

## L9 -Q1

main.dart 9+ x

main.dart > MyApp > build

```
1  import 'package:flutter/material.dart';
2  import 'package:flutter_application_1/pages/page1.dart';
3  import 'package:flutter_application_1/pages/page2.dart';
4  import 'package:flutter_application_1/pages/page3.dart';
5  import 'package:flutter_application_1/pages/page4.dart';
6  import 'package:flutter_application_1/pages/page5.dart';
7  import 'pages/home_screen.dart';
8  //Task01: import page1,2,3,4,5
   Run | Debug
9  void main() {
10 |   runApp(MyApp());
11 | }
12
13 class MyApp extends StatelessWidget {
14 |   const MyApp({super.key});
15 |
16 |   @override
17 |   Widget build(BuildContext context) {
18 |     return MaterialApp(
19 |       title: 'Flutter Named Routes',
20 |       initialRoute: '/',
21 |       //Task02: define routing
22 |       routes: {
23 |         '/': (context) => HomePage(),
24 |         '/page1': (context) => Page1(),
25 |         '/page2': (context) => Page2(),
26 |         '/page3': (context) => Page3(),
27 |         '/page4': (context) => Page4(),
28 |         '/page5': (context) => Page5(),
29 |       },
30 |     );
31 |   }
32 | }
33
34
```

EXPLORER

Q1\_6687049

- pages
  - home\_screen.dart 9+
  - page1.dart 9+
  - page2.dart 9+
  - page3.dart 9+
  - page4.dart 9+
  - page5.dart 9+
  - main.dart 9+

main.dart 9+ page1.dart 9+ x

pages > page1.dart > ...

```
1 import 'package:flutter/material.dart';
2 class Page1 extends StatelessWidget {
3   @override
4   Widget build(BuildContext context) {
5     return Scaffold(
6       appBar: AppBar(title: Text('Page 1')),
7       body: Center(child: Text('Page 1', style: TextStyle(fontSize: 24))),
8       floatingActionButton: FloatingActionButton(
9         onPressed: (){
10           //task04
11           Navigator.pop(context);
12         },
13       child: Icon(Icons.arrow_back),
14     ),
15   );
16 }
17 }
```

EXPLORER

Q1\_6687049

- pages
  - home\_screen.dart 9+
  - page1.dart 9+
  - page2.dart 9+
  - page3.dart 9+
  - page4.dart 9+
  - page5.dart 9+
  - main.dart 9+

main.dart 9+ page2.dart 9+ x

pages > page2.dart > ...

```
1 import 'package:flutter/material.dart';
2 class Page2 extends StatelessWidget {
3   @override
4   Widget build(BuildContext context) {
5     return Scaffold(
6       appBar: AppBar(title: Text('Page 2')),
7       body: Center(child: Text('Page 2', style: TextStyle(fontSize: 24))),
8       floatingActionButton: FloatingActionButton(
9         onPressed: (){
10           //task04
11           Navigator.pop(context);
12         },
13       child: Icon(Icons.arrow_back),
14     ),
15   );
16 }
17 }
```

EXPLORER

Q1\_6687049

pages

home\_screen.dart 9+

page1.dart 9+

page2.dart 9+

page3.dart 9+

page4.dart 9+

page5.dart 9+

main.dart 9+

OUTLINE

TIMELINE

DEPENDENCIES

main.dart 9+

home\_screen.dart 9+ X

pages > home\_screen.dart > ...

```
1 import 'package:flutter/material.dart';
2
3 class HomePage extends StatelessWidget {
4   @override
5   Widget build(BuildContext context) {
6     return Scaffold(
7       appBar: AppBar(title: Text('Home')),
8       drawer: Drawer(
9         child: ListView(
10          padding: EdgeInsets.zero,
11          children: [
12            DrawerHeader(
13              decoration: BoxDecoration(color: Colors.blue),
14              child: Text('Menu', style: TextStyle(color: Colors.white, fontSize: 24)),
15            ),
16            ListTile(
17              title: Text('Page 1'),
18              onTap: (){
19                //Task03: using Navigator.pushNamed
20                Navigator.pushNamed(context, '/page1');
21              },
22            ),
23            ListTile(
24              title: Text('Page 2'),
25              onTap: (){
26                Navigator.pushNamed(context, '/page2');
27              },
28            ),
29            ListTile(
30              title: Text('Page 3'),
31              onTap: (){
32                Navigator.pushNamed(context, '/page3');
33              },
34            ),
35            ListTile(
36              title: Text('Page 4'),
37              onTap: (){
38                Navigator.pushNamed(context, '/page4');
39              },
40            ),
41            ListTile(
42              title: Text('Page 5'),
43              onTap: (){
44                Navigator.pushNamed(context, '/page5');
45              },
46            ),
47          ],
48        ),
49      ),
50    );
51  }
```

EXPLORER

Q1\_6687049

pages

home\_screen.dart 9+

page1.dart 9+

page2.dart 9+

page3.dart 9+

page4.dart 9+

page5.dart 9+

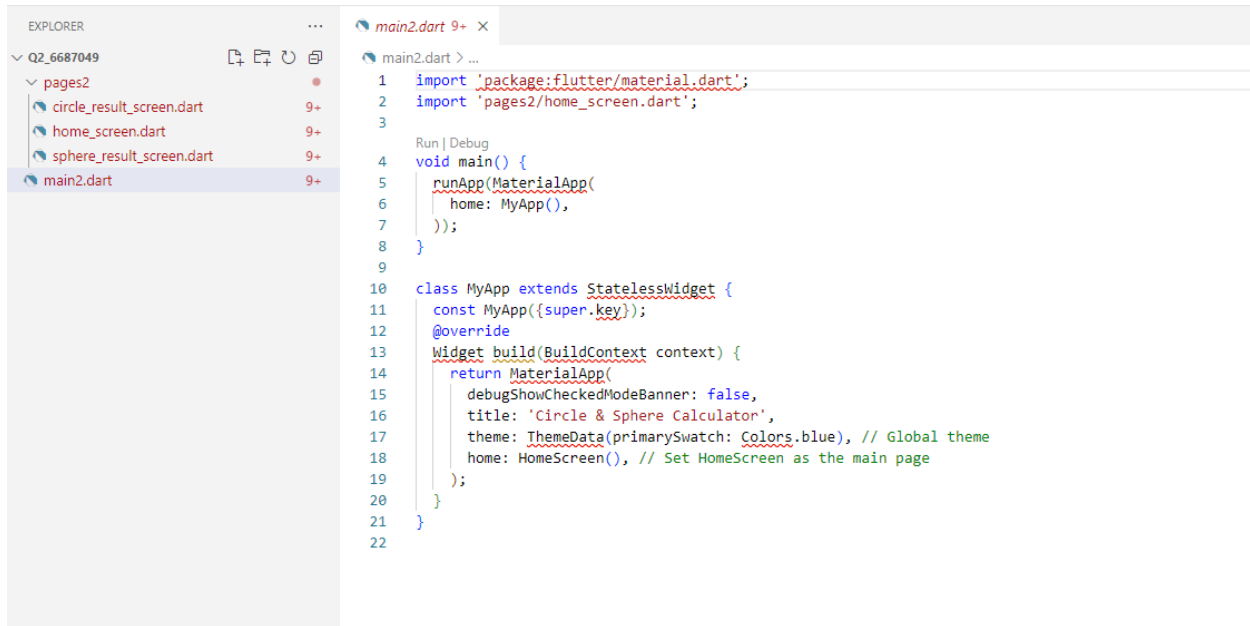
main.dart 9+

main.dart 9+ home\_screen.dart 9+ x

pages > home\_screen.dart > ...

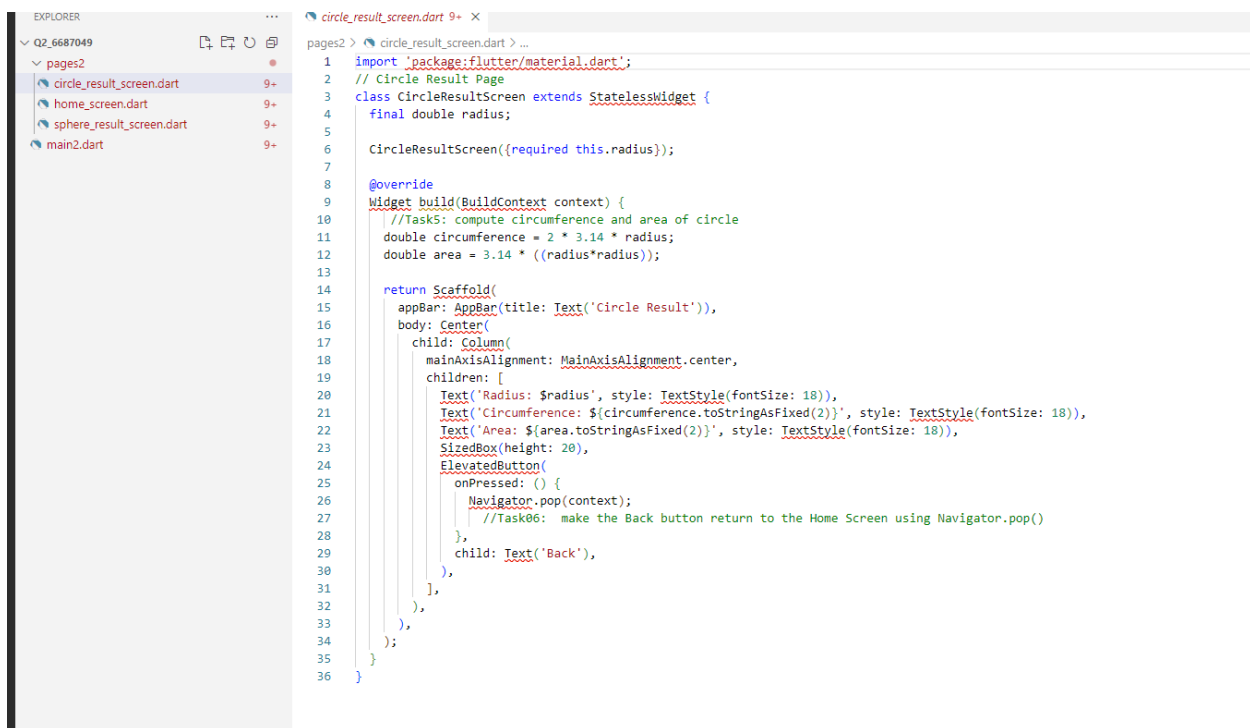
```
3 class HomePage extends StatelessWidget {
5   Widget build(BuildContext context) {
16     ListTile(
17       title: Text('Page 1'),
18       onTap: () {
19         //Task03: using Navigator.pushNamed
20         Navigator.pushNamed(context, '/page1');
21       },
22     ),
23     ListTile(
24       title: Text('Page 2'),
25       onTap: () {
26         Navigator.pushNamed(context, '/page2');
27       },
28     ),
29     ListTile(
30       title: Text('Page 3'),
31       onTap: () {
32         Navigator.pushNamed(context, '/page3');
33       },
34     ),
35     ListTile(
36       title: Text('Page 4'),
37       onTap: () {
38         Navigator.pushNamed(context, '/page4');
39       },
40     ),
41     ListTile(
42       title: Text('Page 5'),
43       onTap: () {
44         Navigator.pushNamed(context, '/page5');
45       },
46     ),
47   ],
48 ),
49 );
50 body: Center(child: Text('Welcome to Home Page', style: TextStyle(fontSize: 24))),
51 );
52 }
53 }
```

## L9-Q2



```
EXPLORER
Q2_6687049
  pages2
    circle_result_screen.dart
    home_screen.dart
    sphere_result_screen.dart
    main2.dart

main2.dart 9+ x
main2.dart > ...
1 import 'package:flutter/material.dart';
2 import 'pages2/home_screen.dart';
3
4 void main() {
5   runApp(MaterialApp(
6     home: MyApp(),
7   ));
8 }
9
10 class MyApp extends StatelessWidget {
11   const MyApp({super.key});
12   @override
13   Widget build(BuildContext context) {
14     return MaterialApp(
15       debugShowCheckedModeBanner: false,
16       title: 'Circle & Sphere Calculator',
17       theme: ThemeData(primarySwatch: Colors.blue), // Global theme
18       home: HomeScreen(), // Set HomeScreen as the main page
19     );
20   }
21 }
22
```



```
EXPLORER
Q2_6687049
  pages2
    circle_result_screen.dart
    home_screen.dart
    sphere_result_screen.dart
    main2.dart

circle_result_screen.dart 9+ x
pages2 > circle_result_screen.dart > ...
1 import 'package:flutter/material.dart';
2 // Circle Result Page
3 class CircleResultScreen extends StatelessWidget {
4   final double radius;
5
6   CircleResultScreen({required this.radius});
7
8   @override
9   Widget build(BuildContext context) {
10     //Task5: compute circumference and area of circle
11     double circumference = 2 * 3.14 * radius;
12     double area = 3.14 * (radius*radius);
13
14     return Scaffold(
15       appBar: AppBar(title: Text('Circle Result')),
16       body: Center(
17         child: Column(
18           mainAxisAlignment: MainAxisAlignment.center,
19           children: [
20             Text('Radius: $radius', style: TextStyle(fontSize: 18)),
21             Text('Circumference: ${(circumference.toStringAsFixed(2))}', style: TextStyle(fontSize: 18)),
22             Text('Area: ${(area.toStringAsFixed(2))}', style: TextStyle(fontSize: 18)),
23             SizedBox(height: 20),
24             ElevatedButton(
25               onPressed: () {
26                 Navigator.pop(context);
27                 //Task06: make the Back button return to the Home Screen using Navigator.pop()
28               },
29               child: Text('Back'),
30             ),
31           ],
32         ),
33       ),
34     );
35   }
36 }
```

```
EXPLORER
Q2_6687049
pages2
circle_result_screen.dart
home_screen.dart
sphere_result_screen.dart
main2.dart

home_screen.dart 9+ x
pages2 > home_screen.dart > _HomeScreenState > build
1 import 'package:flutter/material.dart';
2 import 'circle_result_screen.dart';
3 import 'sphere_result_screen.dart';
4
5 //Task 01: Import Required Screens
6 //import both circle_result_screen.dart and sphere_result_screen.dart.
7
8 class HomeScreen extends StatefulWidget {
9   const HomeScreen({super.key});
10
11   @override
12   _HomeScreenState createState() => _HomeScreenState();
13 }
14
15 class _HomeScreenState extends State<HomeScreen> {
16   // Task 02: Implement a Text Field Controller
17
18   // Create a TextEditingController.
19   // Assign this controller to the TextField widget so it can read user input.
20
21   String _selectedOption = 'Circle'; // Default selection
22   final TextEditingController _radiuscontroller = TextEditingController();
23
24
25   @override
26   Widget build(BuildContext context) {
27     return Scaffold(
28       appBar: AppBar(title: Text('Circle & Sphere Calculator')),
29       body: Padding(
30         padding: const EdgeInsets.all(16.0),
31         child: Column(
32           mainAxisAlignment: MainAxisAlignment.center,
33           crossAxisAlignment: CrossAxisAlignment.start,
34           children: [
35             TextField(
36               controller: _radiuscontroller,
37               keyboardType: TextInputType.number,
38               decoration: InputDecoration(
39                 labelText: 'Enter radius',
40                 border: OutlineInputBorder(),
41               ),
42             ),
43             SizedBox(height: 20),
44             Text("Select Shape:"),
45             Row(
46               children: [
47                 Radio(
48                   value: 'Circle',
```

```
EXPLORER
Q2_6687049
pages2
circle_result_screen.dart
home_screen.dart
sphere_result_screen.dart
main2.dart

home_screen.dart 9+ x
pages2 > home_screen.dart > _HomeScreenState > build
15 class _HomeScreenState extends State<HomeScreen> {
26   Widget build(BuildContext context) {
48     value: 'Circle',
49     groupValue: _selectedOption,
50     onChanged: (value) {
51       //Task 03: Handle Radio Button Selection
52       //When the Circle radio button is selected, update _selectedOption using setState().
53       setState(() {
54         _selectedOption =value.toString();
55       });
56     },
57   ),
58   Text('Circle'),
59   Radio(
60     value: 'Sphere',
61     groupValue: _selectedOption,
62     onChanged: (value) {
63       //Task 03: Handle Radio Button Selection
64       //When the Sphere radio button is selected, update _selectedOption using setState().
65       setState(() {
66         _selectedOption =value.toString();
67       });
68     },
69   ),
70   Text('Sphere'),
71 ],
72 ),
73 SizedBox(height: 20),
74 Center(
75   child: ElevatedButton(
76     //
77     onPressed: () {
78       //Task04: check condition that radius is not empty if so, show snackban
79       //If circle radio is chosen route to page CircleResultScreen and pass radius to its constructor using Navigator.push()
80       //If Sphere radio is chosen route to page SphereResultScreen and pass radius to its constructor usign Navigator.push()
81       double? radius = double.tryParse(_radiuscontroller.text);
82       if (_selectedOption == 'Circle'){
83         Navigator.push(context,MaterialPageRoute(builder: (context)=> CircleResultScreen(radius: radius!)),);
84       } else{
85         Navigator.push(context,MaterialPageRoute(builder: (context)=> SphereResultScreen(radius: radius!)),);
86       }
87     },
88     child: Text('Calculate'),
89   ),
90 ],
91 ),
92 ),
93 }
```

EXPLORER

Q2\_6687049

pages2

circle\_result\_screen.dart

home\_screen.dart

sphere\_result\_screen.dart

main2.dart

sphere\_result\_screen.dart 9+ X

pages2 > sphere\_result\_screen.dart > ...

```
1 import 'package:flutter/material.dart';
2 // Sphere Result Page
3 class SphereResultScreen extends StatelessWidget {
4   final double radius;
5
6   SphereResultScreen({required this.radius});
7
8   @override
9   Widget build(BuildContext context) {
10    //Task5: compute surfaceArea and volume of sphere
11    double surfaceArea = 4 * 3.14 * radius * radius;
12    double volume = (4/3) * 3.14 * radius * radius * radius;
13
14    return Scaffold(
15      appBar: AppBar(title: Text('Sphere Result')),
16      body: Center(
17        child: Column(
18          mainAxisAlignment: MainAxisAlignment.center,
19          children: [
20            Text('Radius: $radius', style: TextStyle(fontSize: 18)),
21            Text('Surface Area: ${surfaceArea.toStringAsFixed(2)}', style: TextStyle(fontSize: 18)),
22            Text('Volume: ${volume.toStringAsFixed(2)}', style: TextStyle(fontSize: 18)),
23            SizedBox(height: 20),
24            ElevatedButton(
25              onPressed: () {
26                Navigator.pop(context);
27                //Task06: When click back using Navigator.pop to back to home screen
28              },
29              child: Text('Back'),
30            ),
31          ],
32        ),
33      ),
34    );
35  }
36 }
```

## L10-Q1 อนิเมชัน

```
C: > Users > minec > Downloads > lab010_q1_6687049 > main.dart > _ImplicitAnimationScreenState > build

Run | Debug
4 void main() {
5   runApp(MyApp());
6 }
7
8 // สร้าง widget หลักของแอป
9 class MyApp extends StatelessWidget {
10   @override
11   Widget build(BuildContext context) {
12     return MaterialApp(
13       debugShowCheckedModeBanner: false, // ซ่อนแถบ debug
14       home: ImplicitAnimationScreen(), // เรียกหน้าหลัก
15     );
16   }
17 }
18
19 // StatefulWidget เพราะเราจะมีการเปลี่ยนค่าภายในหน้า
20 class ImplicitAnimationScreen extends StatefulWidget {
21   @override
22   _ImplicitAnimationScreenState createState() =>
23     _ImplicitAnimationScreenState();
24 }
25
26 class _ImplicitAnimationScreenState extends State<ImplicitAnimationScreen> {
27   final TextEditingController _controller = TextEditingController(); // สำหรับรับค่าจาก TextField
28   int _displayedNumber = 0; // ค่าที่จะแสดงบนหน้าจอ
29
30   // ฟังก์ชันกำหนดลักษณะ transition ของ AnimatedSwitcher
31   // ค่าเริ่มต้นเป็น ScaleTransition
32   Widget Function(Widget, Animation<double>) _transitionFunction =
33     (child, animation) => ScaleTransition(scale: animation, child: child);
34
35   @override
36   Widget build(BuildContext context) {
37     return Scaffold(
38       appBar: AppBar(title: Text("Implicit Animation Effects")),
39       body: Center(
40         child: Column(
41           mainAxisAlignment: MainAxisAlignment.center,
42           children: [
43             // ใช้ AnimatedSwitcher เพื่อเปลี่ยนค่าเลขพร้อมอนิเมชัน
44             AnimatedSwitcher(
45               duration: Duration(milliseconds: 500), // ความเร็วอนิเมชัน
46               transitionBuilder: _transitionFunction, // ฟังก์ชันสำหรับแสดงเอฟเฟกต์
47               child: Text(
48                 '$_displayedNumber',
49                 key: ValueKey<int>(_displayedNumber), // ใช้ key เพื่อให้รู้ว่า widget เปลี่ยนจริง
50                 style: TextStyle(fontSize: 50, fontWeight: FontWeight.bold),
```



```

50         style: TextStyle(fontSize: 50, fontWeight: FontWeight.bold),
51     ),
52 ),
53 SizedBox(height: 20),
54 // ช่องใส่ตัวเลข
55 Padding(
56     padding: const EdgeInsets.symmetric(horizontal: 20),
57     child: TextField(
58         controller: _controller, // ควบคุมค่า Text
59         keyboardType: TextInputType.number, // รับเฉพาะตัวเลข
60         decoration: InputDecoration(labelText: "Enter a number"),
61     ),
62 ),
63 SizedBox(height: 20),
64 // กลุ่มปุ่มเปลี่ยนเอฟเฟกต์
65 Wrap(
66     spacing: 10,
67     runSpacing: 10,
68     alignment: WrapAlignment.center,
69     children: [
70         // ปุ่ม: Scale
71         ElevatedButton(
72             onPressed: () {
73                 setState(() {
74                     _displayedNumber = int.parse(_controller.text); // อัปเดตตัวเลข
75                     _transitionFunction = (child, animation) =>
76                         ScaleTransition(scale: animation, child: child);
77                 });
78             },
79             child: Text("Scale Animation"),
80         ),
81         // ปุ่ม: Fade
82         ElevatedButton(
83             onPressed: () {
84                 setState(() {
85                     _displayedNumber = int.parse(_controller.text);
86                     _transitionFunction = (child, animation) =>
87                         FadeTransition(opacity: animation, child: child);
88                 });
89             },
90             child: Text("Fade Animation"),
91         ),

```

```

91 ),
92 // ปุ่ม: Slide (เลื่อนขึ้นจากด้านล่าง)
93 ElevatedButton(
94   onPressed: () {
95     setState(() {
96       _displayedNumber = int.parse(_controller.text);
97       _transitionFunction =
98         (child, animation) => SlideTransition(
99           position: Tween<Offset>(
100             begin: Offset(0, 1), // เริ่มจากล่าง
101             end: Offset(0, 0),   // ไปตำแหน่งเดิม
102           ).animate(animation),
103           child: child,
104         );
105     });
106   },
107   child: Text("Slide Animation"),
108 ),
109 // ปุ่ม: Rotate (หมุน)
110 ElevatedButton(
111   onPressed: () {
112     setState(() {
113       _displayedNumber = int.parse(_controller.text);
114       _transitionFunction = (child, animation) =>
115         RotationTransition(turns: animation, child: child);
116     });
117   },
118   child: Text("Rotate Animation"),
119 ),
120 ],
121 ),
122 ],
123 ),
124 ),
125 );
126 }
127 }
128

```

## ⚙️ รายละเอียดของอนิเมชันแต่ละแบบ

### ✅ 1. ScaleTransition (Scale Animation)

dart

🗒 คัดลอก

✍ แก้ไข

```
ScaleTransition(scale: animation, child: child);
```

- ขยาย/ย่อ Widget ตามค่า `animation`
- ค่า `animation` อยู่ในช่วง `0.0 → 1.0`
- ใช้สำหรับให้ widget ค่อยๆ โตขึ้นจากขนาด `0 → 100%` (หรือหดลง)

ผลลัพธ์: ตัวเลขจะ "ซูมเข้ามา" หรือ "ซูมออกไป" อย่างนุ่มนวล

### ✅ 2. FadeTransition (Fade Animation)

dart

🗒 คัดลอก

✍ แก้ไข

```
FadeTransition(opacity: animation, child: child);
```

- ค่อยๆ เปลี่ยนค่าความโปร่งใส (`opacity`) ของ widget
- ค่า `animation` บอกระดับความทึบ (`0 = มองไม่เห็น, 1 = มองเห็นเต็มที่`)

ผลลัพธ์: ตัวเลขจะค่อยๆ จางหายไป และตัวใหม่จะค่อยๆ ปรากฏขึ้น

### ✓ 3. SlideTransition (Slide Animation)

dart

🗒 คัดลอก

✍ แก้ไข

```
SlideTransition(  
  position: Tween<Offset>(  
    begin: Offset(0, 1),  
    end: Offset(0, 0),  
  ).animate(animation),  
  child: child,  
);
```

- ขยับตำแหน่ง widget ตามแกน X/Y โดยใช้ค่า `Offset`
- `Offset(0, 1)` หมายถึง เริ่มจาก ล่างขึ้นบน
- `Offset(1, 0)` หมายถึง ขยับจากขวาไปซ้าย

**ผลลัพธ์:** ตัวเลขจะ “เลื่อนขึ้น” เข้ามาจากด้านล่างของหน้าจอ

### ✓ 4. RotationTransition (Rotate Animation)

dart

🗒 คัดลอก

✍ แก้ไข

```
RotationTransition(turns: animation, child: child);
```

- หมุน widget ตามค่า `animation`
- 1.0 = หมุนหนึ่งรอบ (360°)

**ผลลัพธ์:** ตัวเลขจะหมุนเข้ามาหรือหมุนออกไป

## L10-Q2 อนิเมชัน

page2.dart ×

C: > Users > minec > Downloads > lab010\_q2\_6687049 > page2.dart > ...

```
1  import 'package:flutter/material.dart';
2  import 'dart:async'; // สำหรับใช้งาน Timer
3
4  Run | Debug
5  void main() {
6    runApp(MyApp()); // เริ่มต้นแอปโดยเรียกใช้ MyApp
7  }
8  // Widget หลักของแอป
9  class MyApp extends StatelessWidget {
10   @override
11   Widget build(BuildContext context) {
12     return MaterialApp(
13       debugShowCheckedModeBanner: false, // ปิดแถบ debug ที่มุมขวาบน
14       home: Page2(), // ไปยังหน้า Page2
15     );
16   }
17 }
18
19 // สร้าง StatefulWidget สำหรับหน้า Page2
20 class Page2 extends StatefulWidget {
21   @override
22   _Page2State createState() => _Page2State();
23 }
24
25 class _Page2State extends State<Page2> {
26   // ตัวแปรควบคุม TextField
27   TextEditingController _controller = TextEditingController();
28
29   // ข้อความตอบกลับของบอท
30   String _botResponse = "";
31
32   // ข้อความที่จะแสดงบนหน้าจอ (อาจเป็น "Thinking..." หรือตอบจริง)
33   String _displayText = "";
34
35   // ตัวแปรเช็คว่ายู่ในสถานะ "คิด" หรือไม่
36   bool _isThinking = false;
37
38   // นับจำนวนจุดที่แสดงตอนบอทกำลังคิด
39   int _dotCount = 0;
40
41   // จำนวนจุดสูงสุดที่แสดงได้
42   final int _maxDots = 10;
43
44   // ฟังก์ชันเมื่อผู้ใช้กด "Send"
45   void _sendMessage() {
46     String userMessage = _controller.text.trim(); // ตัดช่องว่าง
47     if (userMessage.isEmpty) return; // ถ้าว่างไม่ต้องทำอะไร
```

```

47     if (userMessage.isEmpty) return; // ถ้าว่างไม่ต้องทำอะไร
48
49     // เริ่มต้นสถานะ "คิด"
50     setState(() {
51         _botResponse = "Hello, this is a simulated response!"; // ข้อความจำลองจากบอท
52         _isThinking = true;
53         _displayText = _botResponse; // (ถูกแทนที่หลังด้วย "Thinking...")
54         _dotCount = 0;
55     });
56
57     // แสดง "Thinking..." โดยเพิ่มจุดทุก 500 มิลลิวินาที
58     Timer.periodic(Duration(milliseconds: 500), (timer) {
59         if (!_isThinking) {
60             timer.cancel(); // ถ้าหยุดคิดแล้ว ยกเลิก Timer
61         } else {
62             setState(() {
63                 _dotCount = (_dotCount + 1) % (_maxDots + 1); // วนจุด
64                 _displayText = "Thinking" + "." * _dotCount; // อัปเดตข้อความ
65             });
66         }
67     });
68
69     // ตั้งเวลาให้หยุด "คิด" หลัง 2 วินาที แล้วเริ่มพิมพ์ตอบกลับ
70     Future.delayed(Duration(seconds: 2), () {
71         setState(() {
72             _isThinking = false;
73             _displayText = ""; // เคลียร์ข้อความชั่วคราว
74         });
75         _animateResponse(); // เรียกให้แสดงข้อความตอบกลับทีละตัวอักษร
76     });
77 }
78
79 // แสดงข้อความตอบกลับทีละตัวอักษรแบบพิมพ์ทีละตัว
80 void _animateResponse() {
81     int index = 0;
82
83     Timer.periodic(Duration(milliseconds: 100), (timer) {
84         if (index >= _botResponse.length) {
85             timer.cancel(); // แสดงครบแล้ว ยกเลิก timer
86         } else {
87             setState(() {
88                 _displayText = _botResponse.substring(0, index + 1); // แสดงเพิ่มทีละตัว
89             });

```

```

88     void _animateResponse() {
89     };
90     index++;
91   }
92   });
93 }
94
95 // สร้าง UI หลัก
96 @override
97 Widget build(BuildContext context) {
98   return Scaffold(
99     appBar: AppBar(title: Text("Chat Bot Simulation")), // หัวแอป
100    body: Padding(
101      padding: const EdgeInsets.all(16.0), // เว้นขอบรอบหน้าจอ
102      child: Column(
103        children: [
104          // ส่วนแสดงข้อความบอท
105          Expanded(
106            child: Container(
107              padding: EdgeInsets.all(16.0),
108              alignment: Alignment.centerLeft,
109              child: Text(
110                _displayText, // ข้อความแสดงผล
111                style: TextStyle(fontSize: 20, fontWeight: FontWeight.w500),
112              ),
113            ),
114          ),
115          // ช่องพิมพ์ข้อความ
116          TextField(
117            controller: _controller,
118            decoration: InputDecoration(
119              border: OutlineInputBorder(),
120              labelText: "Type a message", // ชื่อ label
121            ),
122          ),
123          SizedBox(height: 10),
124          // ปุ่มส่งข้อความ
125          ElevatedButton(
126            onPressed: _sendMessage, // เมื่อกดเรียก _sendMessage
127            child: Text("Send"),
128          ),
129        ],
130      ),
131    ),
132  );
133 }
134 }

```

```
16
17 //Task07
18
19 @override
20 void initState() {
21   super.initState();
22   _animationController = AnimationController(
23     vsync: this,
24     duration: Duration(seconds: 2),
25   );
26 }
27
28 //Task08
29 void _startCounting() {
30   int? inputNumber = int.tryParse(_controller.text);
31   if (inputNumber == null || inputNumber < 0) return;
32
33   setState(() {
34     _targetNumber = inputNumber;
35     _displayNumber = _targetNumber;
36   });
37
38   _animation = Tween<double>(begin: 0, end: _targetNumber * 1.0)
39     .animate(_animationController);
40   _animation.addListener(() {
41     setState(() {
42       _displayNumber = _animation.value.round();
43     });
44   });
45   _animationController.reset();
46   _animationController.forward();
47 }
48
```

P1



```

35
36 //task 04
37 PageRouteBuilder _fadeRoute(Widget page) {
38   return PageRouteBuilder(
39     pageBuilder: (context, animation, secondaryAnimation) => page,
40     transitionsBuilder: (context, animation, secondaryAnimation, child) {
41       return FadeTransition(opacity: animation, child: child);
42     },
43   );
44 }
45
46 //task 05
47 PageRouteBuilder _slideRoute(Widget page) {
48   return PageRouteBuilder(
49     pageBuilder: (context, animation, secondaryAnimation) => page,
50     transitionsBuilder: (context, animation, secondaryAnimation, child) {
51       return SlideTransition(
52         position: Tween<Offset>(begin: Offset(1, 0), end: Offset(0, 0))
53           .animate(animation),
54         child: child,
55       );
56     },
57   );
58 }
59
60 //task 06
61 PageRouteBuilder _scaleRoute(Widget page) {
62   return PageRouteBuilder(
63     pageBuilder: (context, animation, secondaryAnimation) => page,
64     transitionsBuilder: (context, animation, secondaryAnimation, child) {
65       return ScaleTransition(scale: animation, child: child);
66     },
67   );
68 }
69

```

h\_p

```

5 PageRouteBuilder _fadeRoute(Widget page) {
6   //Task01
7   return PageRouteBuilder(
8     pageBuilder: (context, animation, secondaryAnimation) => page,
9     transitionsBuilder: (context, animation, secondaryAnimation, child) {
10      return FadeTransition(opacity: animation, child: child);
11    },
12  );
13 }
14

```

L\_p

## L11-Q1 API calls

```
9  class MyApp extends StatelessWidget {
10  @override
11  Widget build(BuildContext context) {
12    return MaterialApp(home: TaskDemoPage());
13  }
14  }
15
16  class TaskDemoPage extends StatefulWidget {
17    const TaskDemoPage({super.key});
18
19    @override
20    State<TaskDemoPage> createState() => _TaskDemoPageState();
21  }
22
23  class _TaskDemoPageState extends State<TaskDemoPage> {
24    String _log = "";
25
26    //Change code at _startAsyncTask() to wait _asyncMethodA() finished before calling _logLine("2=>")
27
28    //เปลี่ยนโค้ดที่ _startAsyncTask() เพื่อให้ _asyncMethodA() เสร็จสิ้นก่อนเรียก _logLine("2=>")
29    Future<void> _startAsyncTask() async {
30      _log = "";
31
32      await _asyncMethodA(); // เพิ่มนี้
33      _logLine("2=>");
34    }
35  }
36
37  Future<void> _asyncMethodA() async {
38    await Future.delayed(Duration(seconds: 2));
39    _logLine("1=>");
40  }
41  }
42
43
44
45  void _logLine(String line) {
46    setState(() {
47      _log += "$line\n";
48    });
49  }
50
51  ..
```

## L11-Q2 API calls

main.dart 9+ x

main.dart > ...

```
1  import 'package:flutter/material.dart';
2  import 'pages/home_page.dart';
3  import 'pages/create_page.dart';
4  import 'pages/read_page.dart';
5  import 'pages/update_page.dart';
6  import 'pages/delete_page.dart';
   Run | Debug
7  void main() {
8    runApp(MyApp());
9  }
10
11 class MyApp extends StatelessWidget {
12   const MyApp({super.key});
13
14   @override
15   Widget build(BuildContext context) {
16     return MaterialApp(
17       title: 'Album CRUD Demo',
18       theme: ThemeData(primarySwatch: Colors.blue),
19       home: HomePage(),
20       routes: {
21         '/create': (context) => CreateAlbumPage(),
22         '/read': (context) => ReadAlbumPage(),
23         '/update': (context) => UpdateAlbumPage(),
24         '/delete': (context) => DeleteAlbumPage(),
25       },
26     );
27   }
28 }
29
```

create\_page.dart 9+ X

pages > create\_page.dart > ...

```
1 import 'dart:convert';
2 import 'package:flutter/material.dart';
3 import 'package:http/http.dart' as http;
4
5 class CreateAlbumPage extends StatefulWidget {
6   @override
7   State<CreateAlbumPage> createState() => _CreateAlbumPageState();
8 }
9
10 class _CreateAlbumPageState extends State<CreateAlbumPage> {
11   final TextEditingController _idController = TextEditingController();
12   final TextEditingController _userIdController = TextEditingController();
13   final TextEditingController _titleController = TextEditingController();
14
15   String _responseText = "";
16
17   Future<void> _createAlbum() async {
18     //task 03
19     final id = _idController.text.trim();
20     final userId = _userIdController.text.trim();
21     final title = _titleController.text.trim();
22
23     if (id.isEmpty || userId.isEmpty || title.isEmpty) {
24       setState(() {
25         _responseText = " ! กรุณากรอกข้อมูลให้ครบถ้วน";
26       });
27       return;
28     }
29
30     //task 04
31     final url = Uri.parse("https://jsonplaceholder.typicode.com/albums");
32
33     Map<String, dynamic> album = {
34       "id": int.parse(id),
35       "userId": int.parse(userId),
36       "title": title,
37     };
38
39     String album_json_string = json.encode(album);
40
41     try {
42       final response = await http.post(
43         url,
44         headers: {"Content-Type": "application/json"},
45         body: album_json_string,
46       ); //task 05
47
48       if (response.statusCode == 201) {
```

```

47
48     if (response.statusCode == 201) {
49         setState(() {
50             _responseText = "✅ Created Album:\n${response.body}";
51         });
52     } else {
53         setState(() {
54             _responseText = "❌ Error: ${response.statusCode}\n${response.body}";
55         });
56     }
57 } catch (e) {
58     setState(() {
59         _responseText = "❌ Exception: $e";
60     });
61 }
62 }
63
64 @override
65 Widget build(BuildContext context) {
66     return Scaffold(
67         appBar: AppBar(title: Text("Create Album")),
68         body: SingleChildScrollView(
69             padding: const EdgeInsets.all(16.0),
70             child: Column(
71                 children: [
72                     TextField(
73                         controller: _idController,
74                         decoration: InputDecoration(labelText: "Album ID"),
75                         keyboardType: TextInputType.number,
76                     ),
77                     SizedBox(height: 10),
78                     TextField(
79                         controller: _userIdController,
80                         decoration: InputDecoration(labelText: "User ID"),
81                         keyboardType: TextInputType.number,
82                     ),
83                     SizedBox(height: 10),
84                     TextField(
85                         controller: _titleController,
86                         decoration: InputDecoration(labelText: "Title"),
87                     ),
88                     SizedBox(height: 20),
89                     ElevatedButton(
90                         onPressed: _createAlbum,

```

```

90         onPressed: _createAlbum,
91         child: Text("Create Album"),
92     ),
93     SizedBox(height: 20),
94     Text(
95         "Response:",
96         style: TextStyle(fontWeight: FontWeight.bold),
97     ),
98     Container(
99         width: double.infinity,
100        padding: EdgeInsets.all(10),
101        color: Colors.grey.shade100,
102        child: SelectableText(_responseText),
103    )
104 ],
105 ),
106 ),
107 );
108 }
109 }
110

```

```

12
13 Future<void> _deleteAlbum() async {
14     //Task07
15     final id = _idController.text.trim();
16
17     if (id.isEmpty) {
18         setState(() {
19             _responseText = " ! กรุณาใส่ ID ของอัลบั้มที่ต้องการลบ";
20         });
21         return;
22     }
23
24     final url = Uri.parse("https://jsonplaceholder.typicode.com/albums/$id");
25
26     try {
27         final response = await http.delete(url); //Task08
28
29         if (response.statusCode == 200) {
30             setState(() {
31                 _responseText = "✅ ลบอัลบั้ม ID $id สำเร็จ (สถานะ: 200)";
32             });
33         } else {
34             setState(() {
35                 _responseText = "❌ ล้มเหลว: ${response.statusCode}";
36             });
37         }
38     } catch (e) {
39         setState(() {
40             _responseText = "❌ ข้อผิดพลาด: $e";
41         });
42     }
43 }

```

```
1
2 import 'package:flutter/material.dart';
3
4 class HomePage extends StatelessWidget {
5   const HomePage({super.key});
6
7   @override
8   Widget build(BuildContext context) {
9     return Scaffold(
10       appBar: AppBar(title: Text('Album CRUD Home')),
11       body: Center(
12         child: Column(
13           mainAxisAlignment: MainAxisAlignment.center,
14           children: [
15             ElevatedButton(
16               onPressed: () => Navigator.pushNamed(context, '/create'),
17               child: Text("Create Album"),
18             ),
19             ElevatedButton(
20               onPressed: () => Navigator.pushNamed(context, '/read'),
21               child: Text("Read Albums"),
22             ),
23             ElevatedButton(
24               onPressed: () => Navigator.pushNamed(context, '/update'),
25               child: Text("Update Album"),
26             ),
27             ElevatedButton(
28               onPressed: () => Navigator.pushNamed(context, '/delete'),
29               child: Text("Delete Album"),
30             ),
31           ],
32         ),
33       ),
34     );
35   }
36 }
37
```



```

14
15 Future<void> _fetchAlbum() async {
16     final id = _controller.text.trim();
17     ;
18     if (id.isEmpty) return;
19
20     final url = Uri.parse("https://jsonplaceholder.typicode.com/albums/$id");
21
22     try {
23         final response = await http.get(url);
24
25         if (response.statusCode == 200) {
26             Map<String, dynamic> decoded = json.decode(response.body);
27             var id = decoded['id'];
28             var userId = decoded['userId'];
29             var title = decoded['title'];
30
31             setState(() {
32                 _rawJson = response.body;
33                 _extractedInfo = "ID: $id \n UserID: $userId \n Title: $title \n";
34             });
35         } else {
36             setState(() {
37                 _rawJson = "Error: ${response.statusCode}";
38                 _extractedInfo = "";
39             });
40         }
41     } catch (e) {
42         setState(() {
43             _rawJson = "Error: $e";
44             _extractedInfo = "";
45         });
46     }
47 }
48

```

```

29
30     Map<String, dynamic> album = {
31         "id": int.parse(id),
32         "userId": int.parse(userId),
33         "title": title,
34     };
35     String album_json_string = json.encode(album);
36
37     //Task06
38     final url = Uri.parse("https://jsonplaceholder.typicode.com/albums/$id");
39
40     try {
41         final response = await http.put(
42             url,
43             headers: {"Content-Type": "application/json"},
44             body: album_json_string,);
45
46         if (response.statusCode == 200) {
47             setState(() {
48                 _responseText = "✅ Updated Album:\n${response.body}";
49             });
50         } else {
51             setState(() {
52                 _responseText = "❌ Error: ${response.statusCode}\n${response.body}";
53             });
54         }
55     } catch (e) {
56         setState(() {
57             _responseText = "❌ Exception: $e";
58         });
59     }
60 }

```

## L11\_ Challenge

C: > Users > minec > Downloads > main.dart > ...

```
1  import 'package:flutter/material.dart';
2  import 'package:http/http.dart' as http;
3  import 'dart:convert'; // for jsonDecode
4
5  Run | Debug
6  void main() {
7    runApp(MyApp());
8  }
9
10 class MyApp extends StatelessWidget {
11   const MyApp({super.key});
12
13   @override
14   Widget build(BuildContext context) {
15     return MaterialApp(
16       title: 'AQI & PM2.5 Forecast',
17       home: ApiFetchPage(),
18     );
19   }
20 }
21
22 class ApiFetchPage extends StatefulWidget {
23   const ApiFetchPage({super.key});
24
25   @override
26   _ApiFetchPageState createState() => _ApiFetchPageState();
27 }
28
29 class _ApiFetchPageState extends State<ApiFetchPage> {
30   String _response = '';
31   bool _loading = false;
32   String _aqi = '';
33   String _forecastText = '';
34
35   // Task02: ทำการเพิ่มรายชื่อเมืองใน list
36   final List<String> _cities = ['bangkok', 'shanghai', 'tokyo', 'london', 'newyork'];
37   String _selectedCity = 'shanghai';
38
39   Future<void> fetchData() async {
40     setState(() {
41       _loading = true;
42       _response = '';
43       _aqi = '';
44       _forecastText = '';
45     });
46
47     await Future.delayed(Duration(seconds: 1)); // Simulated delay
```

```

48 String token = "xxxxxxxxxxxxxxxxxxxxxxxxxxxx"; // Replace with your token
49 String url = "https://api.waqi.info/feed/${_selectedCity}?token=${token}"; // Task03: API URL
50
51 try {
52     http.Response response = await http.get(Uri.parse(url));
53
54     if (response.statusCode == 200) {
55         final jsonData = jsonDecode(response.body);
56         final data = jsonData['data'];
57
58         // Task04: แสดงค่า AQI ปัจจุบัน
59         final aqi = data['aqi'].toString();
60
61         // Task05: แสดงการพยากรณ์ค่า PM2.5 สำหรับ 3 วันถัดไป
62         final pm25List = data['forecast']?['daily']?['pm25'];
63         String forecastText = "";
64         if (pm25List != null && pm25List is List) {
65             for (int i = 0; i < 3 && i < pm25List.length; i++) {
66                 final item = pm25List[i];
67                 final day = item['day'];
68                 final avg = item['avg'];
69                 final min = item['min'];
70                 final max = item['max'];
71                 forecastText += "$day: avg=$avg, min=$min, max=$max\n";
72             }
73         }
74
75         setState(() {
76             _response = const JsonEncoder.withIndent(' ').convert(jsonData);
77             _aqi = "AQI: $aqi";
78             _forecastText = forecastText.trim();
79         });
80     } else {
81         setState(() {
82             _response = 'Error: ${response.statusCode}';
83         });
84     }
85 } catch (e) {
86     setState(() {
87         response = 'Exception: $e';

```

```

87         _response = 'Exception: $e';
88     });
89 } finally {
90     setState(() {
91         _loading = false;
92     });
93 }
94 }
95
96 @override
97 Widget build(BuildContext context) {
98     return Scaffold(
99         appBar: AppBar(
100             title: Text('AQI & PM2.5 Forecast'),
101         ),
102         body: Padding(
103             padding: const EdgeInsets.all(16.0),
104             child: Column(
105                 children: [
106                     // Dropdown for city selection
107                     DropdownButton<String>(
108                         value: _selectedCity,
109                         onChanged: (String? newValue) {
110                             setState(() {
111                                 _selectedCity = newValue!;
112                             });
113                         },
114                         items: _cities.map<DropdownMenuItem<String>>((String city) {
115                             return DropdownMenuItem<String>(
116                                 value: city,
117                                 child: Text(city[0].toUpperCase() + city.substring(1)),
118                             );
119                         }).toList(),
120                     ),
121                     SizedBox(height: 10),
122
123                     // Button with spinner
124                     ElevatedButton(
125                         onPressed: _loading ? null : fetchData,
126                         child: _loading ? RotatingHourglass() : Text('Fetch Data'),
127                     ),
128                     SizedBox(height: 20),
129

```

```

129
130 // AQI display
131 if (_aqi.isNotEmpty) ...[
132   Text(_aqi, style: TextStyle(fontSize: 18, fontWeight: FontWeight.bold)),
133   SizedBox(height: 8),
134 ],
135
136 // Forecast display
137 if (_forecastText.isNotEmpty) ...[
138   Text("PM2.5 Forecast (next 3 days):", style: TextStyle(fontWeight: FontWeight.bold)),
139   Text(_forecastText),
140   SizedBox(height: 20),
141 ],
142
143 // Raw JSON
144 Expanded(
145   child: SingleChildScrollView(
146     child: SelectableText(_response),
147   ),
148 ),
149 ],
150 ),
151 );
152 }
153 }
154 }
155
156 class RotatingHourglass extends StatefulWidget {
157   const RotatingHourglass({super.key});
158
159   @override
160   _RotatingHourglassState createState() => _RotatingHourglassState();
161 }
162
163 class _RotatingHourglassState extends State<RotatingHourglass>
164   with SingleTickerProviderStateMixin {
165   late AnimationController _controller;
166
167   @override
168   void initState() {
169
170     @override
171     void initState() {
172       super.initState();
173       _controller =
174         AnimationController(duration: const Duration(seconds: 1), vsync: this)
175           ..repeat();
176     }
177
178     @override
179     void dispose() {
180       _controller.dispose();
181       super.dispose();
182     }
183
184     @override
185     Widget build(BuildContext context) {
186       return RotationTransition(
187         turns: _controller,
188         child: Icon(Icons.hourglass_top, color: Colors.white),
189       );
190     }
191   }
192 }

```

# L\_12 Flutter and SQLite

## Q\_1

```
28
29 class HomePage extends StatefulWidget {
30   const HomePage({super.key});
31
32   @override
33   State<HomePage> createState() => _HomePageState();
34 }
35
36 class _HomePageState extends State<HomePage> {
37   Database? _db;
38
39   @override
40   void initState() {
41     super.initState();
42     _initDatabase();
43   }
44
45   Future<void> _initDatabase() async {
46     final dbPath = await getDatabasesPath();
47     final path = p.join(dbPath, 'contacts_map.db');
48     _db = await openDatabase(
49       path,
50       version: 1,
51       onCreate: (db, version) async {
52         await db.execute('''
53           CREATE TABLE contacts (
54             id INTEGER PRIMARY KEY AUTOINCREMENT,
55             name TEXT,
56             phone TEXT
57           )
58         ''');
59       },
60     );
61   }
62 }
```

Main

```

13 class _ContactDeletePageState extends State<ContactDeletePage> {
14   final _idController = TextEditingController();
15
16   Future<void> _deleteContact() async {
17     final id = int.tryParse(_idController.text);
18     await widget.database.delete(
19       'contacts',
20       where: 'id = ?',
21       whereArgs: [id],
22     );
23     ScaffoldMessenger.of(context).showSnackBar(
24       const SnackBar(content: Text('ลบข้อมูลเรียบร้อยแล้ว')),
25     );
26     _idController.clear();
27   }
28
29   @override
30   Widget build(BuildContext context) {
31     return Scaffold(
32       appBar: AppBar(title: const Text('ลบผู้ติดต่อ')),
33       body: Padding(
34         padding: const EdgeInsets.all(16.0),
35         child: Column(
36           crossAxisAlignment: CrossAxisAlignment.start,
37           children: [
38             TextField(
39               controller: _idController,
40               decoration: const InputDecoration(labelText: 'ID ที่ต้องการลบ'),
41               keyboardType: TextInputType.number,
42             ),
43             const SizedBox(height: 16),
44             ElevatedButton(
45               onPressed: _deleteContact,
46               child: const Text('ลบ'),
47             ),
48           ],
49         ),
50       ),
51     );
52   }
53 }
54

```

del



contact\_insert\_page.dart 9+ X

pages > contact\_insert\_page.dart > ...

```
1 // pages/contact_insert_page.dart
2 import 'package:flutter/material.dart';
3 import 'package:sqflite/sqflite.dart';
4
5 class ContactInsertPage extends StatefulWidget {
6   final Database database;
7   const ContactInsertPage({super.key, required this.database});
8
9   @override
10  State<ContactInsertPage> createState() => _ContactInsertPageState();
11 }
12
13 class _ContactInsertPageState extends State<ContactInsertPage> {
14   final _nameController = TextEditingController();
15   final _phoneController = TextEditingController();
16
17   Future<void> _insertContact() async {
18     final name = _nameController.text;
19     final phone = _phoneController.text;
20     if (name.isNotEmpty && phone.isNotEmpty) {
21       await widget.database.insert('contacts', {
22         'name': name,
23         'phone': phone,
24       });
25       ScaffoldMessenger.of(context).showSnackBar(
26         const SnackBar(content: Text('บันทึกข้อมูลเรียบร้อยแล้ว')),
27       );
28       _nameController.clear();
29       _phoneController.clear();
30     }
31   }
32 }
```

insert

```

1 // contacts_app_main.dart
2 // Home page that links to each CRUD operation page, with shared database initialization
3
4 import 'package:flutter/material.dart';
5 import 'package:path/path.dart' as p;
6 import 'package:sqflite/sqflite.dart';
7
8 class ContactsSelectPage extends StatefulWidget {
9   final Database database;
10   const ContactsSelectPage({super.key, required this.database});
11
12   @override
13   State<ContactsSelectPage> createState() => _ContactsSelectPageState();
14 }
15
16 class _ContactsSelectPageState extends State<ContactsSelectPage> {
17   List<Map<String, dynamic>> _contacts = [];
18
19   @override
20   void initState() {
21     super.initState();
22     _refreshContacts();
23   }
24
25   Future<void> _refreshContacts() async {
26     final data = await widget.database.query('contacts', orderBy: 'name ASC');
27     setState(() {
28       _contacts = data;
29     });
30   }
31
32   @override
33   Widget build(BuildContext context) {
34     return Scaffold(
35       appBar: AppBar(title: const Text('รายชื่อผู้ติดต่อ (SELECT only)'),
36       body: _contacts.isEmpty
37         ? const Center(child: Text('ไม่มีข้อมูล'))
38         : ListView.builder(
39           itemCount: _contacts.length,
40           itemBuilder: (context, index) {
41             final contact = _contacts[index];
42             return ListTile(
43               leading: CircleAvatar(child: Text('${contact['id']}')),
44               title: Text(contact['name']),
45               subtitle: Text(contact['phone']),
46             );
47           },
48     ),

```

select +

```

1 // pages/contact_update_page.dart
2 import 'package:flutter/material.dart';
3 import 'package:sqflite/sqflite.dart';
4
5 class ContactUpdatePage extends StatefulWidget {
6   final Database database;
7   const ContactUpdatePage({super.key, required this.database});
8
9   @override
10  State<ContactUpdatePage> createState() => _ContactUpdatePageState();
11 }
12
13 class _ContactUpdatePageState extends State<ContactUpdatePage> {
14   final _idController = TextEditingController();
15   final _nameController = TextEditingController();
16   final _phoneController = TextEditingController();
17
18   Future<void> _updateContact() async {
19     final id = int.tryParse(_idController.text);
20     final name = _nameController.text;
21     final phone = _phoneController.text;
22     await widget.database.update(
23       'contacts',
24       {'name': name, 'phone': phone},
25       where: 'id = ?',
26       whereArgs: [id],
27     );
28     ScaffoldMessenger.of(context).showSnackBar(
29       const SnackBar(content: Text('อัปเดตข้อมูลเรียบร้อยแล้ว')),
30     );
31     _idController.clear();
32     _nameController.clear();
33     _phoneController.clear();
34   }
35
36   @override

```

update

## Q\_2

```
class HomePage extends StatefulWidget {
  const HomePage({super.key});

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  Database? _db;

  @override
  void initState() {
    super.initState();
    _initDatabase();
  }

  Future<void> _initDatabase() async {
    final dbPath = await getDatabasesPath();
    final path = p.join(dbPath, 'contacts_map.db');
    _db = await openDatabase(
      path,
      version: 1,
      onCreate: (db, version) async {
        await db.execute('''
          CREATE TABLE contacts (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            name TEXT,
            phone TEXT
          )
        ''');
      },
    );
  }
}
```

main



```
class _ContactDeletePageState extends State<ContactDeletePage> {
  final _idController = TextEditingController();

  Future<void> _deleteContact() async {
    final id = int.tryParse(_idController.text);
    if (id == null) return;
    final count = await widget.database.delete(
      'contacts',
      where: 'id = ?',
      whereArgs: [id],
    );
    if (count == 0) {
      ScaffoldMessenger.of(
        context,
      ).showSnackBar(SnackBar(content: Text('ไม่พบผู้ติดต่อที่มี ID \${id}')));
    } else {
      ScaffoldMessenger.of(
        context,
      ).showSnackBar(const SnackBar(content: Text('ลบข้อมูลเรียบร้อยแล้ว')));
      _idController.clear();
    }
  }
}
```

del

pages > contact\_insert\_page.dart

```
1 // pages/contact_insert_page.dart
2 import 'package:flutter/material.dart';
3 import 'package:sqflite/sqflite.dart';
4 import '../models/contact.dart';
5
6 class ContactInsertPage extends StatefulWidget {
7   final Database database;
8   const ContactInsertPage({super.key, required this.database});
9
10  @override
11  State<ContactInsertPage> createState() => _ContactInsertPageState();
12 }
13
14 class _ContactInsertPageState extends State<ContactInsertPage> {
15   final _nameController = TextEditingController();
16   final _phoneController = TextEditingController();
17
18   Future<void> _insertContact() async {
19     final name = _nameController.text;
20     final phone = _phoneController.text;
21     if (name.isNotEmpty && phone.isNotEmpty) {
22       final contact = Contact(name: name, phone: phone);
23       await widget.database.insert('contacts', contact.toMap());
24       ScaffoldMessenger.of(context).showSnackBar(
25         const SnackBar(content: Text('บันทึกข้อมูลเรียบร้อยแล้ว')),
26       );
27       _nameController.clear();
28       _phoneController.clear();
29     }
30   }
31 }
```

insert

pages > contact\_select\_page.dart

```
1 // contacts_app_main.dart
2 // Home page that links to each CRUD operation page, with shared database initialization
3
4 import 'package:flutter/material.dart';
5 import 'package:path/path.dart' as p;
6 import 'package:sqflite/sqflite.dart';
7 import '../models/contact.dart';
8
9 class ContactsSelectPage extends StatefulWidget {
10   final Database database;
11   const ContactsSelectPage({super.key, required this.database});
12
13   @override
14   State<ContactsSelectPage> createState() => _ContactsSelectPageState();
15 }
16
17 class _ContactsSelectPageState extends State<ContactsSelectPage> {
18   List<Contact> _contacts = [];
19
20   @override
21   void initState() {
22     super.initState();
23     _refreshContacts();
24   }
25
26   Future<void> _refreshContacts() async {
27     final List<Map<String, dynamic>> data = await widget.database.query(
28       'contacts',
29       orderBy: 'name ASC',
30     );
31     final List<Contact> loadedContacts = [];
32     for (var map in data) {
33       loadedContacts.add(
34         Contact(id: map['id'], name: map['name'], phone: map['phone']),
35       );
36     }
37     setState(() {
38       _contacts = loadedContacts;
39     });
40   }
41 }
```

Select

pages > contact\_update\_page.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:sqflite/sqflite.dart';
3 import '../models/contact.dart';
4
5 class ContactUpdatePage extends StatefulWidget {
6   final Database database;
7   const ContactUpdatePage({super.key, required this.database});
8
9   @override
10  State<ContactUpdatePage> createState() => _ContactUpdatePageState();
11 }
12
13 class _ContactUpdatePageState extends State<ContactUpdatePage> {
14   final _idController = TextEditingController();
15   final _nameController = TextEditingController();
16   final _phoneController = TextEditingController();
17
18   Future<void> _updateContact() async {
19     final id = int.tryParse(_idController.text);
20     final name = _nameController.text;
21     final phone = _phoneController.text;
22     final contact = Contact(id: id, name: name, phone: phone);
23     final count = await widget.database.update(
24       'contacts',
25       contact.toMap(),
26       where: 'id = ?',
27       whereArgs: [contact.id],
28     );
29     if (count == 0) {
30       ScaffoldMessenger.of(
31         context,
32       ).showSnackBar(SnackBar(content: Text('ไม่พบผู้ติดต่อที่มี ID \${id}')));
33     } else {
34       ScaffoldMessenger.of(context).showSnackBar(
35         const SnackBar(content: Text('อัปเดตข้อมูลเรียบร้อยแล้ว')),
36       );
37     }
38   }
39 }
```

update

## Q\_3

```
28
29 class HomePage extends StatefulWidget {
30   const HomePage({super.key});
31
32   @override
33   State<HomePage> createState() => _HomePageState();
34 }
35
36 class _HomePageState extends State<HomePage> {
37   Database? _db;
38
39   @override
40   void initState() {
41     super.initState();
42     _initDatabase();
43   }
44
45   Future<void> _initDatabase() async {
46     final dbPath = await getDatabasesPath();
47     final path = p.join(dbPath, 'contacts_map.db');
48     _db = await openDatabase(
49       path,
50       version: 1,
51       onCreate: (db, version) async {
52         await db.execute('''
53           CREATE TABLE contacts (
54             id INTEGER PRIMARY KEY AUTOINCREMENT,
55             name TEXT,
56             phone TEXT
57           )
58         ''');
59       },
60     );
61   }
62 }
```

Main

pages > contact\_delete\_page.dart

```
1 // pages/contact_delete_page.dart
2 import 'package:flutter/material.dart';
3 import 'package:sqflite/sqflite.dart';
4
5 class ContactDeletePage extends StatefulWidget {
6   final Database database;
7   const ContactDeletePage({super.key, required this.database});
8
9   @override
10  State<ContactDeletePage> createState() => _ContactDeletePageState();
11 }
12
13 class _ContactDeletePageState extends State<ContactDeletePage> {
14   final _idController = TextEditingController();
15
16   Future<void> _deleteContact() async {
17     final id = int.tryParse(_idController.text);
18     if (id == null) return;
19     final count = await widget.database.rawDelete(
20       'DELETE FROM contacts WHERE id = ?',
21       [id],
22     );
23     if (count == 0) {
24       ScaffoldMessenger.of(
25         context,
26       ).showSnackBar(SnackBar(content: Text('ไม่พบผู้ติดต่อที่มี ID \${id}')));
27     } else {
28       ScaffoldMessenger.of(
29         context,
30       ).showSnackBar(const SnackBar(content: Text('ลบข้อมูลเรียบร้อยแล้ว')));
31       _idController.clear();
32     }
33   }
34 }
```

del

pages > contact\_insert\_page.dart

```
1 // pages/contact_insert_page.dart
2 import 'package:flutter/material.dart';
3 import 'package:sqflite/sqflite.dart';
4
5 class ContactInsertPage extends StatefulWidget {
6   final Database database;
7   const ContactInsertPage({super.key, required this.database});
8
9   @override
10  State<ContactInsertPage> createState() => _ContactInsertPageState();
11 }
12
13 class _ContactInsertPageState extends State<ContactInsertPage> {
14   final _nameController = TextEditingController();
15   final _phoneController = TextEditingController();
16
17   Future<void> _insertContact() async {
18     final name = _nameController.text;
19     final phone = _phoneController.text;
20     if (name.isNotEmpty && phone.isNotEmpty) {
21       await widget.database.rawInsert(
22         'INSERT INTO contacts (name, phone) VALUES (?, ?)',
23         [name, phone],
24       );
25       ScaffoldMessenger.of(context).showSnackBar(
26         const SnackBar(content: Text('บันทึกข้อมูลเรียบร้อยแล้ว')),
27       );
28       _nameController.clear();
29       _phoneController.clear();
30     }
31   }
32 }
```

insert

```
4 import 'package:flutter/material.dart';
5 import 'package:path/path.dart' as p;
6 import 'package:sqflite/sqflite.dart';
7
8 class ContactsSelectPage extends StatefulWidget {
9   final Database database;
10  const ContactsSelectPage({super.key, required this.database});
11
12  @override
13  State<ContactsSelectPage> createState() => _ContactsSelectPageState();
14 }
15
16 class _ContactsSelectPageState extends State<ContactsSelectPage> {
17   List<Map<String, dynamic>> _contacts = [];
18
19   @override
20   void initState() {
21     super.initState();
22     _refreshContacts();
23   }
24
25   Future<void> _refreshContacts() async {
26     final data = await widget.database.rawQuery(
27       'SELECT * FROM contacts ORDER BY name ASC',
28     );
29     setState(() {
30       _contacts = data;
31     });
32   }
33 }
```

select



pages >  contact\_update\_page.dart

```
1 // pages/contact_update_page.dart
2 import 'package:flutter/material.dart';
3 import 'package:sqflite/sqflite.dart';
4
5 class ContactUpdatePage extends StatefulWidget {
6   final Database database;
7   const ContactUpdatePage({super.key, required this.database});
8
9   @override
10  State<ContactUpdatePage> createState() => _ContactUpdatePageState();
11 }
12
13 class _ContactUpdatePageState extends State<ContactUpdatePage> {
14   final _idController = TextEditingController();
15   final _nameController = TextEditingController();
16   final _phoneController = TextEditingController();
17
18   Future<void> _updateContact() async {
19     final id = int.tryParse(_idController.text);
20     final name = _nameController.text;
21     final phone = _phoneController.text;
22     if (id == null || name.isEmpty || phone.isEmpty) return;
23     final count = await widget.database.rawUpdate(
24       'UPDATE contacts SET name = ?, phone = ? WHERE id = ?',
25       [name, phone, id],
26     );
27     if (count == 0) {
28       ScaffoldMessenger.of(
29         context,
30       ).showSnackBar(SnackBar(content: Text('ไม่พบผู้ติดต่อที่มี ID \${id}')));
31     } else {
32       ScaffoldMessenger.of(context).showSnackBar(
33         const SnackBar(content: Text('อัปเดตข้อมูลเรียบร้อยแล้ว')),
34       );
35       _idController.clear();
36       _nameController.clear();
37       _phoneController.clear();
38     }
39   }
40 }
```

Update