

**Student Name: Jotee Chavishka**  
**Lecturer: Mr Beeharry Shiam**  
**Module: PMH**  
**INFO 3**

## Task 1 – Adding Google Maps to a Flutter App

Using the link provided in this assignment for this specific task, the command line tool is used in Android Studio to get started with adding Google Maps to a flutter app. This is done by resolving dependencies pertaining to google maps plugins.

Adding Google Maps to a Flutter app

1 Introduction

2 Set up your Flutter environment

3 **Getting started**

4 Adding Google Maps to the app

5 Put Google on the Map

6 Next steps

### 3. Getting started

#### Getting started with Flutter

The easiest way to get started with Flutter is to use the flutter command line tool to create all the required code for a simple getting started experience.

```
$ flutter create google_maps_in_flutter --platforms android,ios,web
Creating project google_maps_in_flutter...
Running "flutter pub get" in google_maps_in_flutter... 1,612ms
Wrote 80 files.

All done!
In order to run your application, type:

$ cd google_maps_in_flutter
$ flutter run

Your application code is in google_maps_in_flutter/lib/main.dart.
```

The screenshot shows the Android Studio interface with the 'Terminal' tab active. The terminal displays the output of the command `flutter create google_maps_in_flutter --platforms android,ios,web`. The output indicates that the project was created successfully, dependencies were resolved, and 81 files were written. It also provides links to Flutter documentation and instructions on how to run the application.

```
PS C:\Users\HP\StudioProjects\Gmaps> flutter create google_maps_in_flutter --platforms android,ios,web
Creating project google_maps_in_flutter...
Resolving dependencies in google_maps_in_flutter... (1.9s)
Got dependencies in google_maps_in_flutter.
Wrote 81 files.

All done!
You can find general documentation for Flutter at: https://docs.flutter.dev/
Detailed API documentation is available at: https://api.flutter.dev/
If you prefer video documentation, consider: https://www.youtube.com/c/flutterdev

In order to run your application, type:

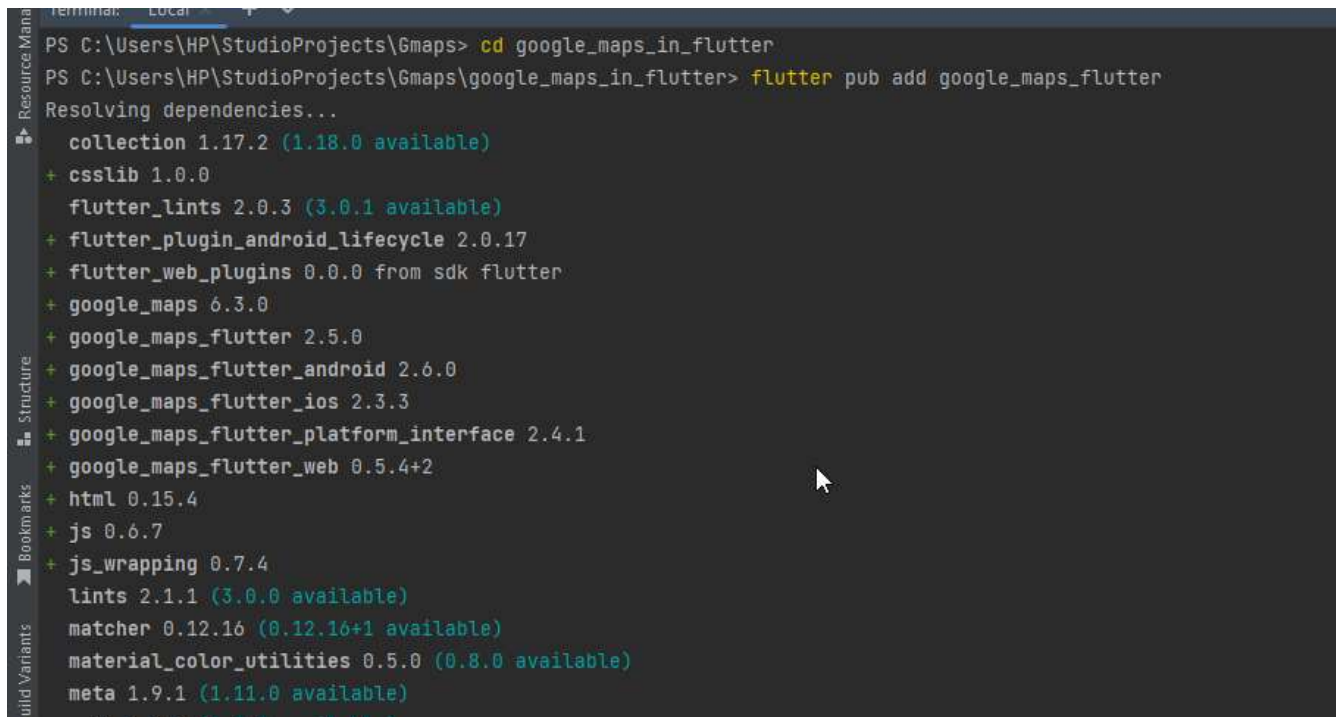
$ cd google_maps_in_flutter
$ flutter run

Your application code is in google_maps_in_flutter\lib\main.dart.

In order to run your application, type:

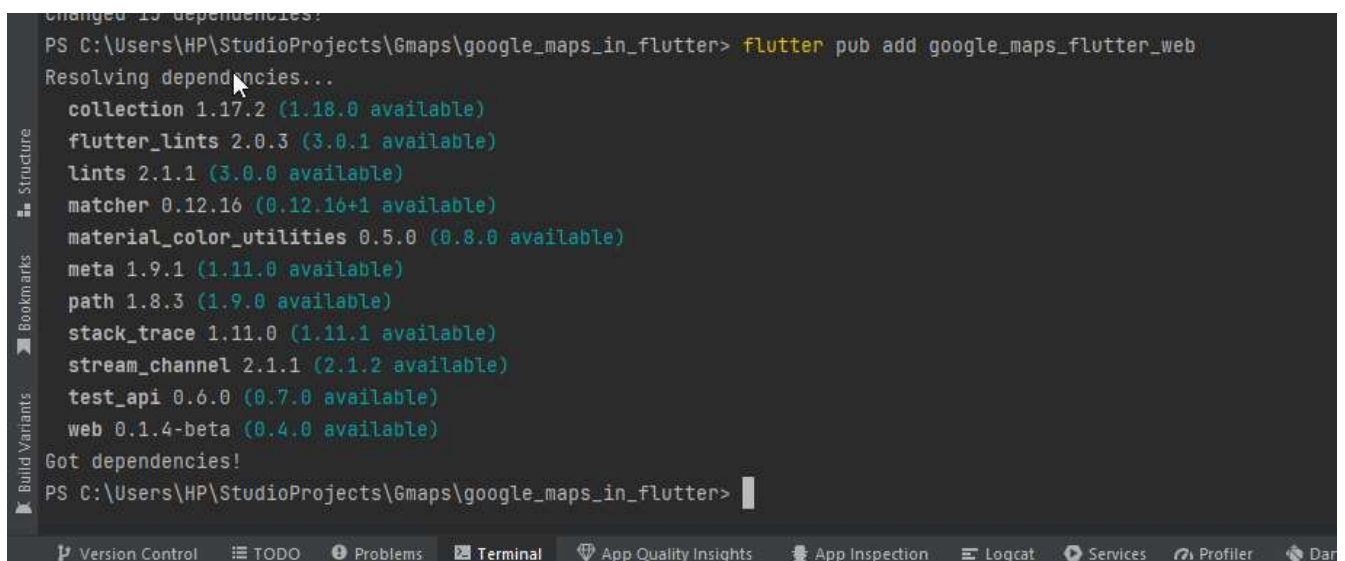
$ cd google_maps_in_flutter
```

The following commands are used to specify and introduce the **google\_maps\_flutter** plugin.



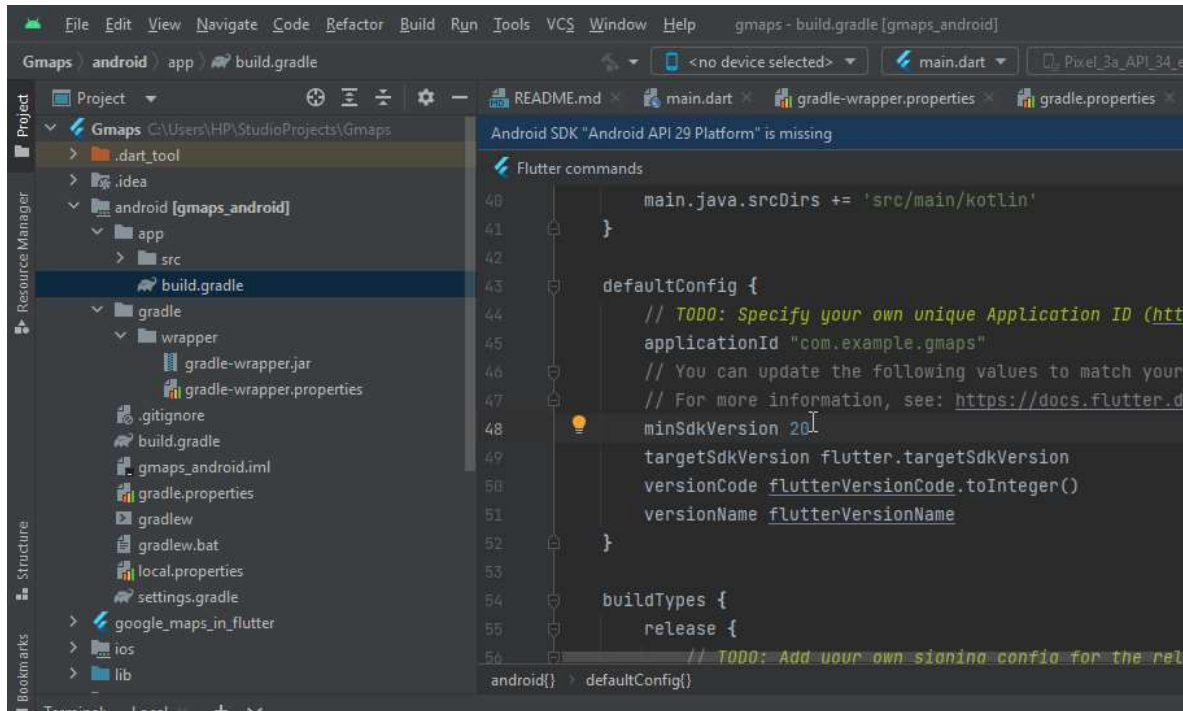
```
PS C:\Users\HP\StudioProjects\Gmaps> cd google_maps_in_flutter
PS C:\Users\HP\StudioProjects\Gmaps\google_maps_in_flutter> flutter pub add google_maps_flutter
Resolving dependencies...
+ collection 1.17.2 (1.18.0 available)
+ csslib 1.0.0
+ flutter_lints 2.0.3 (3.0.1 available)
+ flutter_plugin_android_lifecycle 2.0.17
+ flutter_web_plugins 0.0.0 from sdk flutter
+ google_maps 6.3.0
+ google_maps_flutter 2.5.0
+ google_maps_flutter_android 2.6.0
+ google_maps_flutter_ios 2.3.3
+ google_maps_flutter_platform_interface 2.4.1
+ google_maps_flutter_web 0.5.4+2
+ html 0.15.4
+ js 0.6.7
+ js_wrapping 0.7.4
+ lints 2.1.1 (3.0.0 available)
+ matcher 0.12.16 (0.12.16+1 available)
+ material_color_utilities 0.5.0 (0.8.0 available)
+ meta 1.9.1 (1.11.0 available)
```

While this codelab teaches us how to use Google Maps in Flutter for web, an additional plugin needs to be installed specifically for web support as it is not yet integrated into the main plugin.



```
Changed 10 dependencies:
PS C:\Users\HP\StudioProjects\Gmaps\google_maps_in_flutter> flutter pub add google_maps_flutter_web
Resolving dependencies...
+ collection 1.17.2 (1.18.0 available)
+ flutter_lints 2.0.3 (3.0.1 available)
+ lints 2.1.1 (3.0.0 available)
+ matcher 0.12.16 (0.12.16+1 available)
+ material_color_utilities 0.5.0 (0.8.0 available)
+ meta 1.9.1 (1.11.0 available)
+ path 1.8.3 (1.9.0 available)
+ stack_trace 1.11.0 (1.11.1 available)
+ stream_channel 2.1.1 (2.1.2 available)
+ test_api 0.6.0 (0.7.0 available)
+ web 0.1.4-beta (0.4.0 available)
Got dependencies!
PS C:\Users\HP\StudioProjects\Gmaps\google_maps_in_flutter>
```

Next, the android/app/build.gradle is modified as follows by setting the minSDK to 20.



The API key is then generated for google maps on the google maps platform found online.

