

Flutter Widget Demo – อธิบายโค้ดแบบละเอียดทีละบรรทัด (เพิ่ม **SnackBar**)

เอกสารนี้อธิบายโค้ด Flutter หน้า **PageMain** แบบละเอียดทีละบรรทัด และมีการเพิ่ม **SnackBar** (Material Widget) เพื่อแจ้งเตือน/แจ้งผลการบันทึก รูปแบบเอกสารเป็น **Markdown (.md)** เปิดใน **VS Code** ได้ทันที

Import Library

```
import 'package:flutter/material.dart';
```

- นำเข้าไลบรารี `material.dart` - ใช้สำหรับ Material Widget เช่น `Scaffold`, `AppBar`, `TextFormField`, `ElevatedButton`, `Card`, `SnackBar`

StatefulWidget: PageMain

```
class PageMain extends StatefulWidget {
```

- สร้างหน้าจอชื่อ `PageMain` - ใช้ `StatefulWidget` เพราะมีข้อมูลที่สามารถเปลี่ยนได้ (พลัฟฟอร์มเปลี่ยนตามที่รองรับ)

```
const PageMain({super.key});
```

- constructor ของ widget - `super.key` ใช้ส่ง key ไปยังคลาสแม่

```
@override
State<PageMain> createState() => _PageMainState();
}
```

- สร้าง state ที่ใช้ควบคุมหน้าจอ คือ `_PageMainState`

State Class: `_PageMainState`

```
class _PageMainState extends State<PageMain> {
```

- เก็บ logic และ UI ของหน้าจอ

```
TextEditingController input = TextEditingController();
```

- ใช้ควบคุมและอ่านค่าจากช่องกรอกข้อความ

```
String result = "";
```

- ตัวแปรเก็บข้อความผลลัพธ์ เพื่อแสดงใน `Card`

build() Method

```
@override  
Widget build(BuildContext context) {
```

- Flutter เรียกเมธอดนี้เพื่อสร้าง UI

Scaffold (โครงหน้าจอ)

```
return Scaffold(  
  
```

- โครงหลักของหน้าจอ Material

```
    backgroundColor: Color.fromARGB(255, 250, 250, 250),  
  
```

- สีพื้นหลังของหน้าจอทั้งหมด (เทาอ่อนมาก)

AppBar

```
    appBar: AppBar(  
  
```

- แถบด้านบนของแอป

```
    title: Text("Wiget Demo"),  
  
```

- ข้อความชื่อแอปบน AppBar

```
centerTitle: true,
```

- ให้ title อยู่ตรงกลาง

```
backgroundColor: Color.fromARGB(255, 230, 230, 230),
```

- สีพื้นหลังของ AppBar

```
foregroundColor: Color.fromARGB(255, 60, 60, 60),
```

- สีของตัวหนังสือและไอคอนบน AppBar

```
elevation: 0,
```

- ไม่แสดงเงาของ AppBar

Body + Padding + Column

```
body: Padding(  
  padding: EdgeInsets.all(16),
```

- เว้นระยะรอบเนื้อหาทั้งหมด 16px

```
child: Column(  
  children: [
```

- ใช้ `Column` เรียง widget จากบนลงล่าง

ข้องกรอกข้อความ: TextFormField

```
Container(  
  margin: EdgeInsets.fromLTRB(0, 10, 0, 15),
```

- ใช้ `Container` เพื่อเว้นระยะห่างภายนอก (margin) - เว้นด้านบน 10px และด้านล่าง 15px

```
child: TextFormField(  
  controller: input,
```

- ช่องกรอกข้อความ - ใช้ `controller: input` เพื่ออ่าน/ควบคุมค่าที่พิมพ์

```
onChanged: (value) {  
  print("พิมพ์ข้อความ: $value");  
},
```

- ทำงานทุกครั้งที่ใช้พิมพ์ - `print(...)` แสดงค่าที่พิมพ์ใน Debug Console

```
decoration: InputDecoration(  

```

- ตั้งค่าการตกแต่งช่องกรอก

```
hintText: "กรอกข้อความ",
```

- ข้อความแนะนำในช่องกรอก

```
filled: true,  
fillColor: Color.fromARGB(255, 240, 240, 240),
```

- เติมพื้นหลังของช่องกรอกด้วยสีเทาอ่อน

```
contentPadding: EdgeInsets.fromLTRB(20, 12, 10, 12),
```

- ระยะห่างภายในช่องกรอก (padding ภายใน)

```
prefixIcon: Icon(  
  Icons.edit,  
  color: Color.fromARGB(255, 120, 120, 120),  
),
```

- แสดงไอคอนด้านหน้าช่องกรอก พร้อมกำหนดสีไอคอน

```
border: OutlineInputBorder(  
  borderRadius: BorderRadius.circular(25),  
  borderSide: BorderSide.none,  
),
```

- ขอบมน 25 และไม่แสดงเส้นขอบ

ปุ่มบันทึก: `ElevatedButton`

```
Container(  
  margin: EdgeInsets.only(bottom: 15),
```

- เว้นระยะห่างด้านล่างของปุ่ม 15px

```
child: ElevatedButton.icon(  
  icon: Icon(Icons.add),
```

- ปุ่ม Material แบบมีไอคอนและข้อความ

```
style: ElevatedButton.styleFrom(  
  backgroundColor: Color.fromARGB(255, 200, 200, 200),
```

- สีพื้นหลังของปุ่ม

```
foregroundColor: Color.fromARGB(255, 60, 60, 60),
```

- สีของข้อความและไอคอนบนปุ่ม

```
shape: RoundedRectangleBorder(  
  borderRadius: BorderRadius.circular(25),  
),
```

- ทำปุ่มให้ขอบมน

```
padding: EdgeInsets.symmetric(horizontal: 30, vertical: 12),
```

- เพิ่มพื้นที่ภายในปุ่ม: แนวนอน 30px แนวตั้ง 12px



onPressed: ตรวจสอบค่าว่าง + บันทึก + แสดง Snackbar

```
onPressed: () {
```

- ไล่ดูในส่วนนี้ทำงานเมื่อผู้ใช้กดปุ่ม "บันทึก"

```
if (input.text.trim().isEmpty) {  
  print("ยังไม่ได้กรอกข้อความ");  
}
```

- ตรวจสอบว่าข้อความว่างหรือไม่ (ตัดช่องว่างหัวท้ายด้วย `trim()`) - ถ้าว่าง ให้แสดงข้อความใน Debug Console

```
ScaffoldMessenger.of(context).showSnackBar(
  SnackBar(content: Text("กรุณารอก่อนบันทึก")),
);
```

- แสดง **SnackBar** แจ้งเตือนว่าต้องกรอกก่อน - `ScaffoldMessenger.of(context)` ใช้เรียก SnackBar บนหน้าจอปัจจุบัน

```
return;
}
```

- `return` เพื่อหยุดการทำงาน ไม่ให้บันทึกค่าตอนที่ว่าง

```
setState(() {
  result = input.text;
});
```

- นำค่าจากช่องกรอกไปเก็บใน `result` - ใช้ `setState()` เพื่อให้ UI อัปเดต (Card แสดงผลลัพธ์จะเปลี่ยน)

```
print("บันทึกคำตอบ: ${input.text}");
print("ผลลัพธ์ที่แสดง: $result");
```

- แสดง debug ตอนบันทึก - บรรทัดแรก: แสดงค่าที่ผู้ใช้กรอก - บรรทัดสอง: แสดงค่าที่นำไปแสดงผลบนจอ

```
ScaffoldMessenger.of(context).showSnackBar(
  SnackBar(content: Text("บันทึกข้อมูลเรียบร้อยแล้ว")),
);
```

- แสดง SnackBar แจ้งว่าบันทึกสำเร็จ

```
input.clear();
```

- ล้างข้อความในช่องกรอกหลังบันทึก



ไอคอนและข้อความบนปุ่ม

```
icon: Icon(Icons.save),
label: Text("บันทึก"),
```

- `icon` แสดงไอคอนรูปบันทึก - `label` แสดงข้อความ "บันทึก"

Card แสดงผลลัพธ์

```
Card(  
  color: Color.fromARGB(255, 245, 245, 245),
```

- กล่องแสดงผล - กำหนดสีพื้นหลังของ Card

```
  shape: RoundedRectangleBorder(  
    borderRadius: BorderRadius.circular(20),  
  ),
```

- ทำขอบ Card ให้โค้งมน

```
  elevation: 1,
```

- เพิ่มเงาเล็กน้อย

```
  child: Padding(  
    padding: EdgeInsets.all(14),
```

- เว้นระยะห่างภายใน Card

```
    child: Row(  
      children: [
```

- จัดไอคอนและข้อความให้เรียงแนวนอน

```
        Icon(  
          Icons.bookmark,  
          color: Color.fromARGB(255, 100, 100, 100),  
        ),
```

- ไอคอน bookmark พร้อมกำหนดสี

```
        SizedBox(width: 8),
```

- เว้นช่องว่าง 8px ระหว่างไอคอนกับข้อความ

```
        Text(  
          "ผลลัพธ์: $result",
```

```
style: TextStyle(color: Color.fromARGB(255, 50, 50, 50)),
),
```

- แสดงผลลัพธ์ที่บันทึกไว้ในตัวแปร `result` - กำหนดสีข้อความ

สรุป Material Widgets (ตรงตามโจทย์ 5 ตัว)

โค้ดนี้ใช้ Material Widgets ตามหน้า `docs.flutter.dev/ui/widgets/material` จำนวน **5 ตัว**: 1. **AppBar** – แถบด้านบนของแอป 2. **TextFormField** – ช่องกรอกข้อความ 3. **ElevatedButton** – ปุ่มบันทึก 4. **Card** – กล่องแสดงผล 5. **SnackBar** – แจ้งเตือน/แจ้งผลการบันทึก

สรุปการทำงานของโปรแกรม

- ผู้ใช้พิมพ์ข้อความ → แสดงใน Debug Console ผ่าน `onChanged`
- กดปุ่มบันทึก:
- ถ้าว่าง → แสดง SnackBar “กรุณากรอกข้อความก่อนบันทึก”
- ถ้าไม่ว่าง → เซฟค่าไว้ใน `result` และแสดง SnackBar “บันทึกข้อมูลเรียบร้อยแล้ว”
- ล้างช่องกรอกหลังบันทึก
- Card จะแสดงข้อความล่าสุดในรูปแบบ ผลลัพธ์: ...