

UNIT-5

Publishing and Exporting app and Advanced feature

❖ **Working with location and maps**

- assumes you already have the maps set up in your project using the Google Maps Flutter Package and your own Google Maps API key. If not, follow this link on how to set up your Flutter project to work with Google Maps. Other dependencies include the Flutter Polyline Points package and the Flutter Location Plugin.

➤ **Initial Setup**

Make sure you prep your environment accordingly to enable location tracking on both IOS and Android by following the steps in the package's README regarding the Android manifest file and the iOS Info.plist.

Once set up, the dependences look like

dependencies:

```
flutter:  
  sdk: flutter  
cupertino_icons: ^1.0.2  
flutter_polyline_points: ^1.0.0  
google_maps_flutter: ^2.1.7  
location: ^4.4.0
```

➤ **Google Map**

Create a StatefulWidget called OrderTrackingPage with its corresponding State class, where I imported the required packages as well as some hardcoded source and destination location

```
import 'dart:async';  
import 'package:flutter/material.dart';  
import 'package:google_maps_flutter/google_maps_flutter.dart';  
class OrderTrackingPage extends StatefulWidget {  
  const OrderTrackingPage({Key? key}) : super(key: key);
```

```

@override
State<OrderTrackingPage> createState() => OrderTrackingPageState();
}
class OrderTrackingPageState extends State<OrderTrackingPage> {
  final Completer<GoogleMapController> _controller = Completer();
  static const LatLng sourceLocation = LatLng(37.33500926, -122.03272188);
  static const LatLng destination = LatLng(37.33429383, -122.06600055);
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: ... GoogleMap widget will be here ...,
    );
  }
}

```

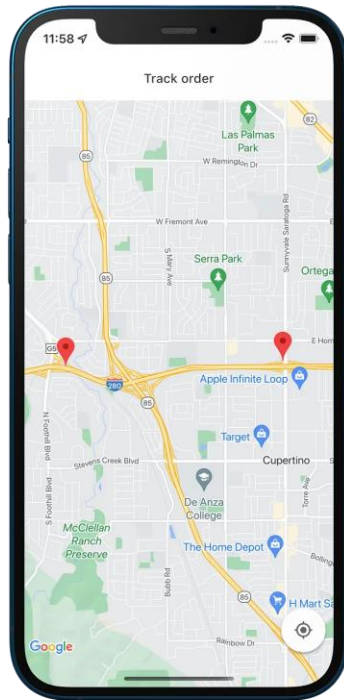
Create the `GoogleMap` widget and set the `initialCameraPosition` to the location of the source. The map needs to be zoomed in a bit, so set it to 13.5.

We need a marker/pin to understand the exact location. Define a marker and set its position to the source location. For the destination, add another marker/pin.

```

GoogleMap(
  initialCameraPosition: const CameraPosition(
    target: sourceLocation,
    zoom: 13.5,
  ),
  markers: {
    const Marker(
      markerId: MarkerId("source"),
      position: sourceLocation,
    ),
    const Marker(
      markerId: MarkerId("destination"),
      position: destination,
    ),
  },
  onMapCreated: (mapController) {
    _controller.complete(mapController);
  },
),

```



➤ Draw Route Direction

The next thing I want to do is draw a line from destination to source. Create an empty list called `polylineCoordinates`. Create an instance of `PolylinePoints` and an async function called `getPolyPoints`. The method `getRouteBetweenCoordinates` returns the list of polyline points. The Google API key, source, and destination locations were required. If the points are not empty, we store them to `polylineCoordinates`.

```
List<LatLng> polylineCoordinates = [];
void getPolyPoints() async {
  PolylinePoints polylinePoints = PolylinePoints();
  PolylineResult result = await polylinePoints.getRouteBetweenCoordinates(
    google_api_key, // Your Google Map Key
    PointLatLng(sourceLocation.latitude, sourceLocation.longitude),
    PointLatLng(destination.latitude, destination.longitude),
  );
  if (result.points.isNotEmpty) {
    result.points.forEach(
      (PointLatLng point) => polylineCoordinates.add(
        LatLng(point.latitude, point.longitude),
      ),
    );
  }
}
```

```

        setState({});
    }
}

```

- On initState call getPolyPoints

```

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

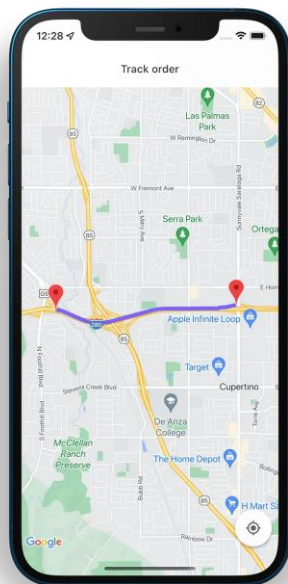
```

- Back to the GoogleMap widget, define the polylines.

```

GoogleMap(
  ...
  polylines: {
    Polyline(
      polylineId: const PolylineId("route"),
      points: polylineCoordinates,
      color: const Color(0xFF7B61FF),
      width: 6,
    ),
  },
),

```



❖ **Telephony and SMS**

- A Flutter plugin fork from telephony to use telephony features such as fetch network info, start phone calls, send and receive SMS, and fixed for listen for incoming SMS.

➤ **Initial Setup**

First we need to add the telephony dependency in the pubspec.yaml file.

dependencies:

flutter:

 sdk: flutter

 url_launcher: #latest_version

 flutter_phone_direct_caller: #latest_version

➤ **Move To Main.dart For Navigate**

```
import 'package:call_sms/CallSms.dart';
```

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

```
void main(){  
  runApp(MaterialApp(  
    debugShowCheckedModeBanner: false,  
    title: 'CallSmS',  
    home: CallSms(),  
  ));  
}
```

➤ **Configuration for CallSms.dart**

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

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

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

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

```
  @override
```

```
  State<CallSms> createState() => _CallSmsState();
```

```
}
```

```
Uri calldailer=Uri(scheme: 'tel',path: '9727238157');
```

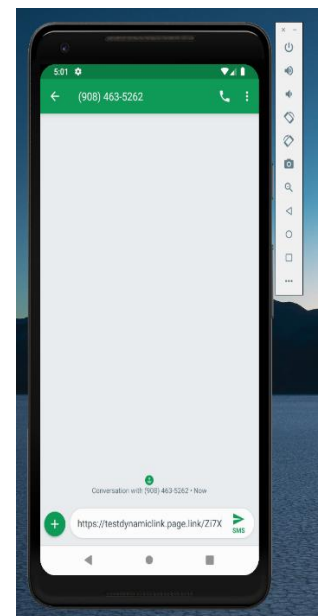
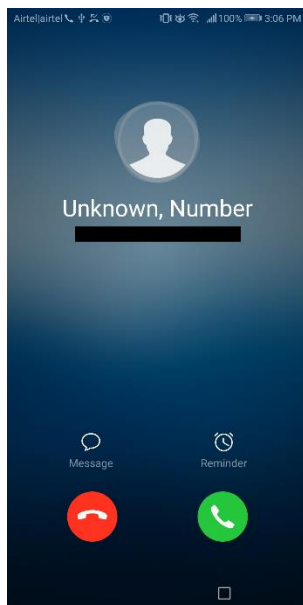
```

CallS()async{
  await launchUrl(calldailer);
}

//directcall()async{
  //await FlutterPhoneDirectCaller.callNumber('9727238157');
  //}

class _CallSmsState extends State<CallSms> {
  @override
  Widget build(BuildContext context) {
    return Container(
      child: Scaffold(
        body: Stack(
          children: [
            Center(child: ElevatedButton(child: Text('CALL'),onPressed: CallS,))
          ],
        ),
      ),
    );
  }
}

```



❖ **Video and using camera**

- We can add video from the gallery as well as from the camera using the image_picker package in Flutter. For this, you'll need to use your real device or web version. Follow the below steps to display the images from the gallery
- Add the dependency to your pubspec.yaml file

```
dependencies:  
  flutter:  
    sdk: flutter  
  cupertino_icons: ^1.0.2  
  image_picker: ^0.8.6+1
```

- Create variable for image picker and file

```
File? selectedImage;  
final picker = ImagePicker();
```

- **Move To Main.dart For Navigate**

```
import 'package:flutter/material.dart';  
import 'package:flutter_application_1/imagepicker.dart';  
  
void main(){  
  runApp(MaterialApp(  
    debugShowCheckedModeBanner: false,  
    title: 'ImagePicker',  
    home:imagePicker() ,  
  ));  
}
```

- **Configuration for imagePicker.dart**

```
import 'dart:convert';  
import 'dart:io';  
import 'package:flutter/material.dart';  
import 'package:image_picker/image_picker.dart';  
  
class imagePicker extends StatefulWidget {  
  const imagePicker({super.key});
```

```

@override
State<imagePicker> createState() => _MyWidgetState();
}

class _MyWidgetState extends State<imagePicker> {
  File? selectedImage;
  //String BaseImage="";

  Future<void> chooseImage(type) async {
    // ignore: prefer_typing_uninitialized_variables
    var image;
    if (type == "camera") {
      image = await ImagePicker()
        .pickImage(source: ImageSource.camera);
    } else {
      image = await ImagePicker()
        .pickImage(source: ImageSource.gallery);
    }
    if (image != null) {
      setState() {
        selectedImage = File(image.path);
        //BaseImage = base64Encode(selectedImage!.readAsBytesSync());
        // won't have any error now
      });
    }
  }
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar:AppBar(title:
Text('ImagePicker'),backgroundColor:Colors.yellow,centerTitle: true,),
    body:Padding(
      padding: const EdgeInsets.all(10.0),
      child: ListView(
        children: [
          CircleAvatar(
            radius: 60,
            backgroundColor: Colors.red,
            child: Padding(
              padding: const EdgeInsets.all(8), // Border radius
              child: ClipOval(

```



```

        child: selectedImage != null
          ? Image.file(
              selectedImage!,
              fit: BoxFit.cover,
              height: 100,
              width: 100,
            )
          : Image.network(
              'https://cdn.pixabay.com/photo/2015/04/23/22/00/tree-736885_1280.jpg',
              fit: BoxFit.cover,
              height: 100,
              width: 100,
            ),
      ),
    ),
    ElevatedButton(
      onPressed: () {
        chooseImage("camera");
      },
      child: const Text(" Image from Camera"),
    ),
    ElevatedButton(
      onPressed: () {
        chooseImage("Gallery");
      },
      child: const Text("Image From Gallery"),
    ),
  ],
),
));
}
}

```

