton(  
12           child: Row(  
13             mainAxisAlignment: MainAxisAlignment.center,  
14             children: [  
15               Text("Let's Begin"),  
16               Icon(Icons.add_shopping_cart_outlined),  
17             ],  
18           ), // Row  
19           //icon: Icon(Icons.add_shopping_cart_outlined),  
20           //label: Text("Let's Begin"),  
21           onPressed: () {},  
22           style: ElevatedButton.styleFrom(  
23             padding: EdgeInsets.all(20.0),  
24             fixedSize: Size(300, 80),  
25             textStyle: TextStyle(fontSize: 25, fontWeight: FontWeight.bold),  
26             primary: Colors.yellow,  
27           ),  
28         ), // Center  
29       ); // Scaffold  
30     }  
31 }
```

Modify for Design TextButton

The screenshot shows the Android Studio interface with the code editor open. The code defines a `TextButton` with a yellow background, bold black text, and a black border. The button's text is "Let's Begin". The preview on the right shows a smartphone screen with the button centered.

```
6
7   @override
8   Widget build(BuildContext context) {
9     return Scaffold(
10       body: Center(
11         child: TextButton(
12           child: Row(
13             mainAxisAlignment: MainAxisAlignment.center,
14             children: [
15               Text("Let's Begin"),
16               Icon(Icons.add_shopping_cart_outlined),
17             ],
18           ), // Row
19           //icon: Icon(Icons.add_shopping_cart_outlined),
20           //label: Text("Let's Begin"),
21           onPressed: () {},
22           style: ElevatedButton.styleFrom(
23             padding: EdgeInsets.all(20.0),
24             fixedSize: Size(300, 80),
25             textStyle: TextStyle(fontSize: 25, fontWeight: FontWeight.bold),
26             //primary: Colors.yellow,
27             onPrimary: Colors.black,
28             //elevation: 15,
29             shadowColor: Colors.yellow,
30             //side: BorderSide(color: Colors.black87, width: 2),
31             //alignment: Alignment.topLeft,
32             shape: StadiumBorder(),
33           ), // TextButton
34         ), // Center
35       ); // Scaffold
36     }
37 }
```

Modify for Design OutlinedButton

The screenshot shows the Android Studio interface with the code editor open. The code defines an `OutlinedButton` with a yellow background, bold black text, and a black border. The button's text is "Let's Begin". The preview on the right shows a smartphone screen with the button centered.

```
6
7   @override
8   Widget build(BuildContext context) {
9     return Scaffold(
10       body: Center(
11         child: OutlinedButton(
12           child: Row(
13             mainAxisAlignment: MainAxisAlignment.center,
14             children: [
15               Text("Let's Begin "),
16               Icon(Icons.add_shopping_cart_outlined),
17             ],
18           ), // Row
19           onPressed: () {},
20           style: ElevatedButton.styleFrom(
21             padding: EdgeInsets.all(20.0),
22             fixedSize: Size(300, 80),
23             textStyle: TextStyle(fontSize: 25, fontWeight: FontWeight.bold),
24             primary: Colors.yellow,
25             onPrimary: Colors.black,
26             elevation: 15,
27             shadowColor: Colors.yellow,
28             side: BorderSide(color: Colors.black87, width: 2),
29             alignment: Alignment.topLeft,
30             shape: StadiumBorder(),
31           ),
32         ), // OutlinedButton
33       ), // Center
34     ); // Scaffold
35   }
36 }
```

Button onPressed : showDialog

The screenshot shows the Android Studio interface with three tabs: Shopping.dart, form.dart, and main.dart. The Shopping.dart tab is active, displaying the following code:

```
18:     Text("Let's Begin "),
19:     Icon(Icons.add_shopping_cart_outlined),
20:   ),
21:   // Row
22:   //icon: Icon(Icons.add_shopping_cart_outlined),
23:   //label: Text("Let's Begin"),
24:   onPressed: () => showDialog<String>(
25:     context: context,
26:     builder: (BuildContext context) => AlertDialog(
27:       title: const Text('AlertDialog Title'),
28:       content: const Text('AlertDialog description'),
29:       actions: <Widget>[
30:         TextButton(
31:           onPressed: () => Navigator.pop(context, 'Cancel'),
32:           child: const Text('Cancel'),
33:         ), // TextButton
34:         TextButton(
35:           onPressed: () => Navigator.pop(context, 'OK'),
36:           child: const Text('OK'),
37:         ), // TextButton
38:       ],
39:     ), // AlertDialog
40:   ),
41: ),
```

To the right of the code editor is a preview window showing a smartphone screen with a modal dialog titled "AlertDialog Title" containing the text "AlertDialog description". The dialog has two buttons: "Cancel" and "OK".

Button onPressed : MaterialPageRoute

1. Create Shopping.dart

The screenshot shows the Android Studio interface with three tabs: Shopping.dart, form.dart, and main.dart. The Shopping.dart tab is active, displaying the following code:

```
9:   return Scaffold(
10:     appBar: AppBar(
11:       title: Text("Form"),
12:     ), // AppBar
13:     body: Center(
14:       child: ElevatedButton(
15:         child: Row(
16:           mainAxisAlignment: MainAxisAlignment.center,
17:           children: [
18:             Text("Go to Shopping "),
19:             Icon(Icons.add_shopping_cart_outlined),
20:           ],
21:         ), // Row
22:         onPressed: () {
23:           Navigator.push(
24:             context,
25:             MaterialPageRoute(
26:               builder: (context) {
27:                 return formShopping();
28:               },
29:             ),
30:           );
31:         },
32:       ),
33:     ),
34:   );
```

To the right of the code editor are two smartphone screens illustrating a MaterialPageRoute transition. The left screen shows a button labeled "Go to Shopping". A blue arrow points from this button to the right screen, which shows the same button now labeled "Go back!". The top bar of the right screen says "← Shopping Screen".

Shopping.dart

The screenshot shows the Android Studio interface with three tabs: Shopping.dart, form.dart, and main.dart. The Shopping.dart tab is active, displaying the following code:

```
1: import 'package:flutter/material.dart';
2:
3: class formShopping extends StatelessWidget {
4:   const formShopping({Key? key}) : super(key: key);
5:
6:   @override
7:   Widget build(BuildContext context) {
8:     return Scaffold(
9:       appBar: AppBar(
10:         title: Text('Shopping Screen'),
11:       ), // AppBar
12:       body: Center(
13:         child: ElevatedButton(
14:           onPressed: () {
15:             Navigator.pop(context);
16:           },
17:           child: const Text('Go back!'),
18:         ), // ElevatedButton
19:       ), // Center
20:     ); // Scaffold
21:
22: }
```

To the right of the code editor are two smartphone screens illustrating a MaterialPageRoute transition. The left screen shows a button labeled "Go back!". A blue arrow points from this button to the right screen, which shows the same button now labeled "Go to Shopping". The top bar of the right screen says "← Shopping Screen".

Design UI Form

Preparation the app Screen for UI with Extract Method.

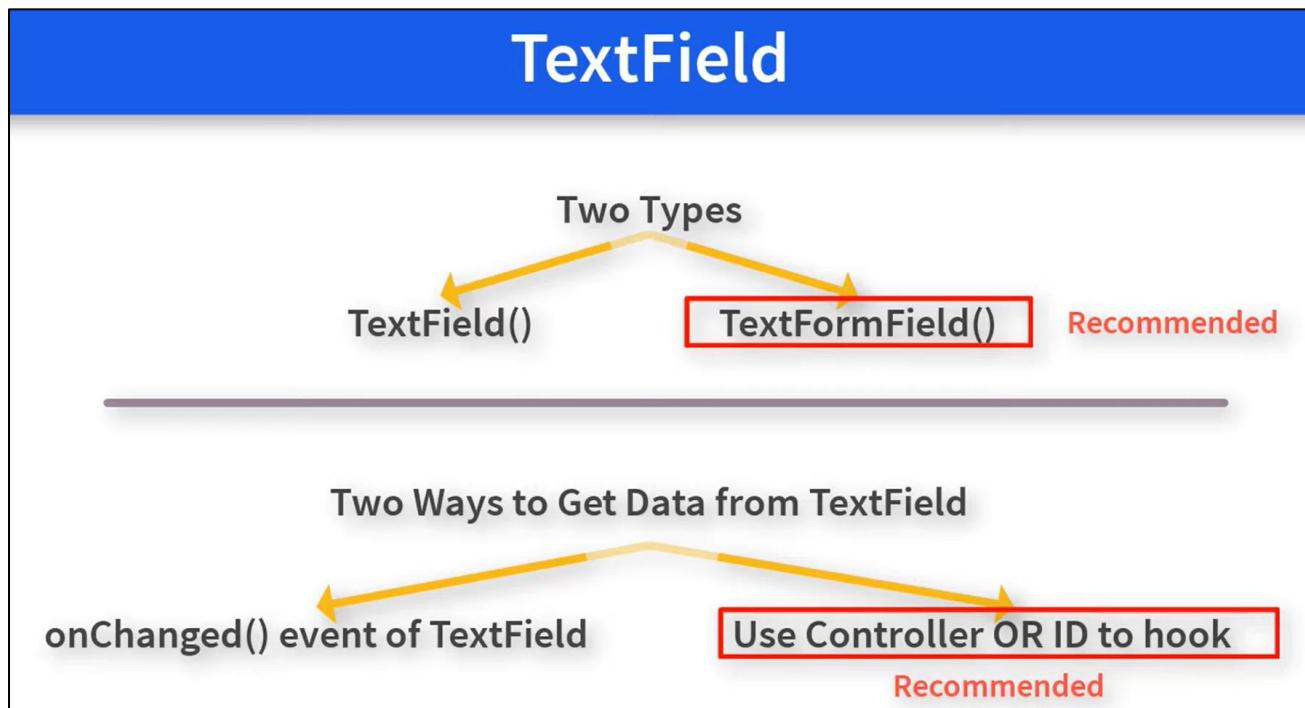
1. Extract Method

The screenshot illustrates the extraction of a UI component into a separate method. In the code editor, the body of a Scaffold widget is selected. A context menu is open, and the 'Extract Method' option is highlighted. A completion dialog shows the proposed method name 'myBtn'. The extracted code is shown in a new section of the file:

```
    @override
    Widget build(BuildContext context) {
      return Scaffold(
        appBar: AppBar(
          title: Text("Form"),
        ), // AppBar
        body: myBtn(context),
      ); // Scaffold
    }

    Center myBtn(BuildContext context) {
      return Center(
        child: ElevatedButton(
```

TextField & TextFormField



2. Create TextField and TextFormField

The screenshot shows the Android Studio code editor with three tabs: Shopping.dart, form.dart, and main.dart. The form.dart tab is active, displaying the following code:

```
lib > lib > form.dart > MyForm > build
1 import 'package:flutter/material.dart';
2 import 'Shopping.dart';
3
4 class MyForm extends StatelessWidget {
5   const MyForm({Key? key}) : super(key: key);
6
7   @override
8   Widget build(BuildContext context) {
9     return Scaffold(
10       appBar: AppBar(
11         title: Text("Form"),
12       ), // AppBar
13       body: Container(
14         padding: EdgeInsets.all(20.0),
15         child: ListView(
16           children: [
17             TextField(),
18             TextFormField(),
19           ],
20         ), // ListView
21       ), // Container
22     ); // Scaffold
23   }
24
25 > Center myBtn(BuildContext context) { ...
26
27
28
29
30
31
32
33
34
35 > Center myBtn(BuildContext context) { ...
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90 }
```

A red box highlights the area from line 25 to line 90, which contains the definition for the `myBtn` function.

To the right of the code editor is a preview window showing a black smartphone screen with a blue header bar labeled "Form". Below the header, there is a single-line text input field containing the placeholder text "dfsd" and the suffix "sdfsfdsf".

The screenshot shows the same setup as the previous one, but the code has been modified. The `TextFormField` now includes an `InputDecoration` with a placeholder and an icon:

```
lib > lib > form.dart > MyForm > build
6
7   @override
8   Widget build(BuildContext context) {
9     return Scaffold(
10       appBar: AppBar(
11         title: Text("Form"),
12       ), // AppBar
13       body: Container(
14         padding: EdgeInsets.all(20.0),
15         child: ListView(
16           children: [
17             TextField(),
18             TextFormField(
19               decoration: InputDecoration(
20                 hintText: 'Product Name',
21                 icon: Icon(Icons.verified_user_outlined),
22               ), // InputDecoration
23             ),
24             TextFormField(
25               decoration: InputDecoration(
26                 labelText: 'Product Name',
27                 icon: Icon(Icons.verified_user_outlined),
28               ), // InputDecoration
29             ),
30           ],
31         ), // ListView
32       ), // Container
33     ); // Scaffold
34
35 > Center myBtn(BuildContext context) { ...
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59
60
61
62
63
64
65
66
67
68
69
69
70
71
72
73
74
75
76
77
78
79
79
80
81
82
83
84
85
86
87
88
89
89
90 }
```

A red box highlights the area from line 25 to line 90, which contains the definition for the `myBtn` function.

To the right of the code editor is a preview window showing the same black smartphone screen. The second text input field now has a placeholder "Product Name" and a blue checkmark icon next to it. The keyboard is visible at the bottom of the screen, showing the English language and a green checkmark button.

Change TextFormField

The screenshot shows the Android Studio interface with three tabs at the top: Shopping.dart, form.dart (selected), and main.dart. The code in form.dart is as follows:

```
8
9  @override
10 Widget build(BuildContext context) {
11   return Scaffold(
12     appBar: AppBar(
13       title: Text("Form"),
14     ), // AppBar
15     body: Container(
16       padding: EdgeInsets.all(20.0),
17       child: ListView(
18         children: [
19           TextFormField(
20             decoration: InputDecoration(
21               labelText: 'Product Name',
22               prefixIcon: Icon(Icons.verified_user_outlined),
23               border: OutlineInputBorder(),
24             ), // InputDecoration
25           ), // TextFormField
26           SizedBox(
27             height: 20.0,
28           ), // SizedBox
29           TextFormField(
30             decoration: InputDecoration(
31               labelText: 'Customer Name',
32               prefixIcon: Icon(Icons.verified_user_outlined),
33               border: OutlineInputBorder(),
34             ), // InputDecoration
35           ), // TextFormField
36           //Text("Product Name is : ${_productName}"),
37           ],
38         ), // ListView
39       ), // Container
40     ),
41   );
42 }
```

The preview on the right shows a smartphone screen with a blue header bar labeled "Form". Below it are two text input fields. The first field has a placeholder "Product Name" and a small shield icon to its left. The second field has a placeholder "Customer Name" and a similar shield icon.

Add Button in Form

The screenshot shows the Android Studio interface with three tabs at the top: Shopping.dart, form.dart (selected), and main.dart. The code in form.dart is as follows:

```
1 import 'package:flutter/material.dart';
2 import 'Shopping.dart';
3
4 var _productName;
5
6 class MyForm extends StatelessWidget {
7   const MyForm({Key? key}) : super(key: key);
8
9   @override
10  Widget build(BuildContext context) {
11    return Scaffold(
12      appBar: AppBar(
13        title: Text("Form"),
14      ), // AppBar
15      body: Container(
16        padding: EdgeInsets.all(20.0),
17        child: ListView(
18          children: [
19            TextFormField( // TextFormField ...
20            SizedBox( // SizedBox ...
21            TextFormField( // TextFormField ...
22            SizedBox( // SizedBox ...
23            myBtn(context),
24            //Text("Product Name is : ${_productName}"),
25            ],
26          ), // ListView
27        ), // Container
28      ); // Scaffold
29    }
30
31    Center myBtn(BuildContext context) {
32      return Center(
33        child: Text("Go to Shopping"),
34      );
35    }
36  }
37 }
```

The preview on the right shows the same smartphone screen as before, but now it includes a blue button at the bottom right with the text "Go to Shopping" and a small shopping bag icon.

Set Button Style

The screenshot shows the Android Studio interface. On the left is the code editor with three tabs: Shopping.dart, form.dart, and main.dart. The current file is form.dart, which contains the following code:

```
46 Center myBtn(BuildContext context) {
47   return Center(
48     child: ElevatedButton(
49       child: Row(
50         mainAxisAlignment: MainAxisAlignment.center,
51         children: [
52           Text("Go to Shopping"),
53           Icon(Icons.add_shopping_cart_outlined),
54         ],
55       ), // Row
56     onPressed: () {
57       Navigator.push(
58         context,
59         MaterialPageRoute(
60           builder: (context) {
61             return formShopping();
62           },
63         ), // MaterialPageRoute
64       );
65     },
66     style: ElevatedButton.styleFrom(
67       padding: EdgeInsets.all(20.0),
68       fixedSize: Size(300, 80),
69       textStyle: TextStyle(fontsize: 25, fontWeight: FontWeight.bold),
70       primary: Colors.blueAccent,
71       onPrimary: Colors.white,
72       elevation: 15,
73       shadowColor: Colors.blueAccent,
74       //side: BorderSide(color: colors.black87, width: 2),
75       alignment: Alignment.topLeft,
76       shape: StadiumBorder(),
77     ),
78   ), // ElevatedButton
79 } // Center
80 }
81 }
```

A red box highlights the entire body of the `myBtn` function. To the right is a preview window showing a smartphone screen with a blue header labeled "Form". It has two input fields: "Product Name" with the placeholder "Customer Name" and "Customer Name" with the placeholder "Customer Name". Below them is a blue button with the text "Go to Shopping". A vertical toolbar on the right side of the preview window provides various device and orientation controls.

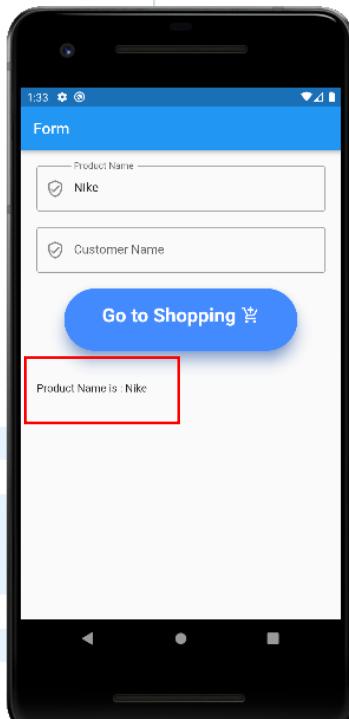
Change to StatefulWidget and Set Var _ProductName

The screenshot shows the Android Studio interface. The code editor is displaying form.dart, which has been modified to use a `StatefulWidget`. The code is as follows:

```
1 import 'package:flutter/material.dart';
2 import 'Shopping.dart';
3
4 class MyForm extends StatefulWidget {
5   const MyForm({Key? key}) : super(key: key);
6
7   @override
8   State<MyForm> createState() => _MyFormState();
9 }
10
11 class _MyFormState extends State<MyForm> {
12   var _productName;
13
14   void _updateText(val) {
15     setState(() {
16       _productName = val;
17     });
18   }
19
20   @override
21   Widget build(BuildContext context) {
22     return Scaffold(
23       appBar: AppBar(
24         title: Text("Form"),
25       ), // AppBar
26       body: Container(
27         padding: EdgeInsets.all(20.0),
28         child: ListView(
```

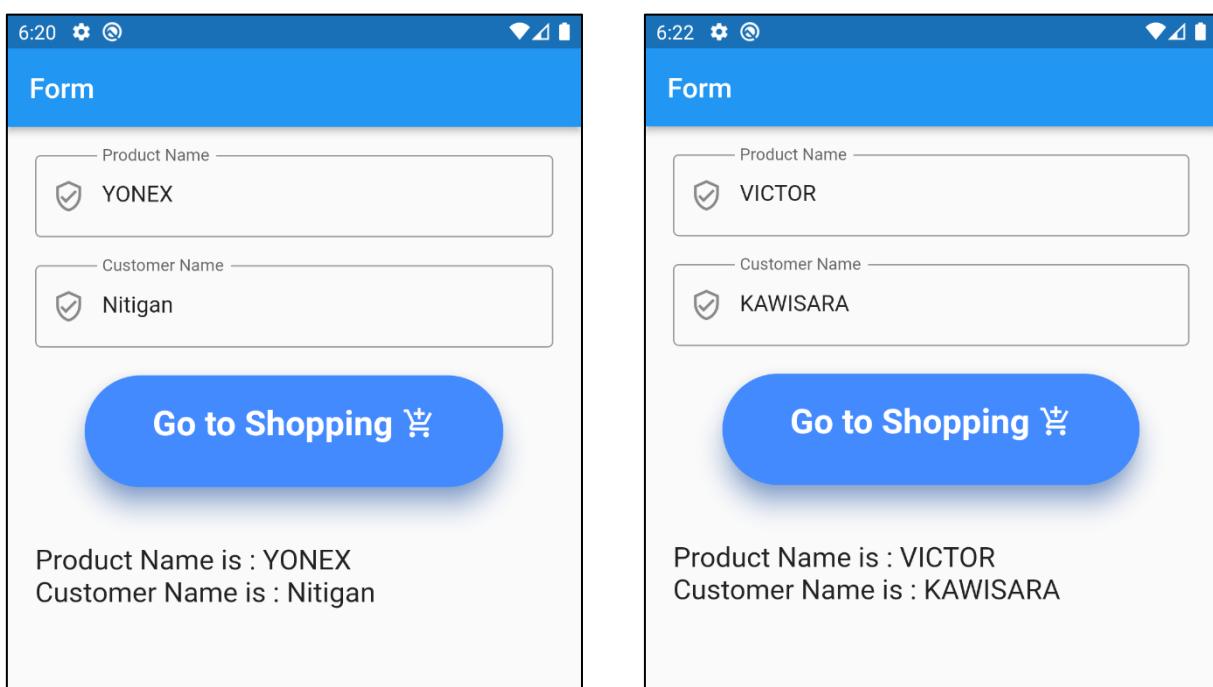
A red box highlights the declaration of `_MyFormState` and its constructor. To the right is a preview window showing a smartphone screen with a blue header labeled "Form". It has two input fields: "Product Name" with the placeholder "Nike" and "Customer Name" with the placeholder "Customer Name". Below them is a blue button with the text "Go to Shopping". At the bottom of the screen, there is a red box around a text overlay that says "Product Name is: Nike". A vertical toolbar on the right side of the preview window provides various device and orientation controls.

Display ProductName with onChange



```
Shoppping.dart  form.dart X main.dart
lib > form.dart > _MyFormState > build
18 }
19
20 @override
21 Widget build(BuildContext context) {
22     return Scaffold(
23         appBar: AppBar(
24             title: Text("Form"),
25         ), // AppBar
26         body: Container(
27             padding: EdgeInsets.all(20.0),
28             child: ListView(
29                 children: [
30                     TextFormField(
31                         onChanged: (val) {
32                             _updateText(val);
33                         },
34                         decoration: InputDecoration( // InputDecoration ...
35                         ),
36                         ),
37                     SizedBox( // SizedBox ...
38                     ),
39                     TextFormField( // TextFormField ...
40                     ),
41                     SizedBox( // SizedBox ...
42                     ),
43                     myBtn(context),
44                     SizedBox( // SizedBox ...
45                     ),
46                     Text("Product Name is : ${_productName}"),
47                 ],
48             ), // ListView
49         ), // Container
50     ); // Scaffold
51 }
52
53 }
```

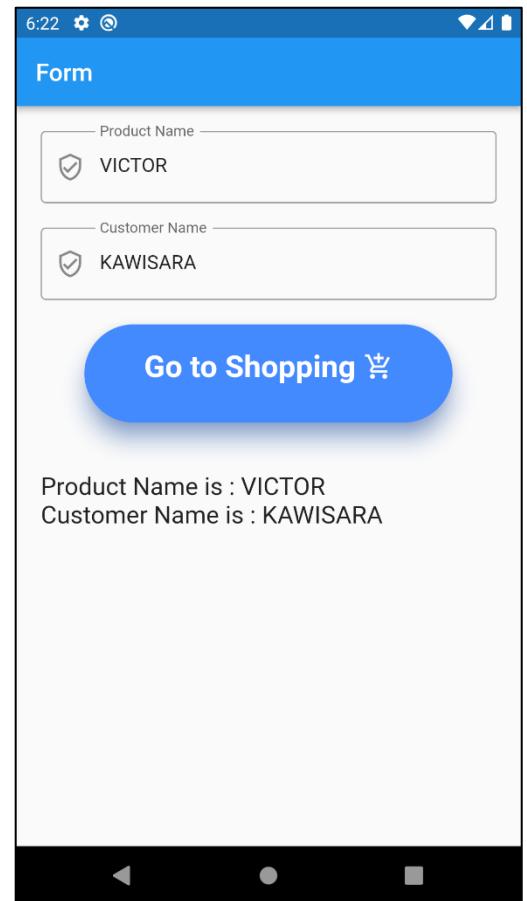
Modify to Show ProductName and CustomerName



Display ProductName and CustomerName with Controller

```
Go Run Terminal Help           form.dart - flutter_additem - Visual Studio Code
Shoping.dart  form.dart  main.dart

lib > lib form.dart > _MyFormState
1 import 'package:flutter/material.dart';
2 import 'Shoping.dart';
3
4 class MyForm extends StatefulWidget {
5   const MyForm({Key? key}) : super(key: key);
6
7   @override
8   State<MyForm> createState() => _MyFormState();
9 }
10
11 class _MyFormState extends State<MyForm> {
12   var _productName;
13   var _customerName;
14
15   final _productController = TextEditingController();
16   final _customerController = TextEditingController();
17
18   void initState() {
19     super.initState();
20     _productController.addListener(_updateText);
21     _customerController.addListener(_updateText);
22   }
23
24   void _updateText() {
25     setState(() {
26       _productName = _productController.text;
27       _customerName = _customerController.text;
28     });
29   }
30
31
32   @override
33   Widget build(BuildContext context) {
34     return Scaffold(
35       appBar: AppBar(
36         title: Text("Form"),
37       ), // AppBar
38       body: Container(
39         padding: EdgeInsets.all(20.0),
40         child: ListView(
41           children: [
42             TextFormField(
43               controller: _productController,
44               decoration: InputDecoration( // InputDecoration ...
45             ), // TextFormField
46             SizedBox( // SizedBox ...
47             TextFormField(
48               controller: _customerController,
49               decoration: InputDecoration(
50                 labelText: "Customer Name",
51                 prefixIcon: Icon(Icons.verified_user_outlined),
52                 border: OutlineInputBorder(),
53               ), // InputDecoration
54             ), // TextFormField
55             SizedBox( // SizedBox ...
56             myBtn(context),
57             SizedBox( // SizedBox ...
58             Text(
59               "Product Name is : ${_productController.text}",
60               style: TextStyle(fontSize: 20),
61             ), // Text
62             Text(
63               "Customer Name is : ${_customerController.text}",
64               style: TextStyle(fontSize: 20),
65             ), // Text
66           ],
67         ), // ListView
68       ), // Container
69     ); // Scaffold
70   }
71
72   Center myBtn(BuildContext context) { ...
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117 }
```



Pass Value to Shopping Screen

```
lib > Shopping.dart > form.dart > main.dart
lib > form.dart > _MyFormState > myBtn
80
81     Center myBtn(BuildContext context) {
82         return Center(
83             child: ElevatedButton(
84                 child: Row(
85                     mainAxisAlignment: MainAxisAlignment.center,
86                     children: [
87                         Text("Go to Shopping "),
88                         Icon(Icons.add_shopping_cart_outlined),
89                     ],
90                 ), // Row
91                 onPressed: () {
92                     Navigator.push(
93                         context,
94                         MaterialPageRoute(
95                             builder: (context) {
96                                 return formShopping(
97                                     productName: _productController.text,
98                                 ); // formShopping
99                             },
100                         ), // MaterialPageRoute
101                     );
102                 },
103             ),
104         );
105     }
106 }
```

```
Go Run Terminal Help Shopping.dart - flutter_additem - Visual Studio Code
lib > Shopping.dart > formShopping > build
1 import 'package:flutter/material.dart';
2
3 class formShopping extends StatelessWidget {
4     formShopping({Key? key, required this.productName}) : super(key: key);
5     String productName;
6
7     @override
8     Widget build(BuildContext context) {
9         return Scaffold(
10             appBar: AppBar(
11                 title: Text('Shopping Screen'),
12                 leading: IconButton(
13                     onPressed: () {
14                         Navigator.pop(context);
15                     },
16                     icon: const Icon(Icons.arrow_back),
17                 ), // IconButton
18             ), // AppBar
19             body: Container(
20                 padding: EdgeInsets.all(20.0),
21                 child: ListView(
22                     children: [
23                         ListTile(
24                             leading: Icon(Icons.account_balance_wallet_outlined),
25                             title: Text(productName),
26                         ), // ListTile
27                     ],
28                 ), // ListView
29             ), // Container
30         ); // Scaffold
31     }
32 }
```

The screenshot shows the Shopping.dart file open in Visual Studio Code. The code defines a `formShopping` widget that extends `StatelessWidget`. It has a `productName` parameter. The `build` method creates a `Scaffold` with an `AppBar` containing a back button and a `body` containing a `ListView` with a single `ListTile` item. The `ListTile` has a leading icon and a title set to `productName`. A red box highlights the `productName` parameter in the constructor and the `title` property of the `ListTile`. To the right, a Flutter emulator displays a smartphone screen titled "Shopping Screen" with a single list item containing the text "ADIDAS".

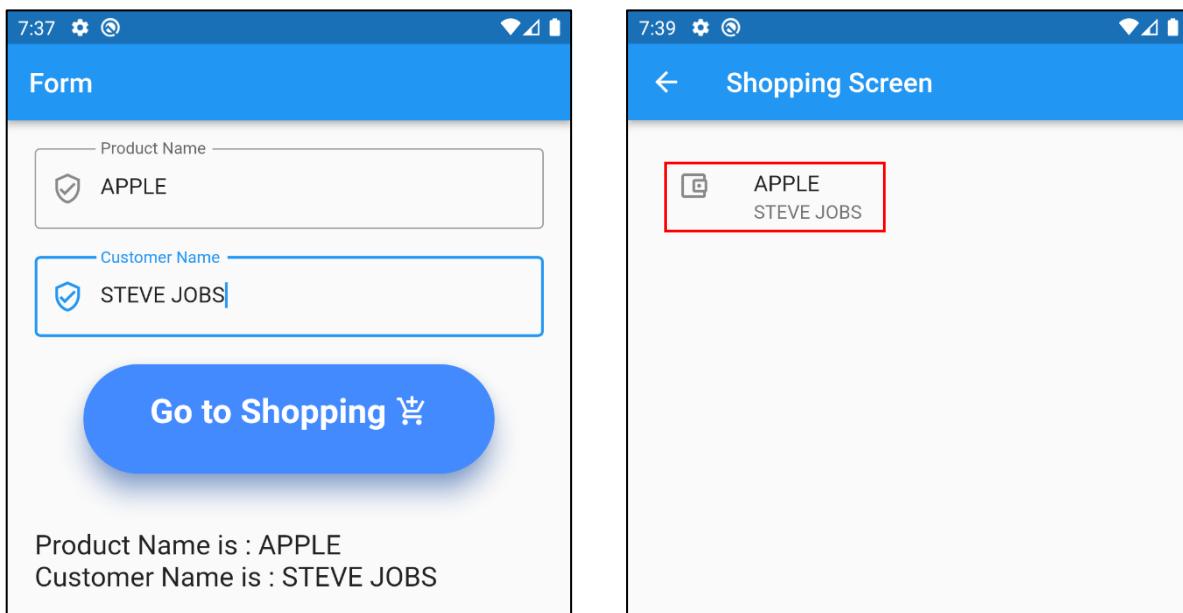
Set AppBar Title with ProductName

The screenshot shows a code editor with three tabs: Shopping.dart, form.dart, and main.dart. The Shopping.dart tab is active, displaying the following Dart code:

```
lib > Shopping.dart > formShopping > build
1 import 'package:flutter/material.dart';
2
3 class formShopping extends StatelessWidget {
4   formShopping({Key? key, required this.productName}) : super(key: key);
5   String productName;
6
7   @override
8   Widget build(BuildContext context) {
9     return Scaffold(
10       appBar: AppBar(
11         title: Text(productName),
12         //title: Text('Shopping Screen'),
13         leading: IconButton(
14           onPressed: () {
15             Navigator.pop(context);
16           },
17           icon: const Icon(Icons.arrow_back),
18         ), // IconButton
19       ), // AppBar
20     );
21   }
22 }
```

To the right of the code editor is a screenshot of a running Flutter application on an iPhone. The app's title bar is blue with white text. A red box highlights the title "ADIDAS". Below the title bar, the screen displays the word "ADIDAS".

Display ProductName and CustomerName



Workshop : Create TextField for Price and Number of Product.

Display Result of Total Price of Order.

