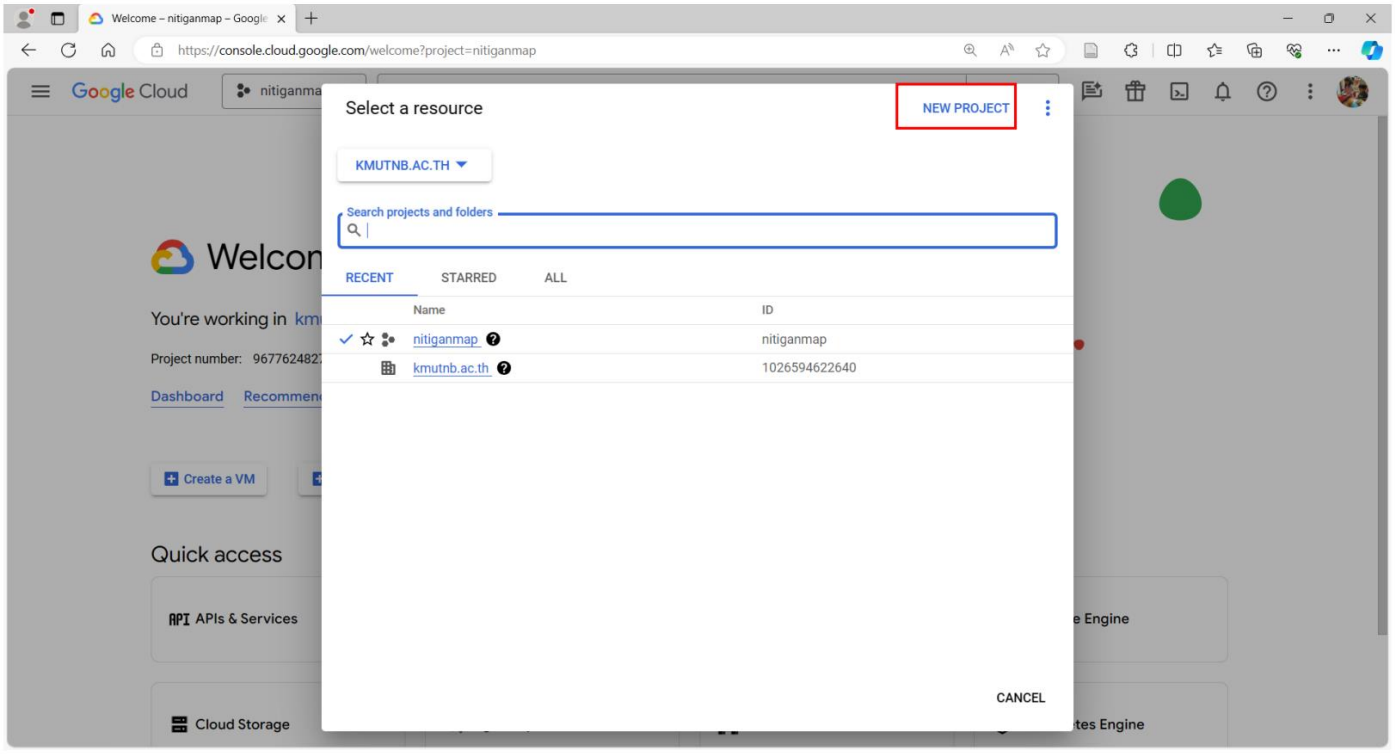
Flutter Basic Google Maps 2025

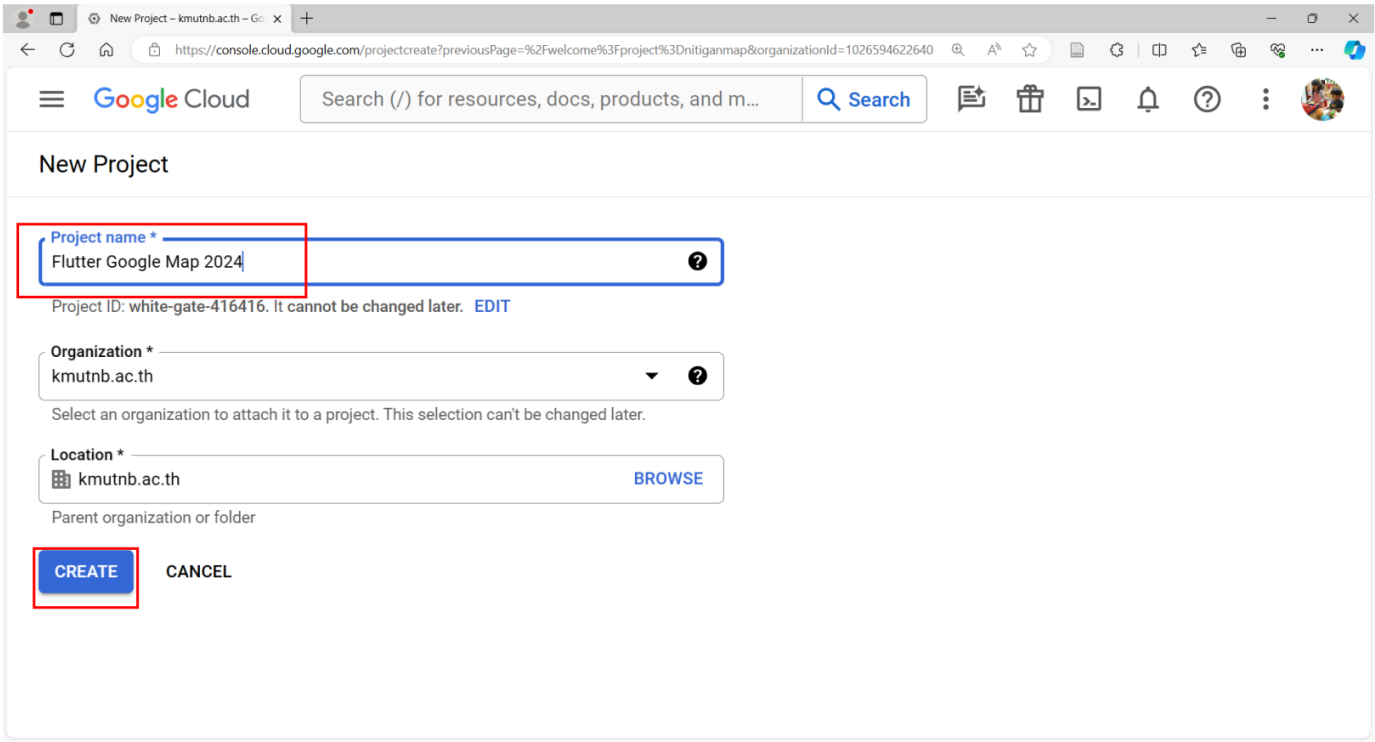
การเขียนแอปพลิเคชันเพื่อแสดงแผนที่ด้วย Google Map สามารถกำหนดการแสดงผลแผนที่ไปยังตำแหน่งของสถานที่ต่าง ๆ ได้โดยการเรียกใช้ package ไลบารี google_maps_flutter ที่ช่วยให้สามารถเรียกใช้งาน Google Map ที่สามารถใช้ในการค้นหาสถานที่ด้วยการกำหนดพิกัดและแผนที่ได้อีกด้วย ซึ่งข้อสำคัญในการใช้งานแผนที่นี้มีข้อสำคัญเกี่ยวกับการเข้าถึงบริการข้อมูลและสิทธิ์ในการใช้ข้อมูลพิกัดจากอุปกรณ์เคลื่อนที่ ดังนั้นแอปพลิเคชันจะต้องถูกตรวจสอบและควบคุม

ดังนั้นการใช้งานจะต้องลงทะเบียนเพื่อสร้าง API KEY สำหรับการใช้งานของผู้พัฒนาและแอปพลิเคชันนั้น ๆ

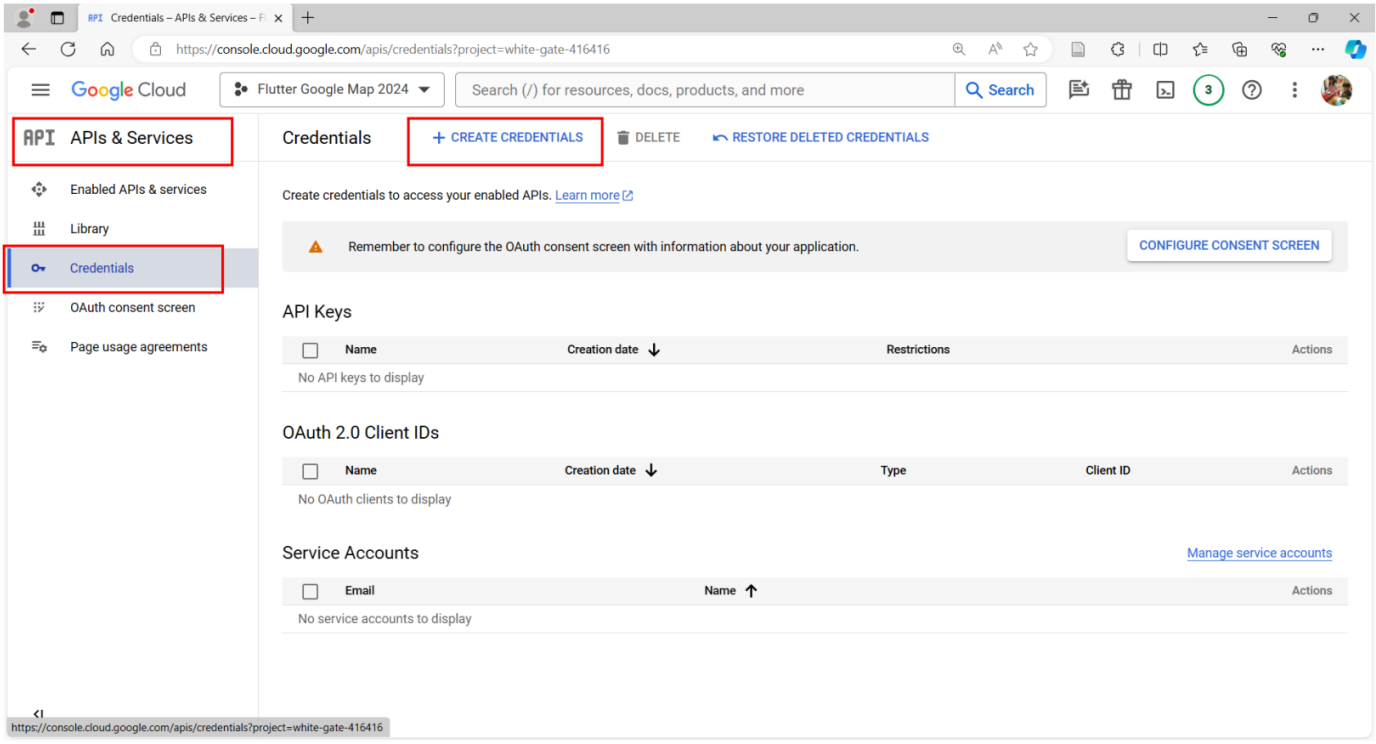
ขั้นตอนการสร้าง API KEY ไปที่ https://console.cloud.google.com/ click ที่ NEW PROJECT

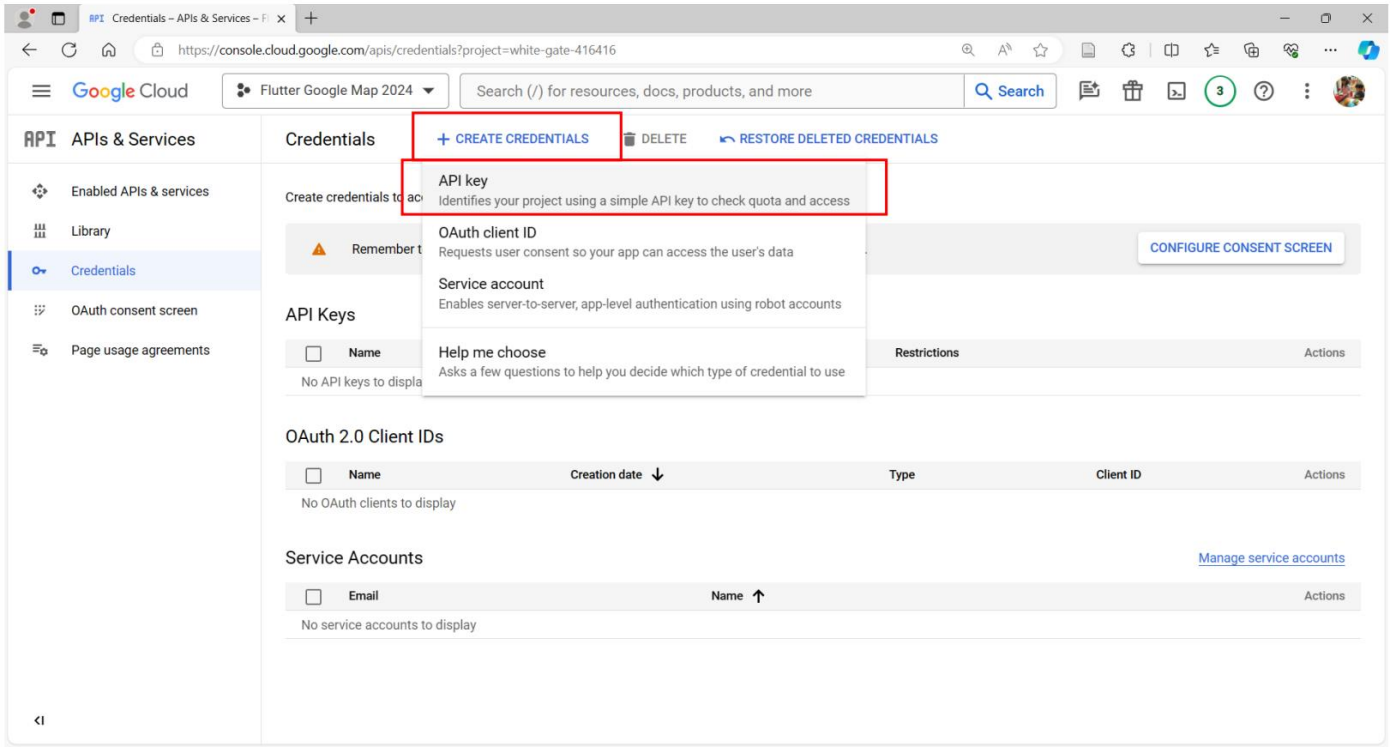


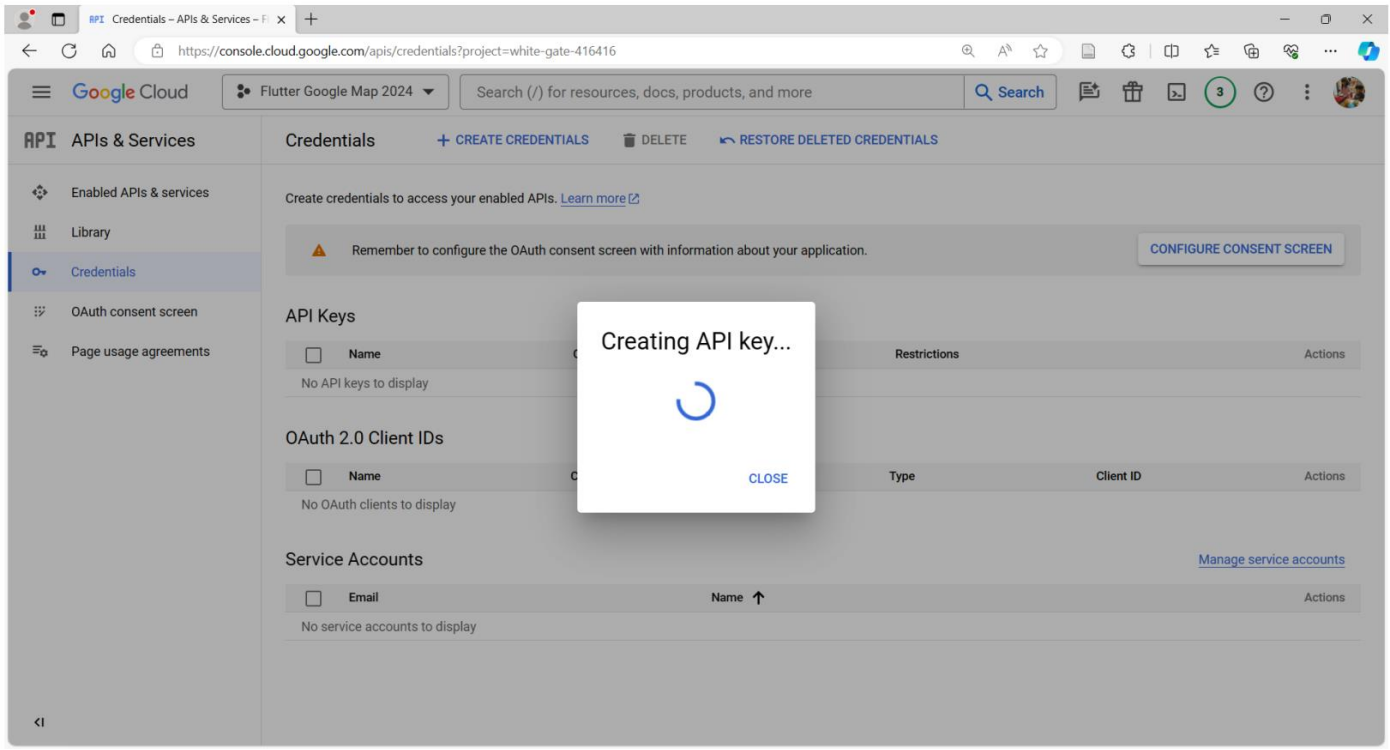
ตั้งชื่อ project (** มีการจำกัด จำนวนของการสร้าง project)

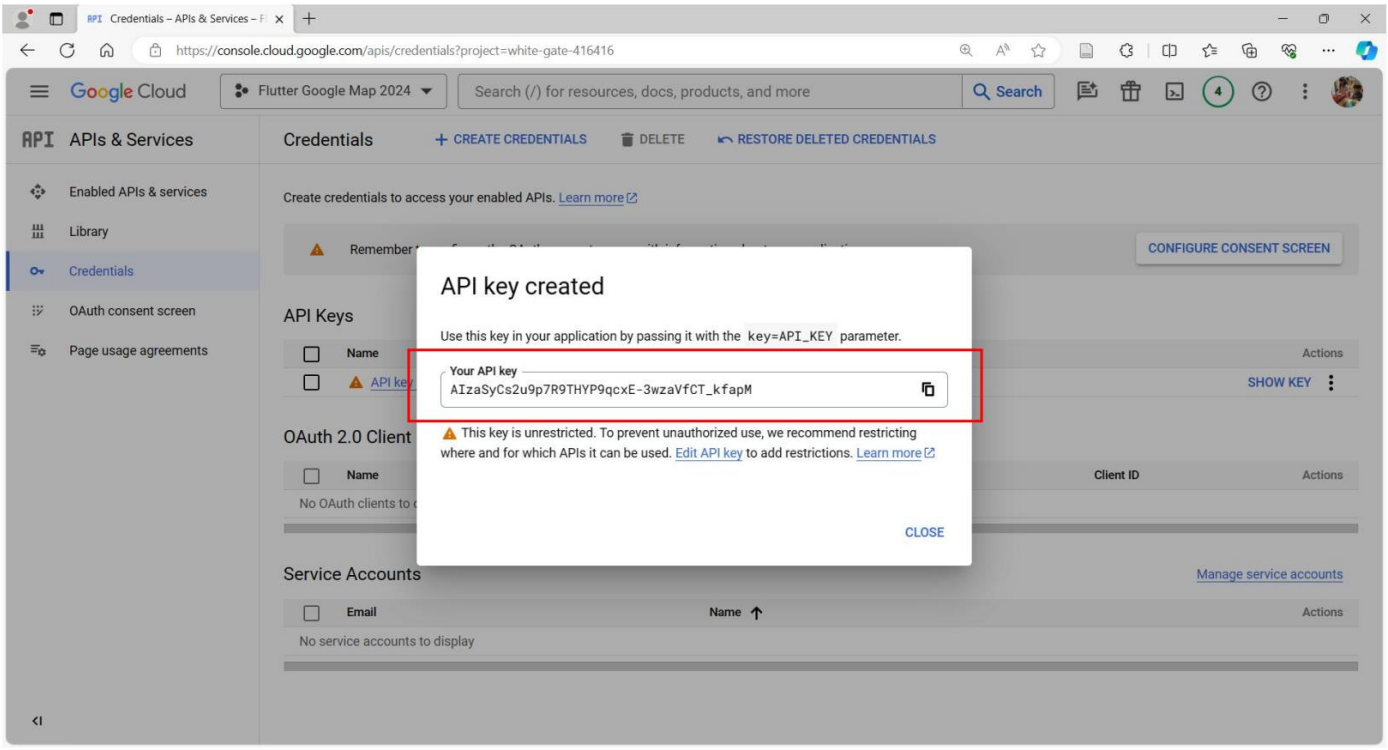


สร้าง api key โดย click ที่ Credentials และ เลือกสร้างที่ CREATE CREDENTIALS

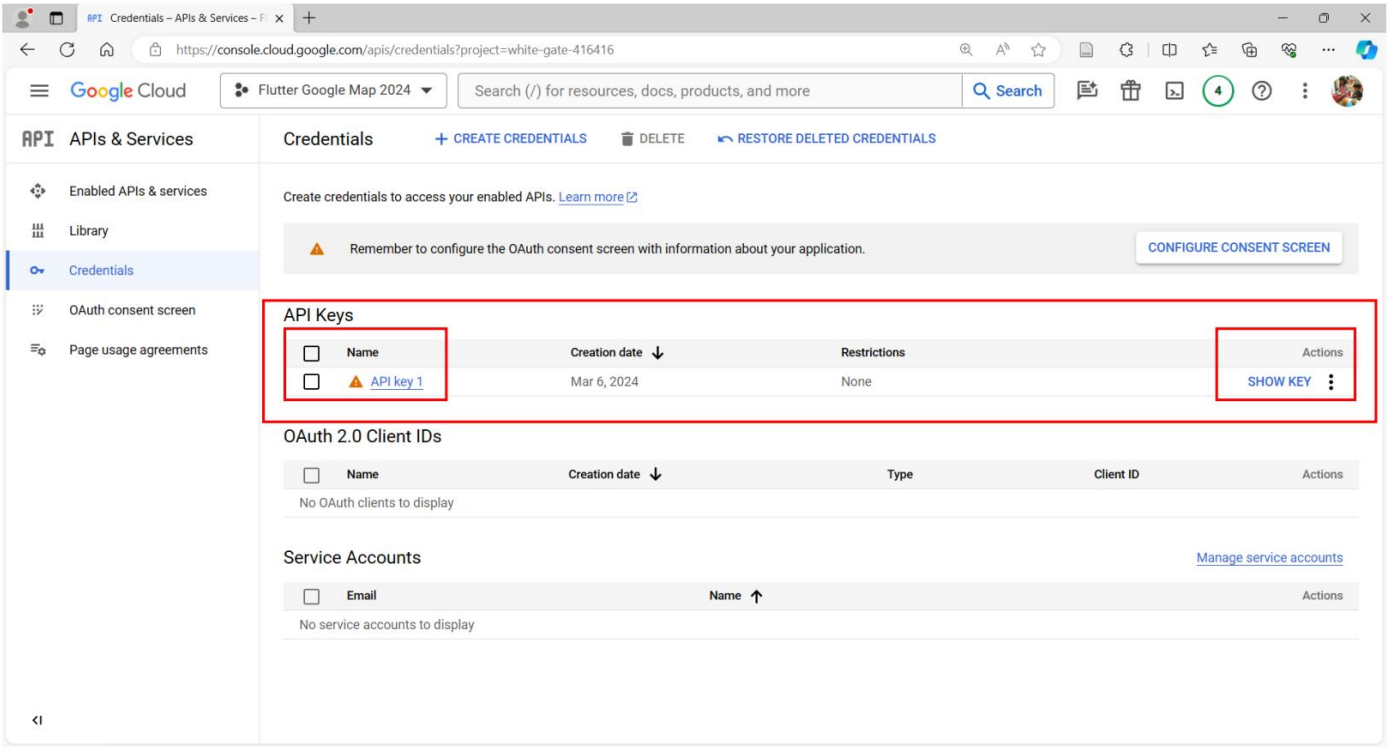



Credentials – APIs & Services – F
https://console.cloud.google.com/apis/credentials?project=white-gate-416416
Google Cloud
Flutter Google Map 2024
Search (/) for resources, docs, products, and more
Search
APIs & Services
Credentials
+ CREATE CREDENTIALS
DELETE
RESTORE DELETED CREDENTIALS
Enabled APIs & services
Library
Credentials
OAuth consent screen
Page usage agreements
API key
Identifies your project using a simple API key to check quota and access
OAuth client ID
Requests user consent so your app can access the user's data
Service account
Enables server-to-server, app-level authentication using robot accounts
Help me choose
Asks a few questions to help you decide which type of credential to use
CONFIGURE CONSENT SCREEN
API Keys
Name
Restrictions
Actions
No API keys to displa
OAuth 2.0 Client IDs
Name
Creation date
Type
Client ID
Actions
No OAuth clients to display
Service Accounts
Manage service accounts
Email
Name
Actions
No service accounts to display


Credentials – APIs & Services – F
https://console.cloud.google.com/apis/credentials?project=white-gate-416416
Google Cloud
Flutter Google Map 2024
Search (/) for resources, docs, products, and more
Search
APIs & Services
Credentials
+ CREATE CREDENTIALS
DELETE
RESTORE DELETED CREDENTIALS
Enabled APIs & services
Library
Credentials
OAuth consent screen
Page usage agreements
Create credentials to access your enabled APIs. Learn more
Remember to configure the OAuth consent screen with information about your application.
CONFIGURE CONSENT SCREEN
API Keys
Name
Restrictions
Actions
No API keys to display
OAuth 2.0 Client IDs
Name
Type
Client ID
Actions
No OAuth clients to display
Service Accounts
Manage service accounts
Email
Name
Actions
No service accounts to display
Creating API key...
CLOSE

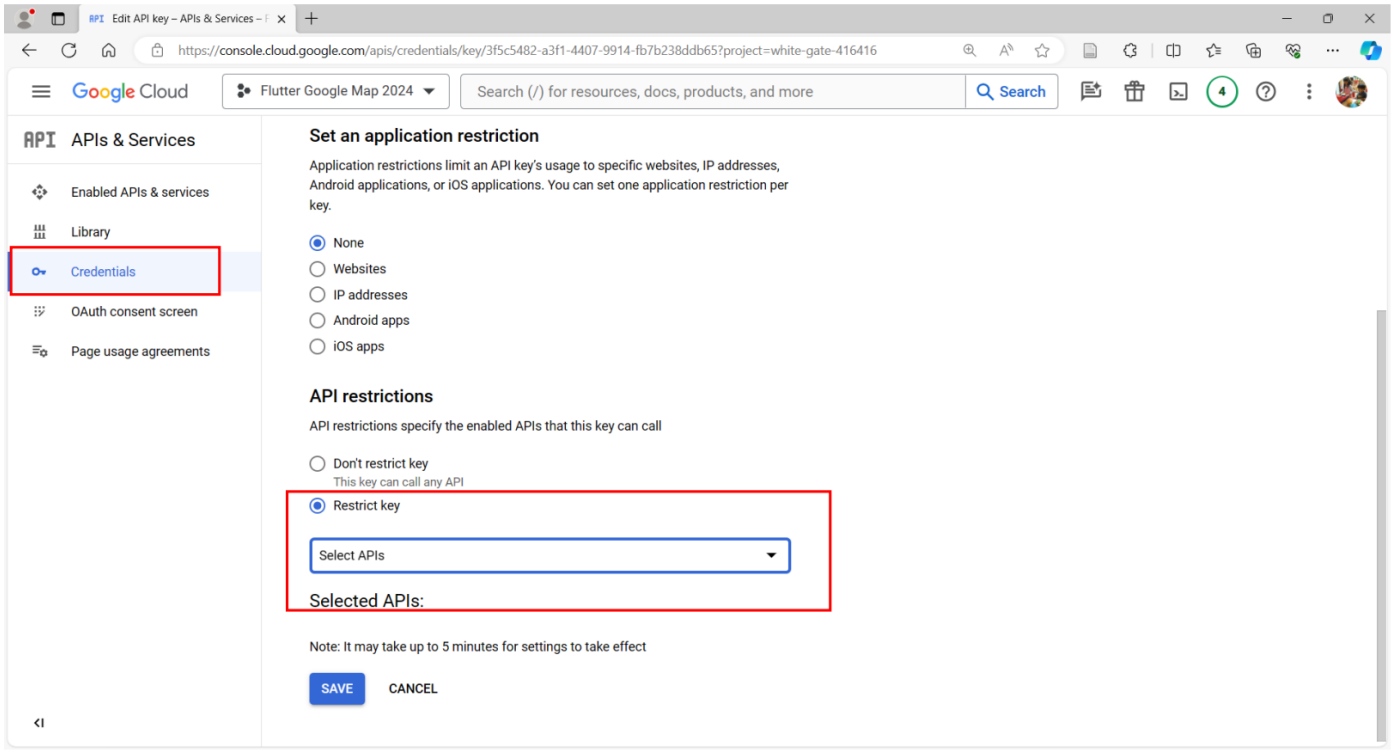


ให้ Copy Key เก็บไว้สำหรับใช้ใน Code ของ Flutter

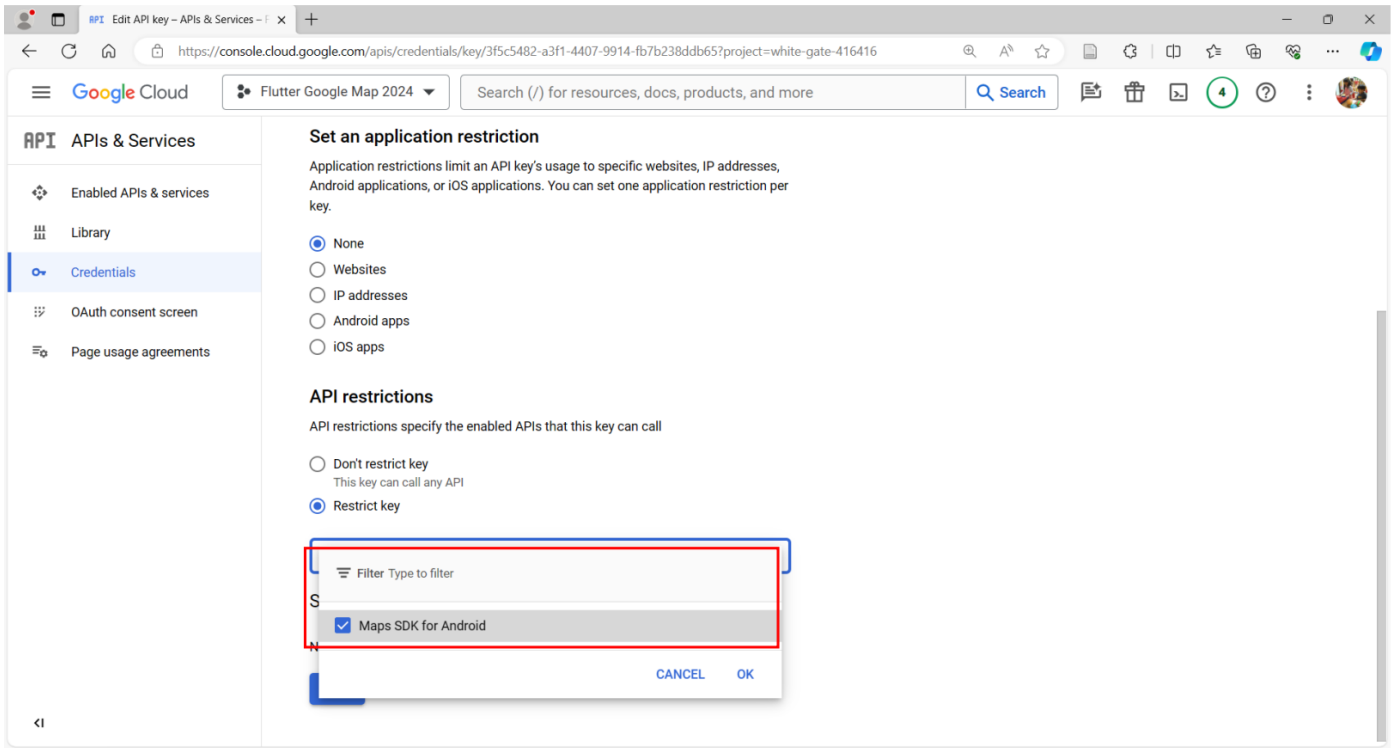


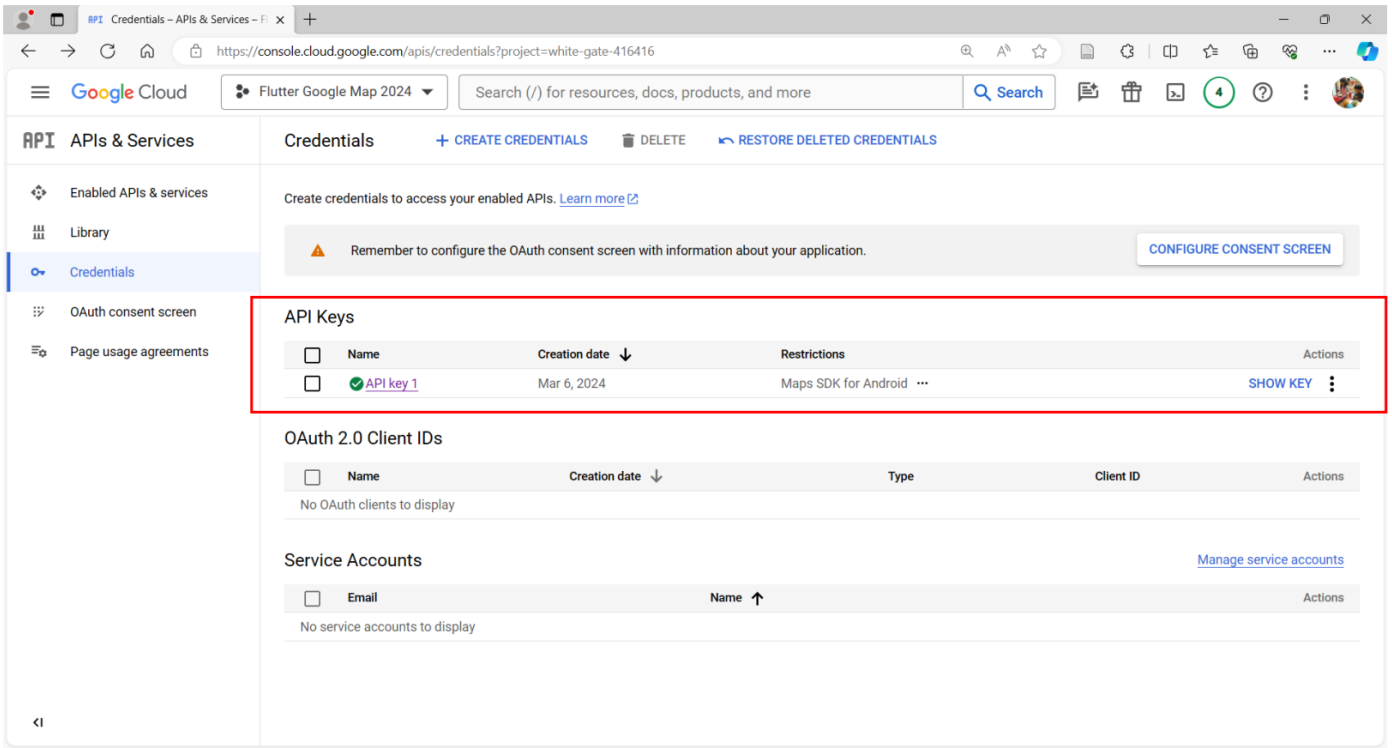
จากนั้น Click ที่ API Key เพื่อกำหนดค่าให้กับ API

กำหนด API restrictions



Check Maps SDK Android และ click SAVE




Credentials – APIs & Services – F
https://console.cloud.google.com/apis/credentials?project=white-gate-416416
Google Cloud
Flutter Google Map 2024
Search (/) for resources, docs, products, and more
Search
APIs & Services
Enabled APIs & services
Library
Credentials
OAuth consent screen
Page usage agreements
Credentials
+ CREATE CREDENTIALS
DELETE
RESTORE DELETED CREDENTIALS
Create credentials to access your enabled APIs. Learn more
Remember to configure the OAuth consent screen with information about your application.
CONFIGURE CONSENT SCREEN
API Keys
Name
Creation date
Restrictions
Actions
API key 1
Mar 6, 2024
Maps SDK for Android
SHOW KEY
OAuth 2.0 Client IDs
Name
Creation date
Type
Client ID
Actions
No OAuth clients to display
Service Accounts
Manage service accounts
Email
Name
Actions
No service accounts to display

การเขียนแอปพลิเคชัน สำหรับ Android ให้เข้าไปเพิ่ม API Key ที่ไฟล์ AndroidManifest.xml

\android\app\src\main\AndroidManifest.xml

```xml
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
    <application
        android:label="google_maps_in_flutter"
        android:name="${applicationName}"
        android:icon="@mipmap/ic_launcher">

        <!-- TODO: Add your Google Maps API key here -->
        <meta-data android:name="com.google.android.geo.API_KEY"
               android:value="AIzaSyCs2u9p7R9THYP9qcxE-3wzaVfCT_kfapM"/>

        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:launchMode="singleTop"
            android:taskAffinity=""
            android:theme="@style/LaunchTheme"
            android:configChanges="orientation|keyboardHidden|keyboard|screenSize|smallestScreenSize|locale|layoutDirection|
            android:hardwareAccelerated="true"
            android:windowSoftInputMode="adjustResize">
            <!-- Specifies an Android theme to apply to this Activity as soon as
                 the Android process has started. This theme is visible to the user
                 while the Flutter UI initializes. After that, this theme continues
                 to determine the Window background behind the Flutter UI. -->
            <meta-data
              android:name="io.flutter.embedding.android.NormalTheme"
              android:resource="@style/NormalTheme"
              />
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <!-- Don't delete the meta-data below.
             This is used by the Flutter tool to generate GeneratedPluginRegistrant.java -->
        <meta-data
            android:name="flutterEmbedding"
            android:value="2" />
    </application>
    <!-- Required to query activities that can process text, see:
         https://developer.android.com/training/package-visibility and
         https://developer.android.com/reference/android/content/Intent#ACTION_PROCESS_TEXT.

         In particular, this is used by the Flutter engine in io.flutter.plugin.text.ProcessTextPlugin. -->
    <queries>
        <intent>
            <action android:name="android.intent.action.PROCESS_TEXT"/>
            <data android:mimeType="text/plain"/>
        </intent>
    </queries>
</manifest>
```

API KEY

ติดตั้ง google_maps_flutter ที่ pubspec.yaml

```yaml
dependencies:
  flutter:
    sdk: flutter

  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.8
  google_maps_flutter: ^2.10.0
```

กำหนด SDK Version ที่ไฟล์ android/app/build.gradle โดยกำหนด minSdkVersion เป็น 21

```
plugins {
    id("com.android.application")
    id("kotlin-android")
    // The Flutter Gradle Plugin must be applied after the Android and Kotlin Gradle plugins.
    id("dev.flutter.flutter-gradle-plugin")
}

android {
    namespace = "com.example.google_maps_in_flutter"
    compileSdk = flutter.compileSdkVersion
    ndkVersion = flutter.ndkVersion

    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_11
        targetCompatibility = JavaVersion.VERSION_11
    }

    kotlinOptions {
        jvmTarget = JavaVersion.VERSION_11.toString()
    }

    defaultConfig {
        // TODO: Specify your own unique Application ID
        // (https://developer.android.com/studio/build/application-id.html).
        applicationId = "com.example.google_maps_in_flutter"
        // You can update the following values to match your application needs.
        // For more information, see: https://flutter.dev/to/review-gradle-config.
        minSdk = 21
        targetSdk = flutter.targetSdkVersion
        versionCode = flutter.versionCode
        versionName = flutter.versionName
    }

    buildTypes {
        release {
            // TODO: Add your own signing config for the release build.
            // Signing with the debug keys for now, so `flutter run --release` works.
            signingConfig = signingConfigs.getByName("debug")
        }
    }
}

flutter {
    source = "../.."
}

```

เริ่มต้นการสร้าง Application

import Library

```
import 'package:google_maps_flutter/google_maps_flutter.dart';
```

สร้างตัวแปรประเภท GoogleMapController คือ class ที่เอาไว้สร้าง Future อีกที เอาไว้ควบคุมการทำงาน

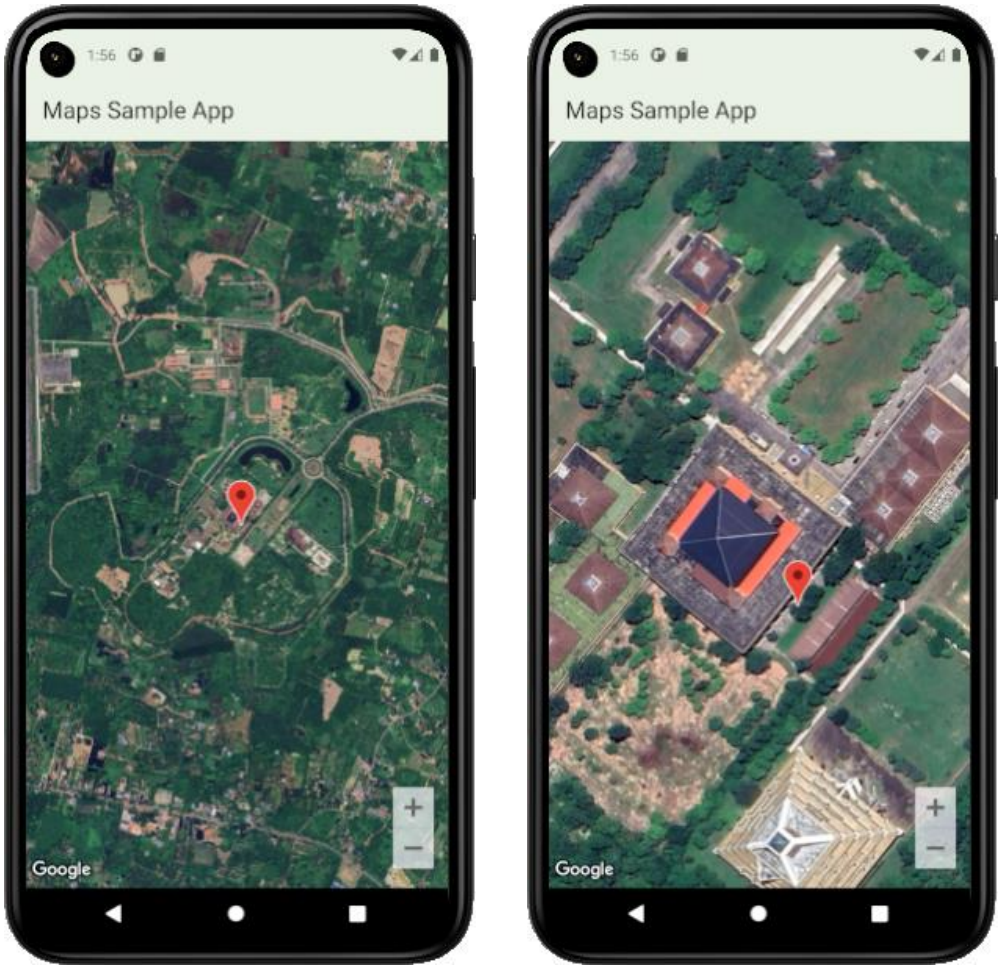
```
class _MyHomePageState extends State<MyHomePage> {
  late GoogleMapController mapController;
```

Library จะมี Widget มาให้ชื่อว่า GoogleMap() เพื่อแสดงแผนที่

เริ่มต้น GoogleMap จะต้อง initialCameraPosition เสมอ โดยกำหนดพิกัดจุดเริ่มต้นสถานที่จะให้ GoogleMap แสดงขึ้นมา

main.dart

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

void main() {
  runApp(const MyApp());
}

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

  @override
  State<MyApp> createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  late GoogleMapController mapController;

  final LatLng _center = const LatLng(14.1585968, 101.3457501);
  final Set<Marker> _markers = Set<Marker>();

  void _onMapCreated(GoogleMapController controller) {
    mapController = controller;
    _addMarker(_center, 'My Maker');
  }

  void _addMarker(LatLng position, String markerId) {
    setState(() {
      _markers.add(
        Marker(
          markerId: MarkerId(markerId),
          position: position,
          infoWindow: InfoWindow(title: markerId),
        ),
      );
    });
  }

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      theme: ThemeData(useMaterial3: true, colorSchemeSeed: Colors.green[700]),
      home: Scaffold(
        appBar: AppBar(title: const Text('Maps Sample App'), elevation: 2),
        body: GoogleMap(
          onMapCreated: _onMapCreated,
          initialCameraPosition: CameraPosition(target: _center, zoom: 16.0),
          markers: _markers,
          mapType: MapType.satellite,
        ),
      ),
    );
  }
}

```



** สามารถกำหนดระยะการซูมแผนที่ ได้ตั้งแต่ 0 – 20

** สามารถรูปแบบการแสดงของแผนที่ จาก mapType ได้

การกำหนด MapType สำหรับรูปแบบการแสดงผลแผนที่

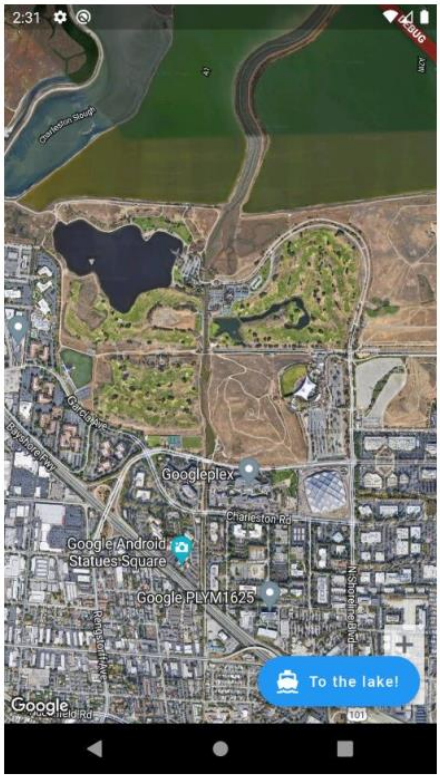
```
45      body: GoogleMap(
46        mapType: MapType.,
47        initialCameraPosi
48        onMapCreated: (Go
49          _controller.com
50        },
51      ), // GoogleMap
52      floatingActionButto
```

hybrid MapType
none
normal
satellite
terrain
values

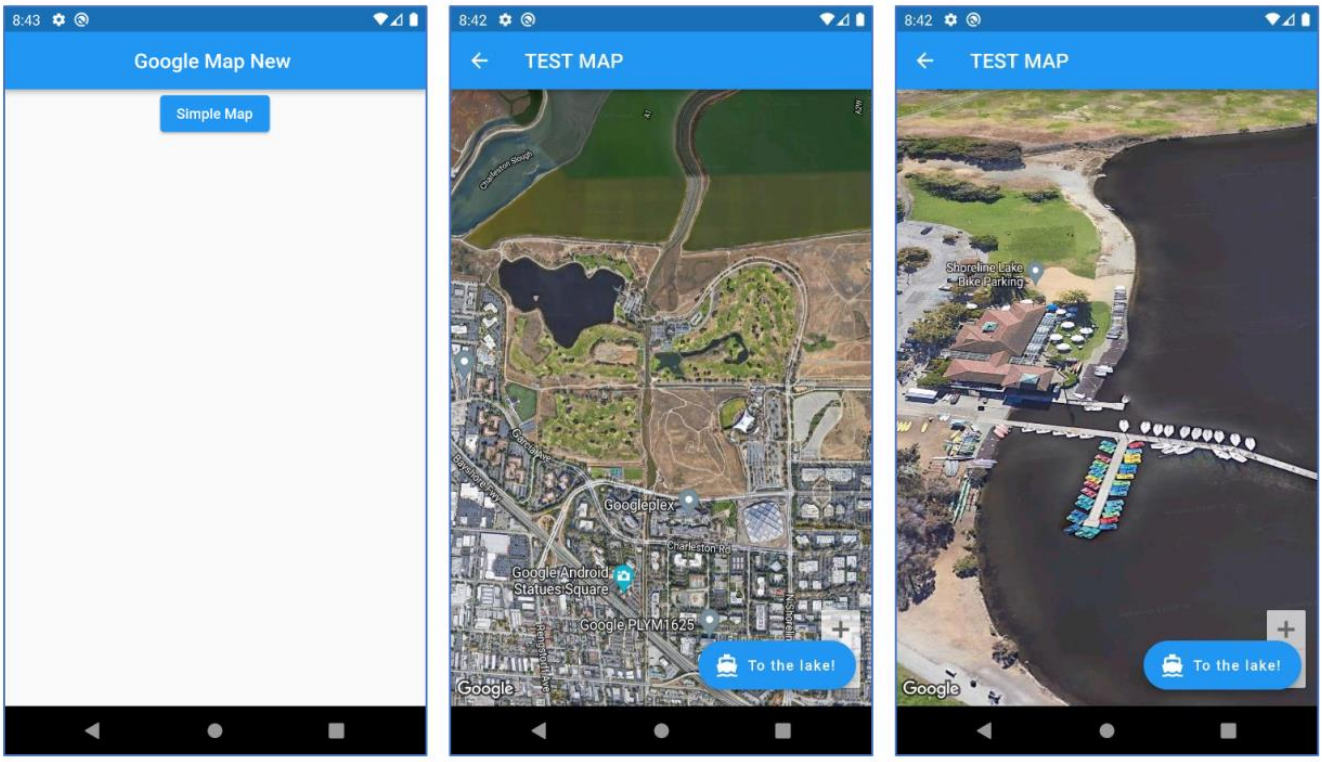
ปรับแก้ main.dart เพิ่มเติม แล้วทดสอบการแสดงผลการทำงานของแอปพลิเคชัน

```dart
import 'dart:async';

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

void main() {
  runApp(MyApp());
}

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

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

class MapSample extends StatefulWidget {
  //const MapSample({super.key});

  @override
  State<MapSample> createState() => _MapSampleState();
}

class _MapSampleState extends State<MapSample> {
  Completer<GoogleMapController> _controller = Completer();

  static final CameraPosition _kGooglePLex = CameraPosition(
    target: LatLng(37.42796133580664, -122.085749655962),
    zoom: 14.0,
  );

  static final CameraPosition _kLake = CameraPosition(
    bearing: 192.8334901395799,
    target: LatLng(37.43296265331129, -122.088832357078792),
    tilt: 59.44,
    zoom: 19.0,
  );

  @override
  Widget build(BuildContext context) {
    return new Scaffold(
      body: GoogleMap(
        mapType: MapType.hybrid,
        initialCameraPosition: _kGooglePLex,
        onMapCreated: (GoogleMapController controller) {
          _controller.complete(controller);
        },
      ),
      floatingActionButton: FloatingActionButton.extended(
        onPressed: _goToTheLake,
        label: Text("Go to the Lake"),
        icon: Icon(Icons.directions_boat),
      ),
    );
  }

  Future<void> _goToTheLake() async {
    final GoogleMapController controller = await _controller.future;
    controller.animateCamera(CameraUpdate.newCameraPosition(_kLake));
  }
}
```

ทดสอบการทำงานของแอปพลิเคชัน



ปรับรูปแบบการแสดงผลของให้สามารถกลับไปที่หน้าแรกของ App ได้



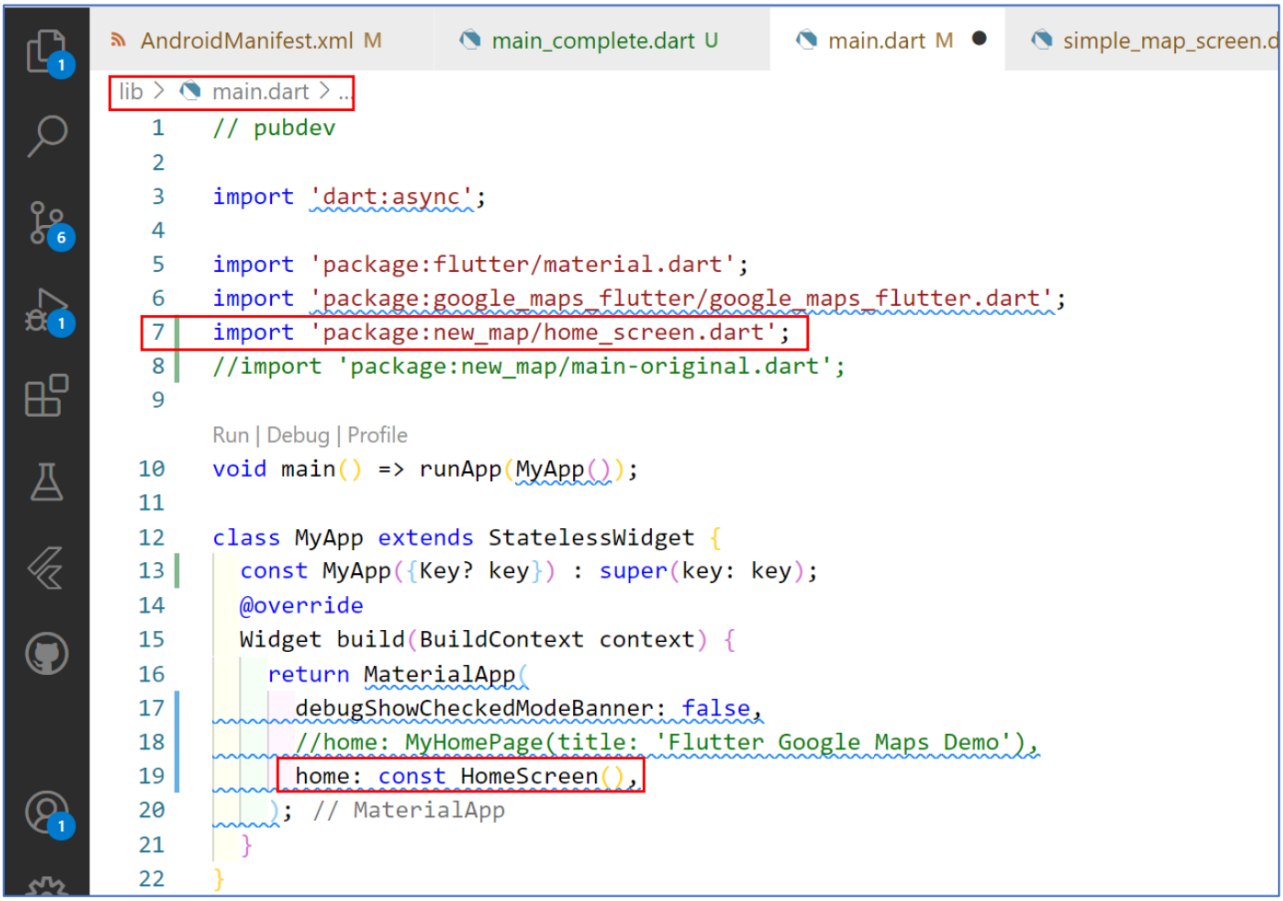
สร้างไฟล์ home_screen.dart ให้อยู่ใน folder lib ที่มี main.dart

lib > home_screen.dart > ..

```dart
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:new_map/screens/simple_map_screen.dart';

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

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text("Google Map New "),
        centerTitle: true,
      ), // AppBar
      body: SizedBox(
        width: MediaQuery.of(context).size.width,
        child: Column(
          children: [
            ElevatedButton(
              onPressed: () {
                Navigator.of(context)
                    .push(MaterialPageRoute(builder: (BuildContext context) {
                  return MapSample();
                })); // MaterialPageRoute
              },
              child: const Text("Simple Map")) // ElevatedButton
          ],
        ), // Column
      ), // SizedBox
    ); // Scaffold
  }
}
```

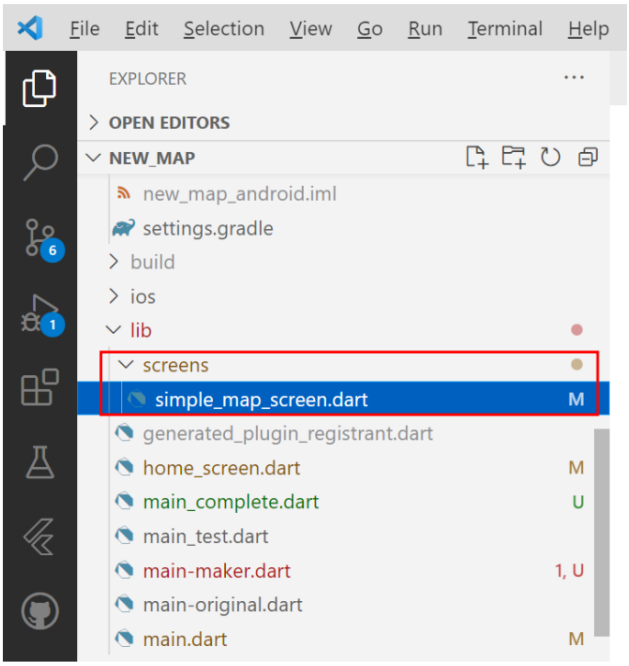
แก้ไขไฟล์ main.dart ให้ไปเรียกใช้ class ของ home_screen.dart

และสร้างการแสดง GUI ของ App ที่ไฟล์ simple_map_screen.dart ที่จะสร้างเพิ่มต่อไป



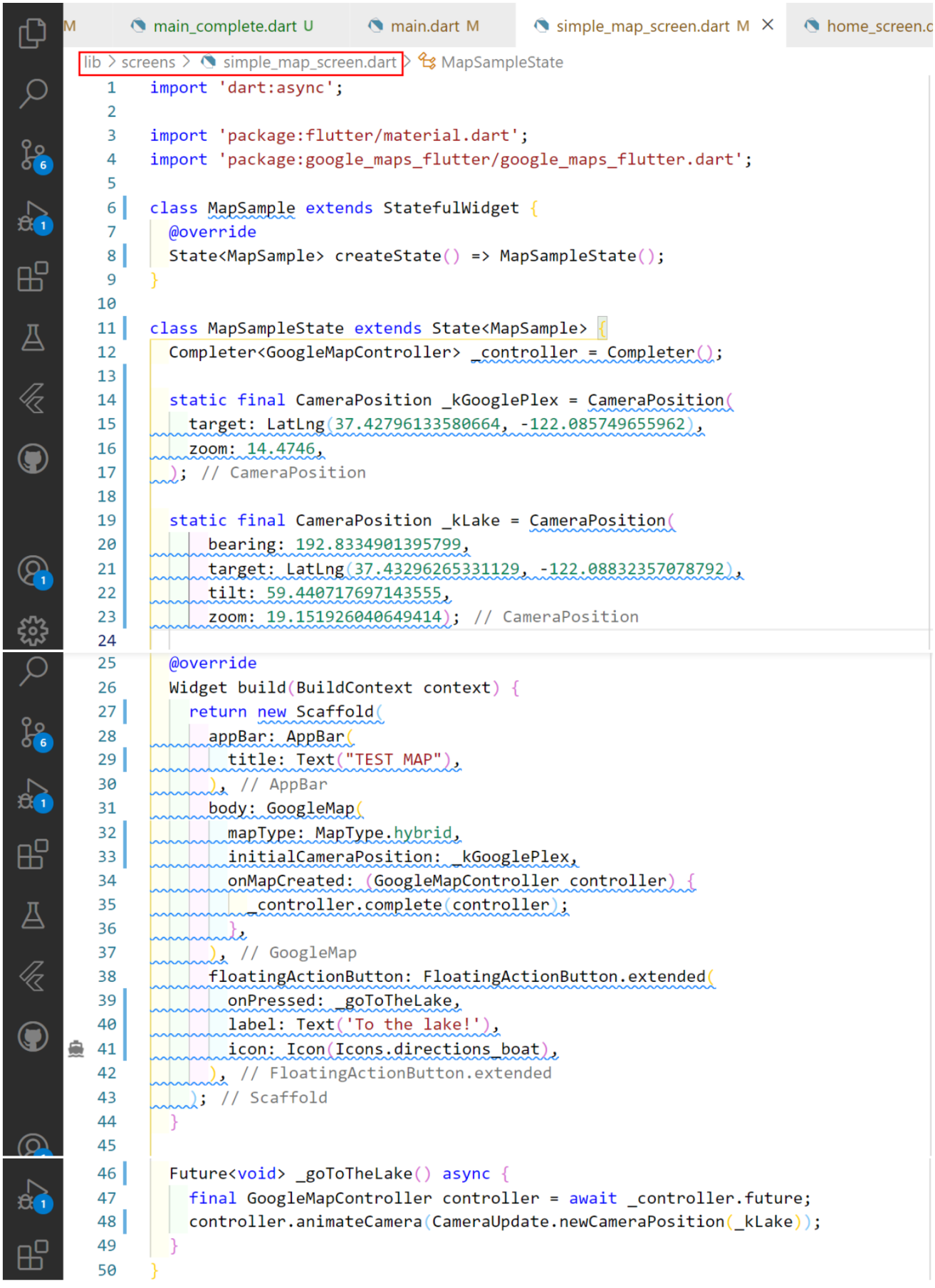
ทำการสร้าง folder screens ภายใต้ folder lib

จากนั้นสร้างไฟล์ simple_map_screen.dart ให้อยู่ใน folder screens



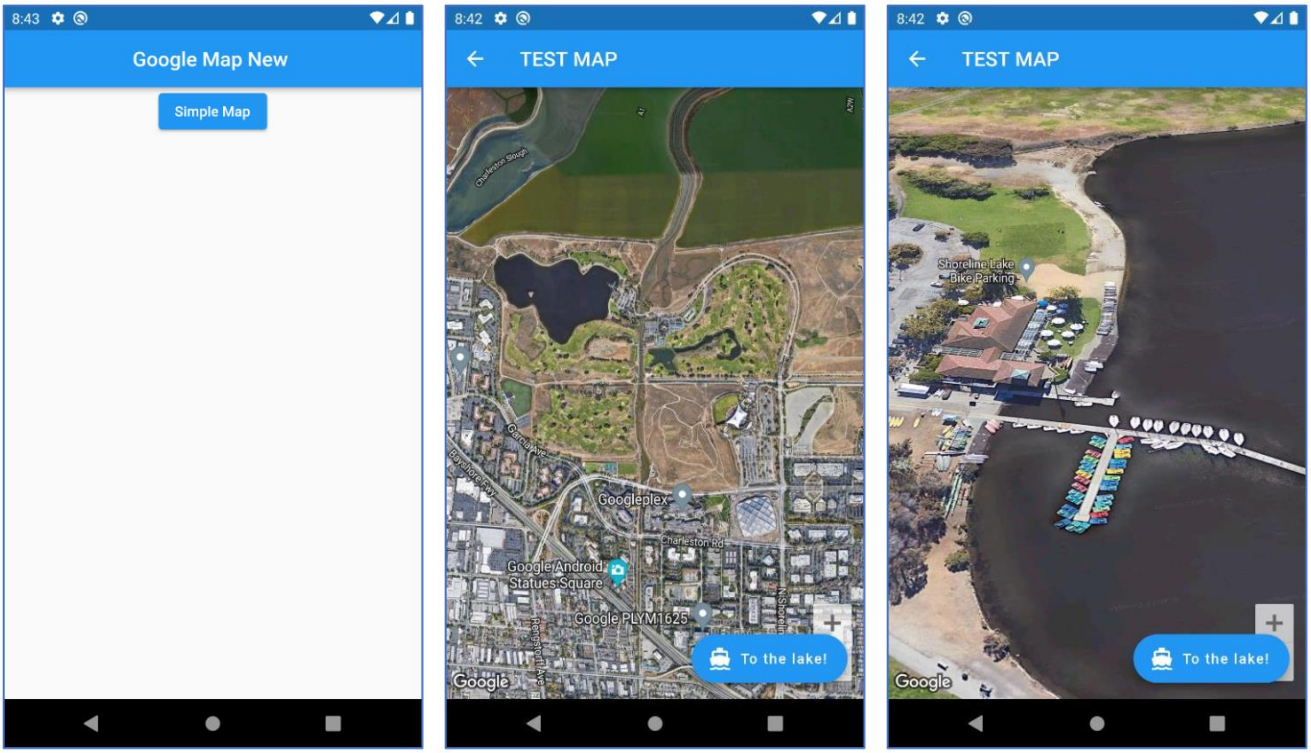
***** หน้า GUI ที่สร้างนี้ จะถูกเรียกใช้งาน เมื่อกดปุ่ม simple map ที่หน้าแรกของ APP**

ไฟล์ simple_map_screen.dart

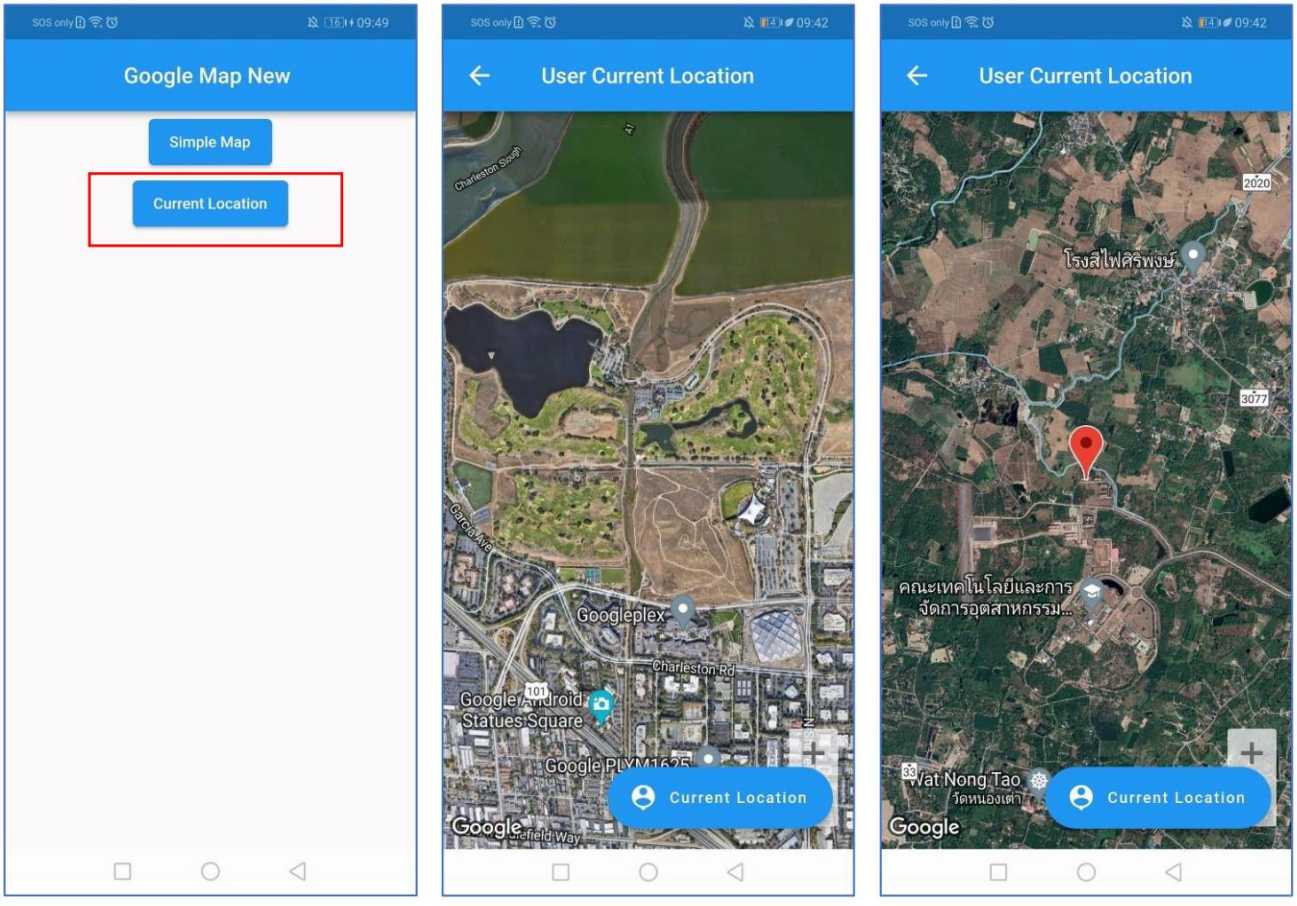



```dart
import 'dart:async';

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

class MapSample extends StatefulWidget {
  @override
  State<MapSample> createState() => MapSampleState();
}

class MapSampleState extends State<MapSample> {
  Completer<GoogleMapController> _controller = Completer();

  static final CameraPosition _kGooglePlex = CameraPosition(
    target: LatLng(37.42796133580664, -122.085749655962),
    zoom: 14.4746,
  ); // CameraPosition

  static final CameraPosition _kLake = CameraPosition(
      bearing: 192.8334901395799,
      target: LatLng(37.43296265331129, -122.08832357078792),
      tilt: 59.440717697143555,
      zoom: 19.151926040649414); // CameraPosition

  @override
  Widget build(BuildContext context) {
    return new Scaffold(
      appBar: AppBar(
        title: Text("TEST MAP"),
      ), // AppBar
      body: GoogleMap(
        mapType: MapType.hybrid,
        initialCameraPosition: _kGooglePlex,
        onMapCreated: (GoogleMapController controller) {
          _controller.complete(controller);
        },
      ), // GoogleMap
      floatingActionButton: FloatingActionButton.extended(
        onPressed: _goToTheLake,
        label: Text('To the lake!'),
        icon: Icon(Icons.directions_boat),
      ), // FloatingActionButton.extended
    ); // Scaffold
  }

  Future<void> _goToTheLake() async {
    final GoogleMapController controller = await _controller.future;
    controller.animateCamera(CameraUpdate.newCameraPosition(_kLake));
  }
}
```

ทดสอบการทำงานของ APP



เพิ่มการทำงานให้แสดง maker ที่พิกัดตำแหน่งปัจจุบันของอุปกรณ์

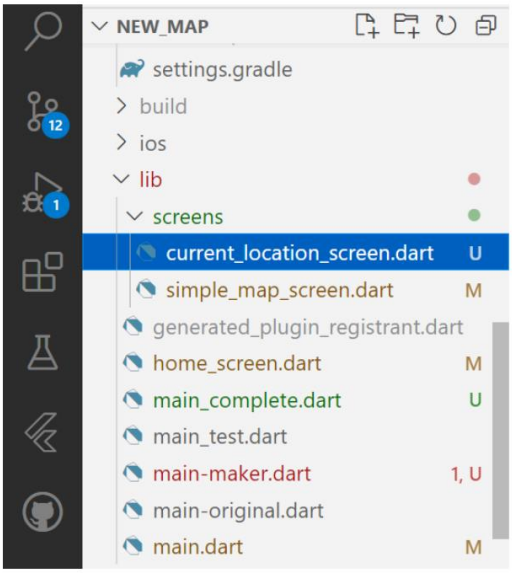


แก้ไข home_screen.dart ให้เพิ่มปุ่มสำหรับเรียกหน้า GUI สำหรับแสดงตำแหน่งปัจจุบัน

lib > home_screen.dart > _HomeScreenState > build

```dart
    appBar: AppBar(
      title: const Text("Google Map New "),
      centerTitle: true,
    ), // AppBar
    body: SizedBox(
      width: MediaQuery.of(context).size.width,
      child: Column(
        children: [
          ElevatedButton(
            onPressed: () {
              Navigator.of(context)
                  .push(MaterialPageRoute(builder: (BuildContext context) {
                return MapSample();
              })); // MaterialPageRoute
            },
            child: const Text("Simple Map")), // ElevatedButton
          ElevatedButton(
            onPressed: () {
              Navigator.of(context)
                  .push(MaterialPageRoute(builder: (BuildContext context) {
                return CurrentLocation();
              })); // MaterialPageRoute
            },
            child: const Text("Current Location")) // ElevatedButton
        ],
      ), // Column
    ), // SizedBox
  ); // Scaffold
  }
}
```

สร้างไฟล์ current_location_screen.dart ใน folder screens



ไฟล์ current_location_screen.dart

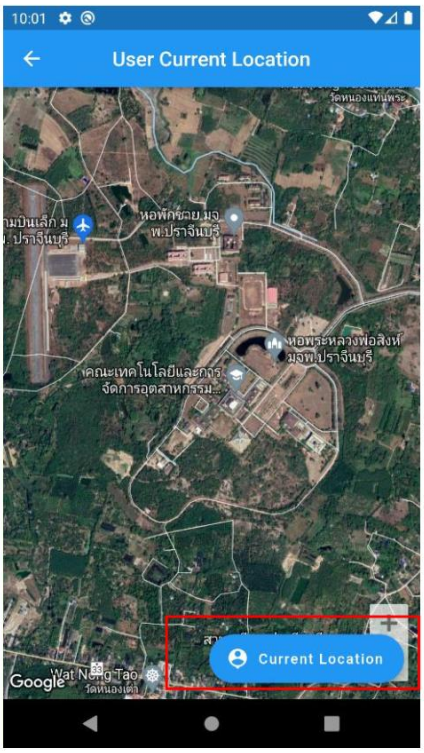
```yaml
dependencies:
  flutter:
    sdk: flutter

  # The following adds the Cupertino Icons font to your
  # Use with the CupertinoIcons class for iOS style icon
  cupertino_icons: ^1.0.8
  google_maps_flutter: ^2.10.0
  geolocator: ^13.0.2
```

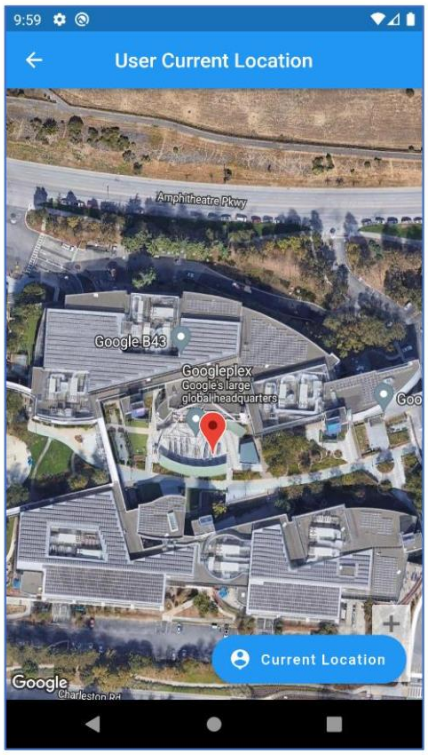
**** ติดตั้ง geolocator:^13.0.2**

lib > screens > current_location_screen.dart > _CurrentLocationState > build

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

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

  @override
  State<CurrentLocation> createState() => _CurrentLocationState();
}

class _CurrentLocationState extends State<CurrentLocation> {
  late GoogleMapController googleMapController;

  static const CameraPosition _initialCameraPosition = CameraPosition(
      target: LatLng(37.42796133580664, -122.085749655962), zoom: 14.4746); // CameraPosition

  Set<Marker> markers = {};
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("User Current Location"),
        centerTitle: true,
      ), // AppBar
      body: GoogleMap(
        markers: markers,
        mapType: MapType.hybrid,
        initialCameraPosition: _initialCameraPosition,
        onMapCreated: (GoogleMapController controller) {
          googleMapController = controller;
        },
      ), // GoogleMap
      floatingActionButton: FloatingActionButton.extended(
        onPressed: () async {
          Position position = await _determinePosition();
          googleMapController.animateCamera(CameraUpdate.newCameraPosition(
              CameraPosition(
                  target: LatLng(position.latitude, position.longitude),
                  zoom: 14))); // CameraPosition
          markers.clear();
          markers.add(Marker(
              markerId: MarkerId("Cerrent Location"),
              position: LatLng(position.latitude, position.longitude))); // Marker
          setState(() {});
        },
        label: const Text("Current Location"),
        icon: const Icon(Icons.location_history),
      ), // FloatingActionButton.extended
    ); // Scaffold
  }

  Future<Position> _determinePosition() async {
    bool serviceEnable;
    LocationPermission permission;

    serviceEnable = await Geolocator.isLocationServiceEnabled();
    if (!serviceEnable) {
      return Future.error("Location Service Disable");
    }

    permission = await Geolocator.checkPermission();
    if (permission == LocationPermission.denied) {
      permission = await Geolocator.requestPermission();

      if (permission == LocationPermission.denied) {
        return Future.error("Location Permission Denied");
      }
    }

    if (permission == LocationPermission.deniedForever) {
      return Future.error("Location Permission are permanently denied");
    }

    Position position = await Geolocator.getCurrentPosition();
    return position;
  }
}
```

ทดสอบด้วย Emulator เมื่อกดปุ่ม Current Location จะแสดงตำแหน่ง ที Googleplex





ให้ทำการ Build apk แล้วติดตั้งบนอุปกรณ์จริง เพื่อทดสอบการทำงาน

เมื่อกดปุ่ม Current Location จะแสดงตำแหน่งของอุปกรณ์