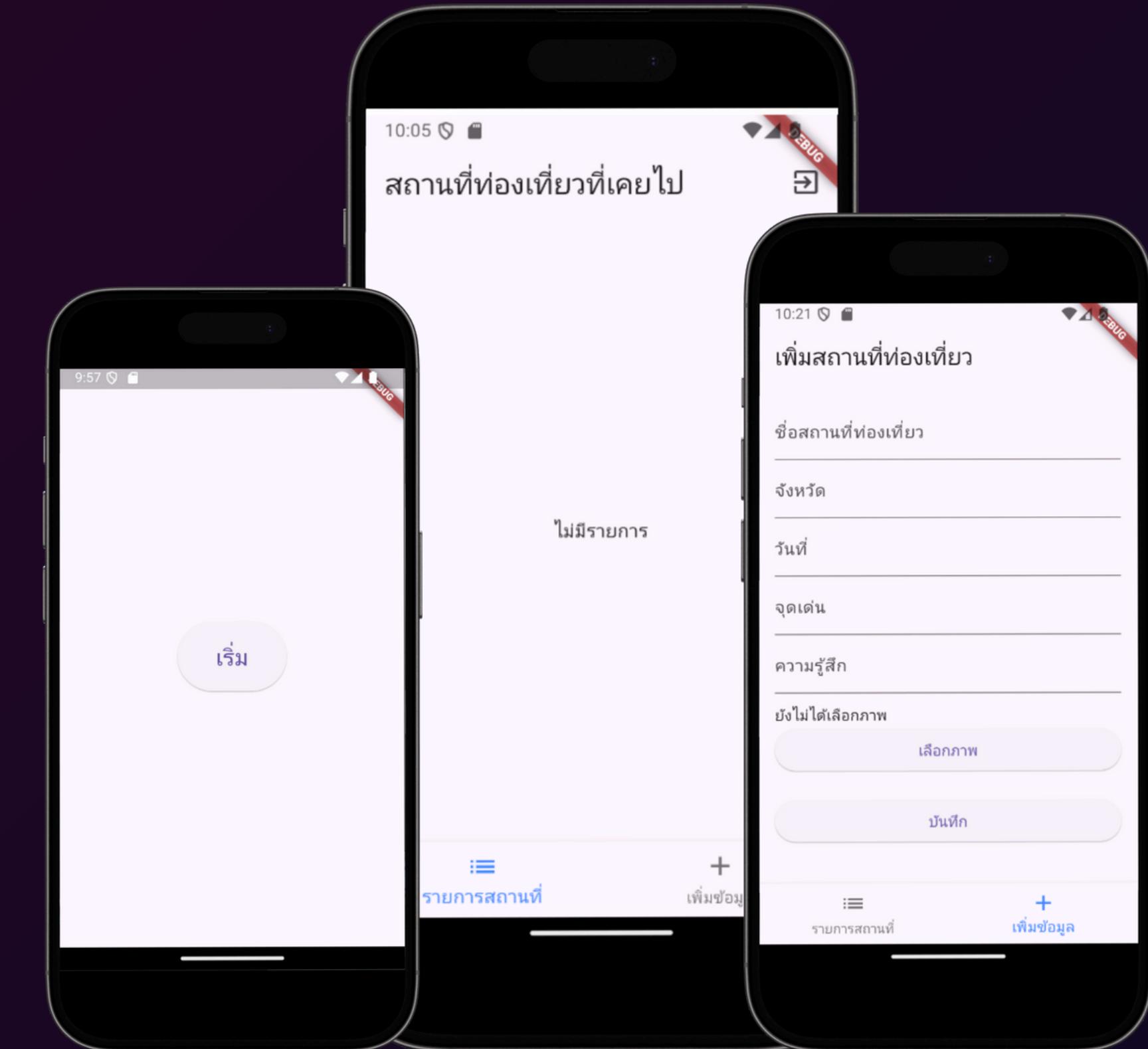
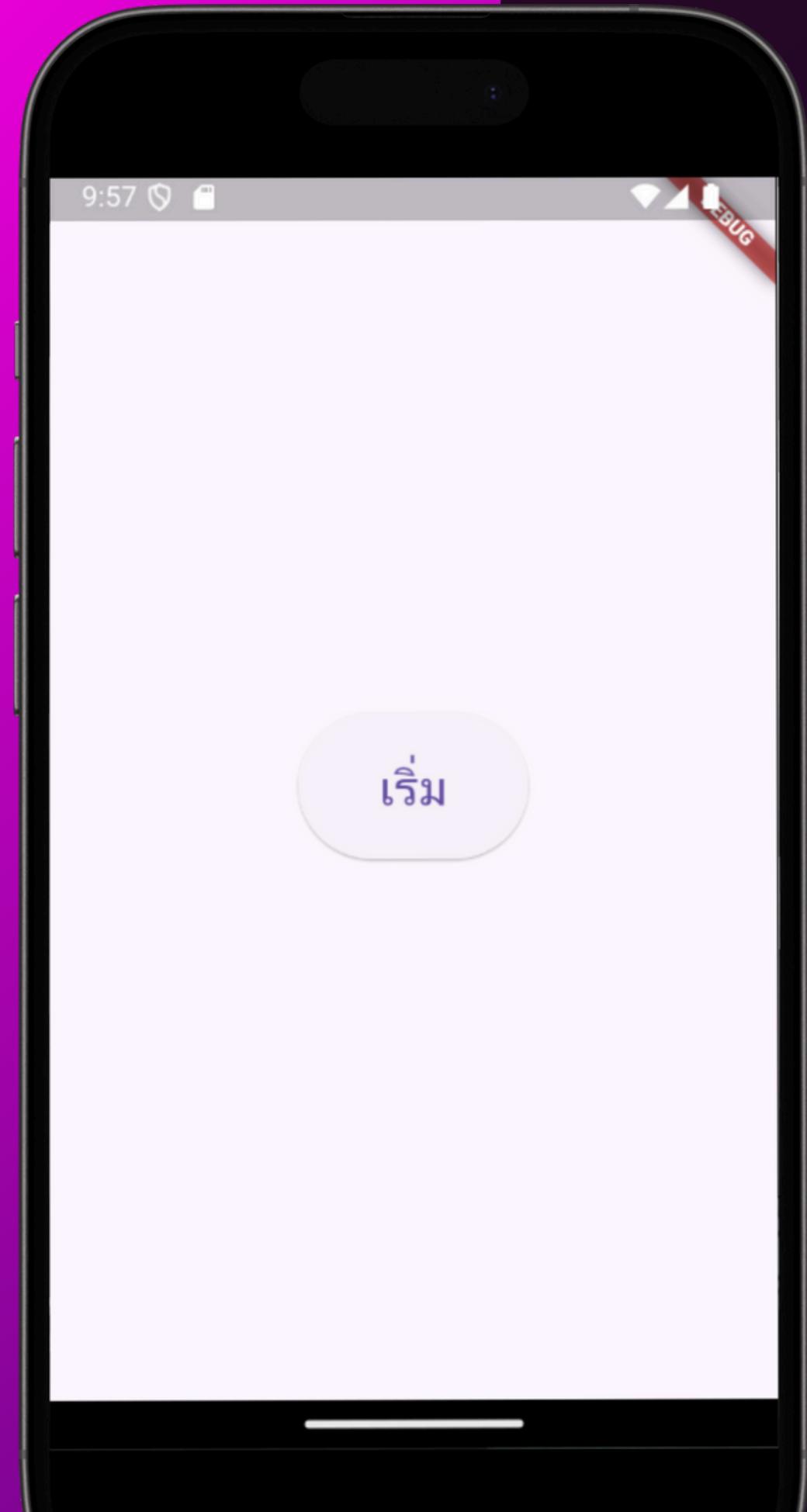


# บันทึก...

## สถานที่ท่องเที่ยว ที่เคยไป

MOBILE APP





# Application นี้เกี่ยวกับอะไร?

Application นี้เกี่ยวกับการบันทึกความทรงจำให้ feel เหมือนเขียน diary  
แต่เป็นการจดบันทึกสถานที่ท่องเที่ยวที่เคยไปซึ่งมี ชื่อสถานที่ ลังหมัด วันที่  
จุดเด่นของสถานที่นั้น และ ความรู้สึกที่ได้ไปเที่ยวที่สถานที่นั้นซึ่งตัว  
application สามารถเข้าไปดู แก้ไข หรือลบ ข้อมูลได้

# Application นี้ใช้งานยังไง?

01

เมื่อคุณเปิดแอปจะขึ้นหน้าจอชี้แจงว่า  
ปุ่มกดตรงกลางเพื่อเริ่มต้นใช้งาน

02

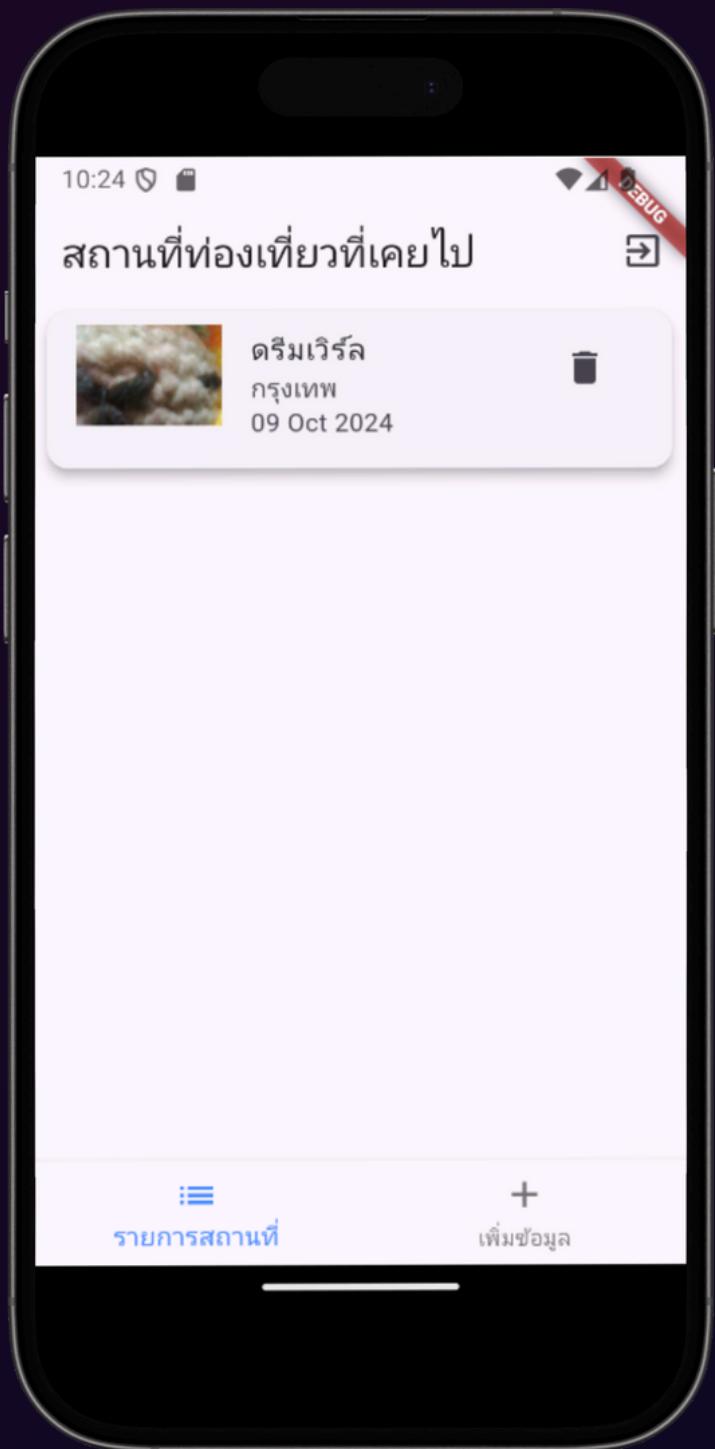
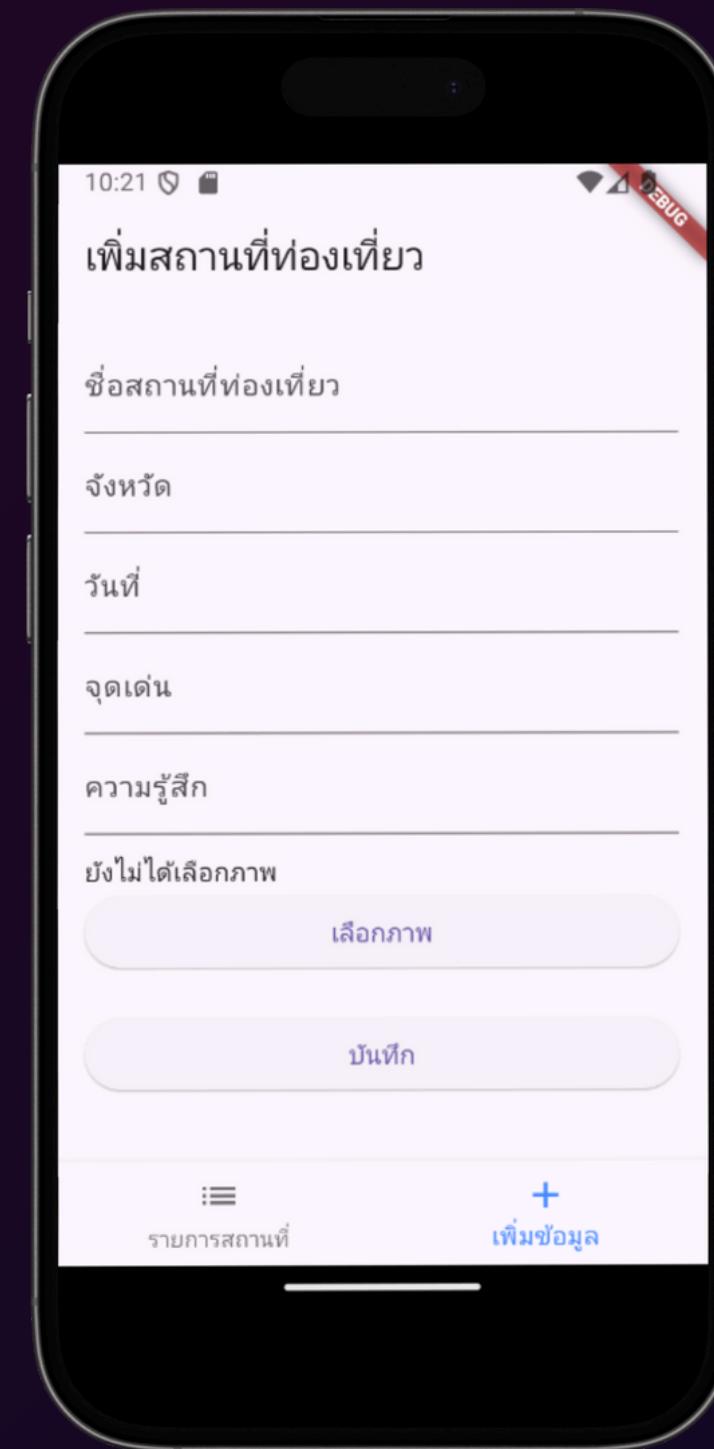
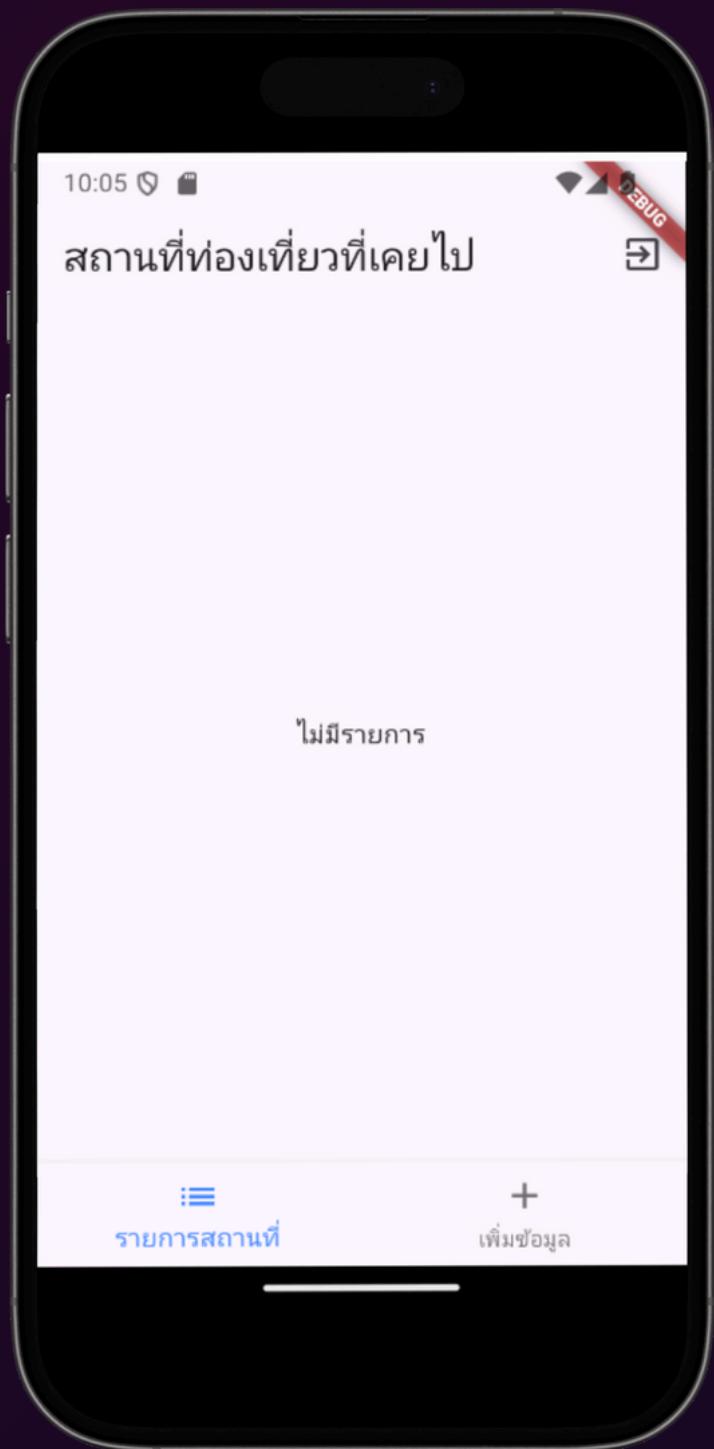
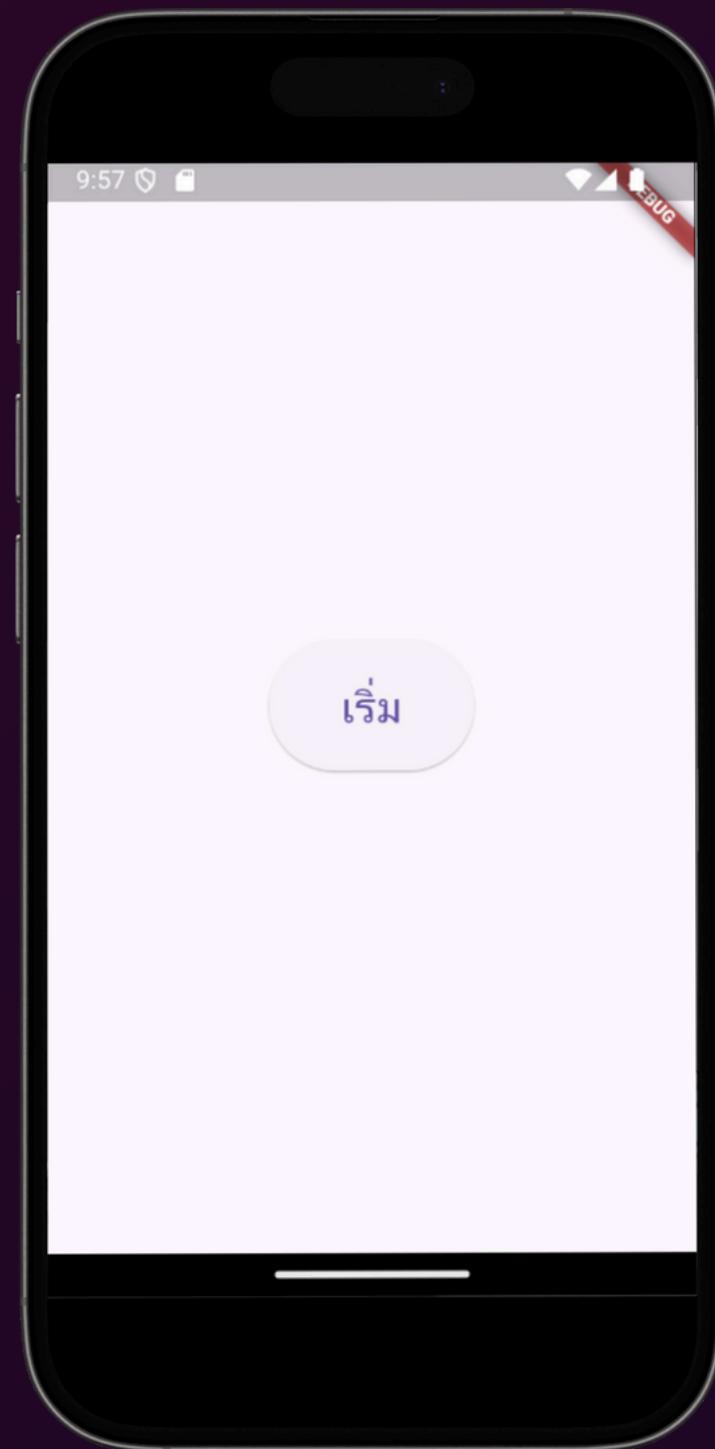
เมื่อคุณเปิดแอปจะขึ้นหน้าจอชี้แจงว่า  
ปุ่มกดตรงกลางเพื่อเริ่มต้นใช้งาน  
และปุ่มไอคอนทางขวาบนเมื่อคุณ  
แล้วละออกจากแอปนี้

03

เมื่อคุณเพิ่มข้อมูลจะมีให้กรอก  
ข้อมูลการบันทึกสถานที่ท่องเที่ยว  
กีฬาใช้งานไป

04

เมื่อคุณบันทึกแล้วแอปจะเด้งมา  
หน้าแรกแต่จะแสดงประวัติกีฬาใช้  
งานกรอกไว้และสามารถกดเข้าไป  
เพื่อดูหรือแก้ไขข้อมูลข้างในได้



# \*\*\*ໃນໄຟລ໌ databases\*\*\*

## Code ທີ່ແກ້ໄຂທີ່ຈະນັດ



```
lib
  └── databases
      ├── attraction_db.dart //ເພີ່ມເຂົ້າມາ
      └── transaction_db.dart //ໄມ່ໃຊ້
  └── models
      └── touristAttraction //ປັບປຸງແປ່ງຂໍ້ມູນບາງສ່ວນ
  └── provider
      ├── attraction_provider.dart //ເພີ່ມເຂົ້າມາ
      └── transaction_provider.dart //ໄມ່ໃຊ້
  └── screens
      ├── detail_screen.dart //ເພີ່ມເຂົ້າມາ
      ├── edit_screen.dart //ປັບປຸງແປ່ງຂໍ້ມູນບາງສ່ວນ
      ├── form_screen.dart //ປັບປຸງແປ່ງຂໍ້ມູນບາງສ່ວນ
      ├── home_screen.dart //ປັບປຸງແປ່ງຂໍ້ມູນບາງສ່ວນ
      └── welcome_screen.dart //ເພີ່ມເຂົ້າມາ
  main.dart //ປັບປຸງແລ້ງຂໍ້ມູນບາງສ່ວນ
```

```
import 'dart:io';
import 'package:path/path.dart';
import 'package:path_provider/path_provider.dart';
import 'package:sembast/sembast.dart';
import 'package:sembast/sembast_io.dart';
import '../models/touristAttraction.dart';

class AttractionDB {
    String dbName;

    AttractionDB({required this.dbName});

    Future<Database> openDatabase() async {
        Directory appDirectory = await getApplicationDocumentsDirectory();
        String dbLocation = join(appDirectory.path, dbName);

        DatabaseFactory dbFactory = databaseFactoryIo;
        Database db = await dbFactory.openDatabase(dbLocation);
        return db;
    }

    Future<int> insertAttraction(TouristAttraction attraction) async {
        var db = await openDatabase();
        var store = intMapStoreFactory.store('attractions');

        var keyID = await store.add(db, {
            "name": attraction.name,
            "province": attraction.province,
            "date": attraction.date.toIso8601String(),
            "highlight": attraction.highlight,
            "feeling": attraction.feeling,
            "imagePath": attraction.imagePath,
        });
        await db.close();
        return keyID;
    }
}
```

```
Future<List<TouristAttraction>> loadAllAttractions() async {
    var db = await openDatabase();
    var store = intMapStoreFactory.store('attractions');
    var snapshot = await store.find(db, finder: Finder(sortOrders: [SortOrder(Field.key, false)]));
    List<TouristAttraction> attractions = [];
    for (var record in snapshot) {
        attractions.add(TouristAttraction(
            keyID: record.key,
            name: record['name'].toString(),
            province: record['province'].toString(),
            date: DateTime.parse(record['date'].toString()),
            highlight: record['highlight'].toString(),
            feeling: record['feeling'].toString(),
            imagePath: record['imagePath'].toString(),
        ));
    }
    await db.close();
    return attractions;
}

Future<void> deleteAttraction(int? keyID) async {
    var db = await openDatabase();
    var store = intMapStoreFactory.store('attractions');
    await store.delete(db, finder: Finder(filter: Filter.equals(Field.key, keyID)));
    await db.close();
}

Future<void> updateAttraction(TouristAttraction attraction) async {
    var db = await openDatabase();
    var store = intMapStoreFactory.store('attractions');
    await store.record(attraction.keyID!).update(db, {
        "name": attraction.name,
        "province": attraction.province,
        "date": attraction.date.toIso8601String(),
        "highlight": attraction.highlight,
        "feeling": attraction.feeling,
        "imagePath": attraction.imagePath,
    });
    await db.close();
}
```

## \*\*ໄຟລ໌ attraction\_db.dart\*\*

ການປັບປຸງແປ່ງ : ສ້າງໄຟລ໌ໃໝ່: ໄຟລ໌ attraction\_db.dart ເປັນໄຟລ໌ຮູານຂໍ້ມູນສໍາຮັບລັດການຂໍ້ມູນ ສານທີ່ກ່ອງເຖິງວ່າ ຊື່ໄໝ່ເຕີຍມີມາກ່ອນໃນໂປຣເຈກຕົ່ມ

- ສ້າງຄລາສ AttractionDB ສໍາຮັບລັດການການເຊື່ອມຕ່ອງແລະການກຳຈານກັບຮູານຂໍ້ມູນ sembast ທີ່ເກີ່ວຂ້ອງກັບສານທີ່ກ່ອງເຖິງວ່າ
- ມີເມນອດຕ່າງ ຈະ ເຊັ່ນ openDatabase(), insertAttraction(), loadAllAttractions(), deleteAttraction(), updateAttraction()
- ໃຊ້ intMapStoreFactory.store('attractions') ເພື່ອເກີບຂໍ້ມູນສານທີ່ກ່ອງເຖິງວ່າໃນຮູານຂໍ້ມູນ

## \*\*\*ในไฟล์ models\*\*

```
class TouristAttraction {  
    final int? keyID;  
    final String name;  
    final String province;  
    final DateTime date;  
    final String highlight;  
    final String feeling;  
    final String imagePath;  
  
    TouristAttraction({  
        this.keyID,  
        required this.name,  
        required this.province,  
        required this.date,  
        required this.highlight,  
        required this.feeling,  
        required this.imagePath,  
    });  
}
```

### \*\*\*ไฟล์ attraction\_provider.dart\*\*\*

การเปลี่ยนแปลง : สร้างไฟล์ใหม่: ไฟล์ attraction\_provider.dart เป็น Provider สำหรับจัดการสถานะของข้อมูลสถานที่ท่องเที่ยวในแอป

- สร้างคลาส AttractionProvider ที่ extends ChangeNotifier
- มีรายการ attractions สำหรับเก็บข้อมูลสถานที่ท่องเที่ยว
- メソドต่าง ๆ:
  - initData(): โหลดข้อมูลจากฐานข้อมูล
  - addAttraction(): เพิ่มสถานที่ท่องเที่ยวใหม่
  - deleteAttraction(): ลบสถานที่ท่องเที่ยว
  - updateAttraction(): แก้ไขข้อมูลสถานที่ท่องเที่ยว

### \*\*\*ไฟล์ touristAttraction.dart\*\*\*

การเปลี่ยนแปลง : สร้างไฟล์ใหม่: ไฟล์ touristAttraction เป็นโมเดลข้อมูลสำหรับสถานที่ท่องเที่ยว

-สร้างคลาส touristAttraction ที่มีคุณสมบัติ:

- keyID: หมายเลขบุเดพะในฐานข้อมูล
- name: ชื่อสถานที่ท่องเที่ยว
- province: จังหวัดที่ตั้ง
- date: วันที่ไปเที่ยว
- highlight: จุดเด่นของสถานที่
- feeling: ความรู้สึกที่มีต่อสถานที่
- imagePath: ไฟล์ภาพของสถานที่

## \*\*\*ในไฟล์ models\*\*\*

```
import 'package:flutter/foundation.dart';  
import '../databases/attraction_db.dart';  
import '../models/touristAttraction.dart';  
  
class AttractionProvider with ChangeNotifier {  
    List<TouristAttraction> attractions = [];  
  
    List<TouristAttraction> getAttractions() {  
        return attractions;  
    }  
  
    void initData() async {  
        var db = AttractionDB(dbName: 'attractions.db');  
        attractions = await db.loadAllAttractions();  
        notifyListeners();  
    }  
  
    void addAttraction(TouristAttraction attraction) async {  
        var db = AttractionDB(dbName: 'attractions.db');  
        await db.insertAttraction(attraction);  
        attractions = await db.loadAllAttractions();  
        notifyListeners();  
    }  
  
    void deleteAttraction(int? keyID) async {  
        var db = AttractionDB(dbName: 'attractions.db');  
        await db.deleteAttraction(keyID);  
        attractions = await db.loadAllAttractions();  
        notifyListeners();  
    }  
  
    void updateAttraction(TouristAttraction attraction) async {  
        var db = AttractionDB(dbName: 'attractions.db');  
        await db.updateAttraction(attraction);  
        attractions = await db.loadAllAttractions();  
        notifyListeners();  
    }  
}
```

# \*\*\*ในไฟล์ screens\*\*\*

```
●●●  
import 'dart:io';  
import 'package:flutter/material.dart';  
import 'package:intl/intl.dart';  
import '../models/touristAttraction.dart';  
import 'edit_screen.dart';  
  
class DetailScreen extends StatelessWidget {  
    final TouristAttraction attraction;  
  
    DetailScreen({required this.attraction});  
  
    @override  
    Widget build(BuildContext context) {  
        return Scaffold(  
            appBar: AppBar(  
                title: Text(attraction.name),  
                actions: [  
                    IconButton(  
                        icon: Icon(Icons.edit),  
                        onPressed: () {  
                            Navigator.push(context, MaterialPageRoute(  
                                builder: (context) => EditScreen(attraction: attraction),  
                            ));  
                        },  
                    ),  
                ],  
            ),  
            body: ListView(  
                children: [  
                    Image.file(  
                        File(attraction.imagePath),  
                        width: double.infinity,  
                        height: 250,  
                        fit: BoxFit.cover,  
                    ),  
                    ListTile(  
                        title: Text('จังหวัด'),  
                        subtitle: Text(attraction.province),  
                    ),  
                    ListTile(  
                        title: Text('วันที่'),  
                        subtitle: Text(DateTimeFormat('dd MMM  
yyyy').format(attraction.date)),  
                    ),  
                    ListTile(  
                        title: Text('จุดเด่น'),  
                        subtitle: Text(attraction.highlight),  
                    ),  
                    ListTile(  
                        title: Text('ความรู้สึก'),  
                        subtitle: Text(attraction.feeling),  
                    ),  
                ],  
            );  
    }  
}
```

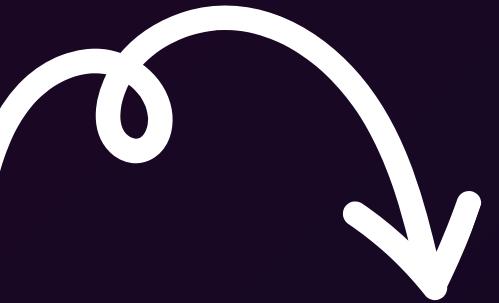
## \*\*ไฟล์ detail\_screen.dart\*\*

การเปลี่ยนแปลง : สร้างไฟล์ใหม่: ไฟล์ detail\_screen.dart เป็นหน้าจอสำหรับแสดงรายละเอียดของสถานที่ท่องเที่ยว

- แสดงข้อมูลรายละเอียดของสถานที่ท่องเที่ยวที่ผู้ใช้เลือก
- มีการแสดงผลภาพ ชื่อ จังหวัด วันที่ จุดเด่น และความรู้สึก
- มีปุ่มแก้ไข (IconButton ที่มีไอคอน Icons.edit) ที่เมื่อกดแล้วจะนำไปยังหน้าแก้ไข (edit\_screen.dart)



```
●●●  
  
// lib/screens/welcome_screen.dart  
import 'package:flutter/material.dart';  
import '../main.dart'; // นำเข้า MyHomePage  
  
class WelcomeScreen extends StatelessWidget {  
    @override  
    Widget build(BuildContext context) {  
        return Scaffold(  
            // ใช้ Stack เพื่อวางภาพพื้นหลังและปุ่มนบานภาพ  
            body: Stack(  
                children: [  
                    // ภาพพื้นหลัง  
                    Container(  
                        decoration: BoxDecoration(  
                            image: DecorationImage(  
                                image: AssetImage(  
                                    'assets/images/background.jpg'), // ใส่ภาพพื้นหลัง (ถ้ามี)  
                                fit: BoxFit.cover,  
                            ),  
                        ),  
                    ),  
                    // ปุ่ม "เริ่ม"  
                    Center(  
                        child: ElevatedButton(  
                            onPressed: () {  
                                Navigator.pushReplacement(  
                                    context,  
                                    MaterialPageRoute(builder: (context) => MyHomePage()),  
                                );  
                            },  
                            child: Text('เริ่ม', style: TextStyle(fontSize: 24)),  
                            style: ElevatedButton.styleFrom(  
                                padding: EdgeInsets.symmetric(horizontal: 40, vertical: 20),  
                            ),  
                        ),  
                    ),  
                ],  
            );  
    }  
}
```



## \*\*ไฟล์ welcome\_screen.dart\*\*

การเปลี่ยนแปลง : สร้างไฟล์ใหม่: ไฟล์ welcome\_screen.dart เป็นหน้าจอต้อนรับที่มีปุ่ม "เริ่ม" เมื่อกดแล้วจะนำไปยังหน้าแรกของแอป

- ออกแบบหน้าจอต้อนรับที่มีภาพพื้นหลังและปุ่ม "เริ่ม"
- ปรับแต่งลักษณะของปุ่มและพื้นหลังเพื่อให้ดูสวยงาม
- เมื่อกดปุ่ม "เริ่ม" จะใช้ Navigator.pushReplacement เพื่อนำทางไปยัง MyHomePage

# \*\*\*ໃນໄຟລ໌ screens\*\*\*

```
import 'dart:io';
import 'package:flutter/material.dart';
import 'package:image_picker/image_picker.dart';
import 'package:provider/provider.dart';
import '../models/touristAttraction.dart';
import '../provider/attraction_provider.dart';

class EditScreen extends StatefulWidget {
    final TouristAttraction attraction;

    EditScreen({required this.attraction});

    @override
    _EditScreenState createState() => _EditScreenState();
}

class _EditScreenState extends State<EditScreen> {
    final GlobalKey<FormState> formKey = GlobalKey<FormState>();
    late TextEditingController nameController;
    late TextEditingController provinceController;
    late TextEditingController dateController;
    late TextEditingController highlightController;
    late TextEditingController feelingController;
    File? _image;
    final picker = ImagePicker();

    late DateTime selectedDate;

    @override
    void initState() {
        super.initState();
        nameController = TextEditingController(text: widget.attraction.name);
        provinceController = TextEditingController(text: widget.attraction.province);
        dateController = TextEditingController(text:
            widget.attraction.date.toLocal().toString().split(' ')[0]);
        highlightController = TextEditingController(text: widget.attraction.highlight);
        feelingController = TextEditingController(text: widget.attraction.feeling);
        _image = File(widget.attraction.imagePath);
        selectedDate = widget.attraction.date;
    }

    Future<void> _pickImage() async {
        final pickedFile = await picker.pickImage(source: ImageSource.gallery);
        setState(() {
            _image = pickedFile != null ? File(pickedFile.path) : _image;
        });
    }

    Future<void> _selectDate(BuildContext context) async {
        final DateTime? picked = await showDatePicker(
            context: context,
            initialDate: selectedDate,
            firstDate: DateTime(2000),
            lastDate: DateTime.now(),
        );
        if (picked != null)
            setState(() {
                selectedDate = picked;
                dateController.text = picked.toLocal().toString().split(' ')[0];
            });
    }
}
```

```
@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: const Text('ແກ້ໄຂສານທີ່ທອງເຫັນ'),
        ),
        body: Form(
            key: formKey,
            child: ListView(
                padding: EdgeInsets.all(16.0),
                children: [
                    // ສຳຄັນທີ່ທອງເຫັນ
                    TextFormField(
                        decoration: const InputDecoration(labelText: 'ສຳຄັນທີ່ທອງເຫັນ'),
                        controller: nameController,
                        validator: (value) {
                            if (value == null || value.isEmpty) {
                                return 'ກະຕຸາກຮອກສຳຄັນທີ່ທອງເຫັນ';
                            }
                            return null;
                        },
                    ),
                    // ຈົງວັດ
                    TextFormField(
                        decoration: const InputDecoration(labelText: 'ຈົງວັດ'),
                        controller: provinceController,
                        validator: (value) {
                            if (value == null || value.isEmpty) {
                                return 'ກະຕຸາກຮອກຈົງວັດ';
                            }
                            return null;
                        },
                    ),
                    // ວັນທີ
                    TextFormField(
                        decoration: const InputDecoration(labelText: 'ວັນທີ'),
                        controller: dateController,
                        readOnly: true,
                        onTap: () => _selectDate(context),
                        validator: (value) {
                            if (value == null || value.isEmpty) {
                                return 'ກະຕຸາເລືອກວັນທີ';
                            }
                            return null;
                        },
                    ),
                    // ຈຸດເຕັນ
                    TextFormField(
                        decoration: const InputDecoration(labelText: 'ຈຸດເຕັນ'),
                        controller: highlightController,
                        validator: (value) {
                            if (value == null || value.isEmpty) {
                                return 'ກະຕຸາກຮອກຈຸດເຕັນ';
                            }
                            return null;
                        },
                    ),
                ],
            ),
        ),
    );
}
```

```
TextFormField(
    decoration: const InputDecoration(labelText: 'ຄວາມຮູ້ສຶກ'),
    controller: feelingController,
    validator: (value) {
        if (value == null || value.isEmpty) {
            return 'ກະຕຸາກຮອກຄວາມຮູ້ສຶກ';
        }
        return null;
    },
),
// ຖັນການ
SizedBox(height: 10),
_image == null
    ? Text('ບັນໄນໄດ້ເລືອກການ')
    : Image.file(_image!),
ElevatedButton(
    onPressed: _pickImage,
    child: Text('ເປົ້າຍັນການ'),
),
SizedBox(height: 20),
// ເປົ້າຍັນທິກ
ElevatedButton(
    child: const Text('ໜັນທິກການແກ້ໄຂ'),
    onPressed: () {
        if (formKey.currentState!.validate()) {
            var updatedAttraction = TouristAttraction(
                keyID: widget.attraction.keyID,
                name: nameController.text,
                province: provinceController.text,
                date: selectedDate,
                highlight: highlightController.text,
                feeling: feelingController.text,
                imagePath: _image!.path,
            );
            var provider = Provider.of<AttractionProvider>(context, listen: false);
            provider.updateAttraction(updatedAttraction);
            Navigator.pop(context);
        }
    },
),
],
),
),
);
}
}
```

## \*\*ໄຟລ໌ edit\_screen.dart\*\*

ກາຣເປົ້າຍັນແປລງ : ສ້າງໄຟລ໌ໃໝ່: ໄຟລ໌ edit\_screen.dart ເປັນໜ້າລວມສໍາຮັບແກ້ໄຂຂໍ້ມູນລສຖານທີ່ກ່ອງເກີຍວ  
-ຝອຣັມທີ່ມີຟິລ໌ຕ່າງ ຫ້າງ ເຊັ່ນ ຊື່ວ່າ ລົງວັດ ວັນທີ ອຸດເດັ່ນ ຄວາມຮູ້ສຶກ ແລະກາຣເລືອກການ  
-ໂນລດຂໍ້ມູນລສຖານທີ່ກ່ອງເກີຍວທີ່ຕ້ອງກາຣແກ້ໄຂມາແສດງໃນຝອຣັມ  
-ເມື່ອຜູ້ໃຊ້ແກ້ໄຂຂໍ້ມູນລແລະກາບັນທຶກ ຂໍ້ມູນລຈະຖຸກອັບເດີໃນສານຂໍ້ມູນຜ່ານ AttractionProvider

# \*\*\*ໃນໄຟລ໌ screens \*

```
import 'dart:io';
import 'package:flutter/material.dart';
import 'package:image_picker/image_picker.dart';
import 'package:provider/provider.dart';
import '../models/touristAttraction.dart';
import '../provider/attraction_provider.dart';
import '../main.dart';

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

  @override
  _FormScreenState createState() => _FormScreenState();
}

class _FormScreenState extends State<FormScreen> {
  final GlobalKey<FormState> formKey = GlobalKey<FormState>();
  final TextEditingController nameController = TextEditingController();
  final TextEditingController provinceController = TextEditingController();
  final TextEditingController dateController = TextEditingController();
  final TextEditingController highlightController = TextEditingController();
  final TextEditingController feelingController = TextEditingController();
  File? _image;
  final picker = ImagePicker();

  DateTime selectedDate = DateTime.now();

  Future<void> _pickImage() async {
    final pickedFile = await picker.pickImage(source: ImageSource.gallery);
    setState(() {
      _image = pickedFile != null ? File(pickedFile.path) : null;
    });
  }

  Future<void> _selectDate(BuildContext context) async {
    final DateTime? picked = await showDatePicker(
      context: context,
      initialDate: selectedDate,
      firstDate: DateTime(2000),
      lastDate: DateTime.now(),
    );
    if (picked != null)
      setState(() {
        selectedDate = picked;
        dateController.text = picked.toLocal().toString().split(' ')[@];
      });
  }
}
```

```
@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: const Text('เพิ่มสถานที่ท่องเที่ยว'),
        ),
        body: Form(
            key: formKey,
            child: ListView(
                padding: EdgeInsets.all(16.0),
                children: [
                    // ชื่อสถานที่ท่องเที่ยว
                    TextFormField(
                        decoration:
                            const InputDecoration(labelText: 'ชื่อสถานที่ท่องเที่ยว'),
                        controller: nameController,
                        validator: (value) {
                            if (value == null || value.isEmpty) {
                                return 'กรุณากรอกชื่อสถานที่ท่องเที่ยว';
                            }
                            return null;
                        },
                    ),
                    // จังหวัด
                    TextFormField(
                        decoration: const InputDecoration(labelText: 'จังหวัด'),
                        controller: provinceController,
                        validator: (value) {
                            if (value == null || value.isEmpty) {
                                return 'กรุณากรอกจังหวัด';
                            }
                            return null;
                        },
                    ),
                    // วันที่
                    TextFormField(
                        decoration: const InputDecoration(labelText: 'วันที่'),
                        controller: dateController,
                        readOnly: true,
                        onTap: () => _selectDate(context),
                        validator: (value) {
                            if (value == null || value.isEmpty) {
                                return 'กรุณาเลือกวันที่';
                            }
                            return null;
                        },
                    ),
                    // จุดเด่น
                    TextFormField(
                        decoration: const InputDecoration(labelText: 'จุดเด่น'),
                        controller: highlightController,
                        validator: (value) {
                            if (value == null || value.isEmpty) {
                                return 'กรุณากรอกจุดเด่น';
                            }
                            return null;
                        },
                    ),
                ],
            ),
        ),
    );
}
```

```
TextFormField(
    decoration: const InputDecoration(labelText: 'គាម្ព់សីក'),
    controller: feelingController,
    validator: (value) {
        if (value == null || value.isEmpty) {
            return 'ក្នុងករណកម្ព់សីក';
        }
        return null;
    },
),
// រូបភាព
SizedBox(height: 10),
_image == null ? Text('មិនໄនៀតិដើរកម្ពោជាបារ') : Image.file(_image!),
ElevatedButton(
    onPressed: _pickImage,
    child: Text('ផែនភាព'),
),
SizedBox(height: 20),
// ឈុំប៉ុនទិក
// ឈុំប៉ុនទិក
ElevatedButton(
    child: const Text('បញ្ជីក'),
    onPressed: () {
        if (formKey.currentState!.validate()) {
            if (_image == null) {
                ScaffoldMessenger.of(context).showSnackBar(
                    SnackBar(content: Text('ក្នុងករណកម្ពោជាបារ')),
                );
                return;
            }
            var attraction = TouristAttraction(
                keyID: null,
                name: nameController.text,
                province: provinceController.text,
                date: selectedDate,
                highlight: highlightController.text,
                feeling: feelingController.text,
                imagePath: _image!.path,
            );
            var provider =
                Provider.of<AttractionProvider>(context, listen: false);
            provider.addAttraction(attraction);

            // ចាប់ពីថ្ងៃនេះនូវការបង្ហាញខាងក្រោមនេះ
            Navigator.pushAndRemoveUntil(
                context,
                MaterialPageRoute(builder: (context) => MyHomePage()),
                (Route<dynamic> route) => false,
            );
        }
    },
),
],
),
),
),
);
```

## \*\*ไฟล์ form\_screen.dart

- ปรับเปลี่ยนฟิล์ตในฟอร์ม จาก title และ amount เป็น name, province, date, highlight, feeling, และการเลือกภาพ (image)
  - เพิ่มฟังก์ชันการเลือกภาพโดยใช้ image\_picker
  - ปรับปรุงการตรวจสอบความถูกต้อง (validator) ของแต่ละฟิล์ต
  - เมื่อกดปุ่มบันทึก จะสร้างวัตถุ TouristAttraction และส่งไปยัง AttractionProvider เพื่อบันทึกลงฐานข้อมูล
  - หลังจากบันทึกแล้ว จะนำทางกลับไปยังหน้าแรกของแอป

# \*\*\*ในไฟล์ screens\*\*

● ● ●

```
import 'dart:io';
import 'package:flutter/material.dart';
import 'package:intl/intl.dart';
import 'package:provider/provider.dart';
import '../provider/attraction_provider.dart';
import '../models/touristAttraction.dart';
import 'detail_screen.dart';
import 'package:flutter/services.dart';
import 'dart:io';

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

  @override
  _HomeScreenState createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  @override
  void initState() {
    super.initState();
    WidgetsBinding.instance.addPostFrameCallback((_) {
      Provider.of<AttractionProvider>(context, listen: false).initData();
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('สถานที่ท่องเที่ยวที่เคยไป'),
        actions: [
          IconButton(
            icon: const Icon(Icons.exit_to_app),
            onPressed: () {
              // ออกจากแอปเป็นอุดมคุณ
              SystemNavigator.pop(); // หรือใช้ exit(0); หากต้องการ
            },
          ),
        ],
    ),
```

```
body: Consumer<AttractionProvider>(
  builder: (context, provider, child) {
    if (provider.attractions.isEmpty) {
      return const Center(child: Text('ไม่มีรายการ'));
    } else {
      return ListView.builder(
        itemCount: provider.attractions.length,
        itemBuilder: (context, index) {
          TouristAttraction attraction = provider.attractions[index];
          return Card(
            elevation: 5,
            margin: const EdgeInsets.symmetric(vertical: 5, horizontal: 8),
            child: ListTile(
              leading: Image.file(
                File(attraction.imagePath),
                width: 80,
                fit: BoxFit.cover,
              ),
              title: Text(attraction.name),
              subtitle: Text('${attraction.province}\n${DateFormat('dd MMM yyyy').format(attraction.date)}'),
              isThreeLine: true,
              onTap: () {
                Navigator.push(context, MaterialPageRoute(
                  builder: (context) => DetailScreen(attraction: attraction),
                ));
              },
              trailing: IconButton(
                icon: const Icon(Icons.delete),
                onPressed: () {
                  provider.deleteAttraction(attraction.keyID);
                },
              ),
            ),
          );
        },
      );
    }
  },
)
```

## \*\*ไฟล์ home\_screen.dart\*\*

การเปลี่ยนแปลง : เดิมแสดงรายการธุรกิจรวม ตอนนี้ถูกปรับปรุงให้แสดงรายการสถานที่ท่องเที่ยว

-เปลี่ยนการแสดงผลรายการ:

- จากรายการธุรกิจเป็นรายการสถานที่ท่องเที่ยว
  - แสดงภาพของสถานที่ ชื่อ วันที่ และมีปุ่มลบ
- เมื่อผู้ใช้กดที่รายการ จะนำทางไปยังหน้ารายละเอียด (detail\_screen.dart)
- เพิ่มปุ่มออกจากแอปใน AppBar โดยใช้ **SystemNavigator.pop()**

## \*\*ไฟล์ main.dart\*\*

```
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'provider/attraction_provider.dart';
import 'screens/home_screen.dart';
import 'screens/form_screen.dart';
import 'screens/welcome_screen.dart';

void main() {
  WidgetsFlutterBinding.ensureInitialized();
  FlutterError.onError = (FlutterErrorDetails details) {
    FlutterError.dumpErrorToConsole(details);
  };
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // Root of your application
  @override
  Widget build(BuildContext context) {
    return ChangeNotifierProvider(
      create: (context) => AttractionProvider(),
      child: MaterialApp(
        title: 'แอปบันทึกสถานที่ท่องเที่ยว',
        theme: ThemeData(
          primarySwatch: Colors.blue,
        ),
        home: WelcomeScreen(),
      ),
    );
  }
}
```

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

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  int _selectedIndex = 0;

  static List<Widget> _widgetOptions = <Widget>[
    HomeScreen(), // แทน const ด้วย
    FormScreen(),
  ];

  void _onItemTapped(int index) {
    setState(() {
      _selectedIndex = index;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: _widgetOptions.elementAt(_selectedIndex),
      bottomNavigationBar: BottomNavigationBar(
        items: const <BottomNavigationBarItem>[
          BottomNavigationBarItem(
            icon: Icon(Icons.list),
            label: 'รายการสถานที่',
          ),
          BottomNavigationBarItem(
            icon: Icon(Icons.add),
            label: 'เพิ่มข้อมูล',
          ),
        ],
        currentIndex: _selectedIndex,
        selectedItemColor: Colors.blueAccent,
        onTap: _onItemTapped,
      ),
    );
  }
}
```

### การเปลี่ยนแปลง :

-เพิ่มการนำเข้าไฟล์ใหม่

- welcome\_screen.dart
- home\_screen.dart
- form\_screen.dart

-เปลี่ยนค่า home ใน MaterialApp จาก MyHomePage() เป็น

WelcomeScreen()

-ปรับปรุงการจัดการการนำทางและการใช้ Provider

-ใน MyHomePage ยังคงมีการจัดการเก็บเพื่อสั่งระหว่าง HomeScreen และ FormScreen

# สรุป

## ผังก์ชันหลักที่เพิ่มเข้ามา

01

เพิ่มไฟล์ใหม่: attraction\_db.dart , touristAttraction.dart ,  
attraction\_provider.dart , detail\_screen.dart ,  
edit\_screen.dart , welcome\_screen.dart

02

ปรับปรุงไฟล์เดิม: form\_screen.dart, home\_screen.dart, main.dart

03

ไฟล์ที่ไม่ได้ใช้: transaction\_db.dart, transactions.dart,  
transaction\_provider.dart

01

การบันทึกข้อมูลสถานที่ท่องเที่ยวพร้อมรูปภาพ

02

การแสดงรายการสถานที่ท่องเที่ยวที่บันทึกไว้

03

หน้าจอต้อนรับที่มีปุ่ม "เริ่ม"

# Thank You