Introduction to Flutter

Developing a simple mobile app
By Wan Muzaffar Wan Hashim

MDR Tech

Anak2ü

common room bangi

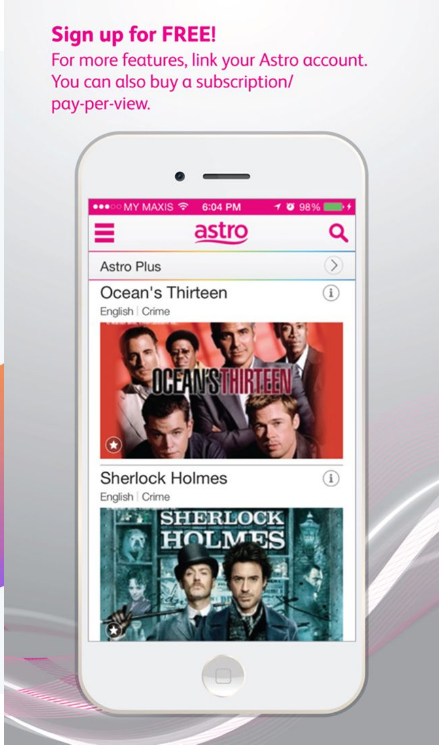
TMA
THE MOOSE ACADEMY

Muzaffar

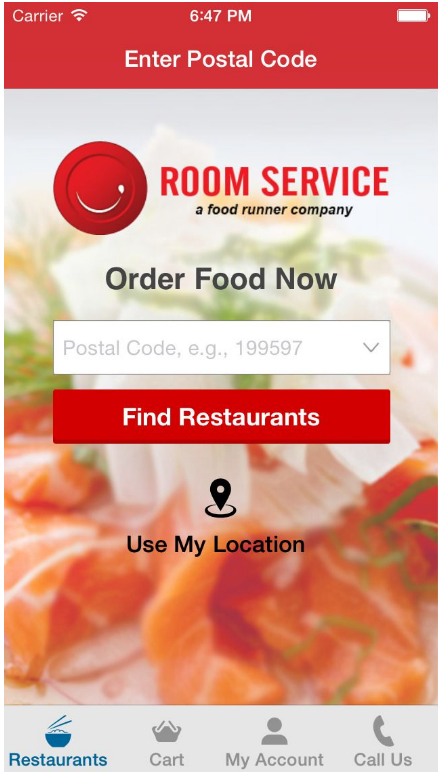
Founder of MDR-Tech, Co-founder of Anak2U

Worked with mobile industry since 2011

Different industry: M-Commerce, Newsfeed, Media Broadcasting, Food Delivery , Airline,Loyalty, Education.



Sign up for FREE!
For more features, link your Astro account.
You can also buy a subscription/
pay-per-view.



Enter Postal Code

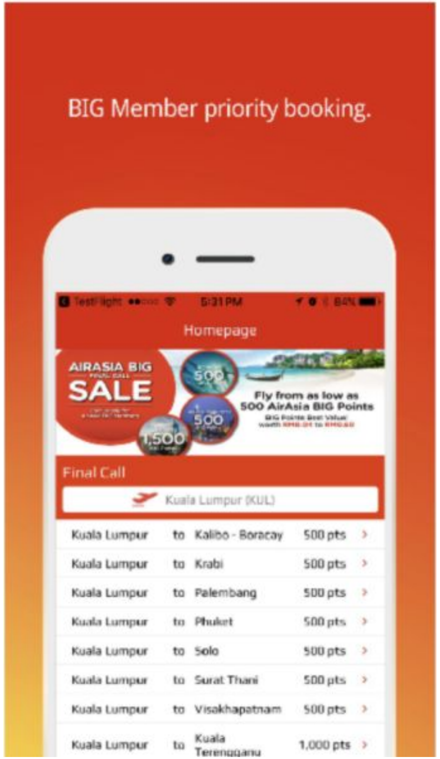
ROOM SERVICE
a food runner company

Order Food Now

Postal Code, e.g., 199597

Find Restaurants

Use My Location



BIG Member priority booking.

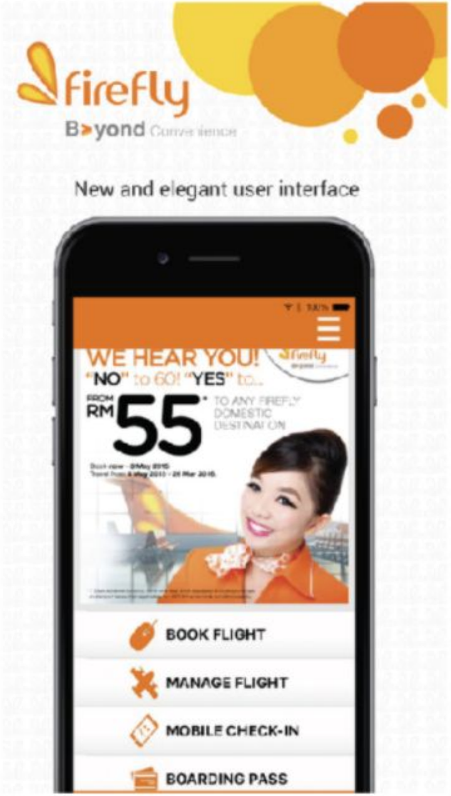
Homepage

AIRASIA BIG SALE
Fly from as low as 500 AirAsia BIG Points

Final Call

Kuala Lumpur (KUL)

Kuala Lumpur	to	Kalibo - Boracay	500 pts
Kuala Lumpur	to	Krabi	500 pts
Kuala Lumpur	to	Palembang	500 pts
Kuala Lumpur	to	Phuket	500 pts
Kuala Lumpur	to	Solo	500 pts
Kuala Lumpur	to	Surat Thani	500 pts
Kuala Lumpur	to	Visakhapatnam	500 pts
Kuala Lumpur	to	Kuala Terengganu	1,000 pts

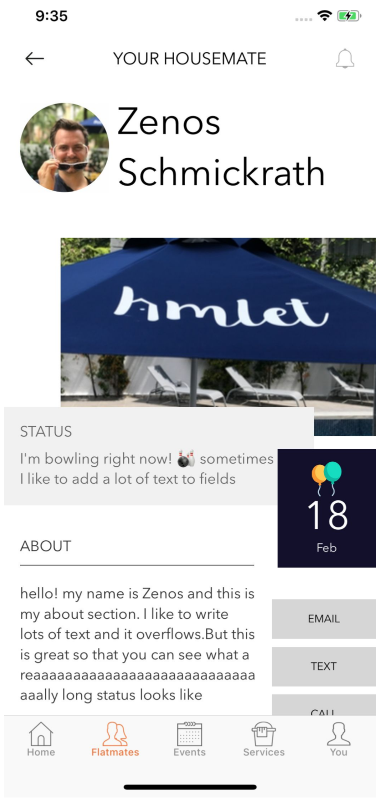


firefly
Beyond Convenience

New and elegant user interface

WE HEAR YOU! "NO" to 60! "YES" to...
FROM RM 55 TO ANY FIREFLY DOMESTIC DESTINATION

BOOK FLIGHT
MANAGE FLIGHT
MOBILE CHECK-IN
BOARDING PASS



9:35

YOUR HOUSEMATE

Zenos Schmickrath

STATUS

I'm bowling right now! 🎳 sometimes I like to add a lot of text to fields

18 Feb

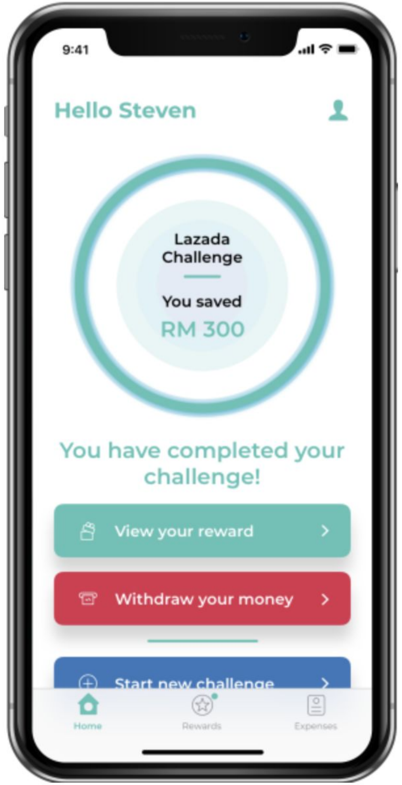
ABOUT

hello! my name is Zenos and this is my about section. I like to write lots of text and it overflows.But this is great so that you can see what a reaaaaaaaaaaaaaaaaaaaaaaaaaaaaaally long status looks like

EMAIL

TEXT

CALL



9:41

Hello Steven

Lazada Challenge

You saved

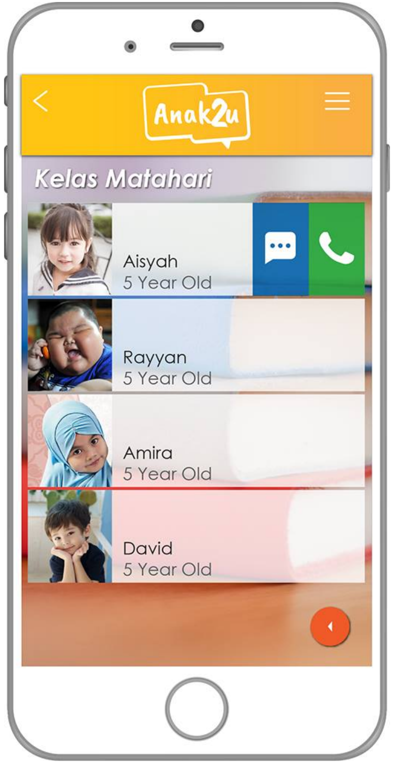
RM 300

You have completed your challenge!

View your reward ›

Withdraw your money ›

Start new challenge ›



Anak2u

Kelas Matahari

Aisyah
5 Year Old

Rayyan
5 Year Old

Amira
5 Year Old

David
5 Year Old

AMIC

aerospace malaysia innovation centre

AIRBUS

Mobile App Development

- A mobile application is a software application designed to run on smartphones, tablet computers and other mobile devices.
- Users on smartphones typically check the news, weather, email or their social networks. They have a choice between the mobile web version or a specially-created mobile app.

Mobile App Dev: Current State

Native Development	**Crossplatform Development**
● Android - *Kotlin* or Java ● iOS - *Swift* or Objective C	● Flutter - Dart - Bridge to native code (bring out native element) - 3 tahun - 1 code for all platform (android, ios, web, desktop..) ● React Native / ReactJs - JS - Bridge to native code (bring out native element) , stable 2017, instagram, facebook, facebook messenger ● Ionic - JS - Webview

Mobile App Types

- **Native**
  - Programmed using Swift/Objective C on the iPhone or using Java/Kotlin on Android devices.
- **Hybrid**
  - Mix between these two types of mobile applications.
  - Normally based on web programming language, eg: HTML, CSS, Javascript, Dart
  - Built once to be run on Android and iOS.
- **Web Apps / Progressive Web Apps. (selangkah) -> (Add to home screen)**
  - Web based.
    Runs in the phone’s browser.
  - Can have native features based on HTML5

What is Flutter

Open source UI Framework by Google

Able to create iOS, Android and web application using Dart

High performance, high fidelity, low latency, as it renders **the Native UI.**

Use DART as main programming language

Open source / github.

About Dart

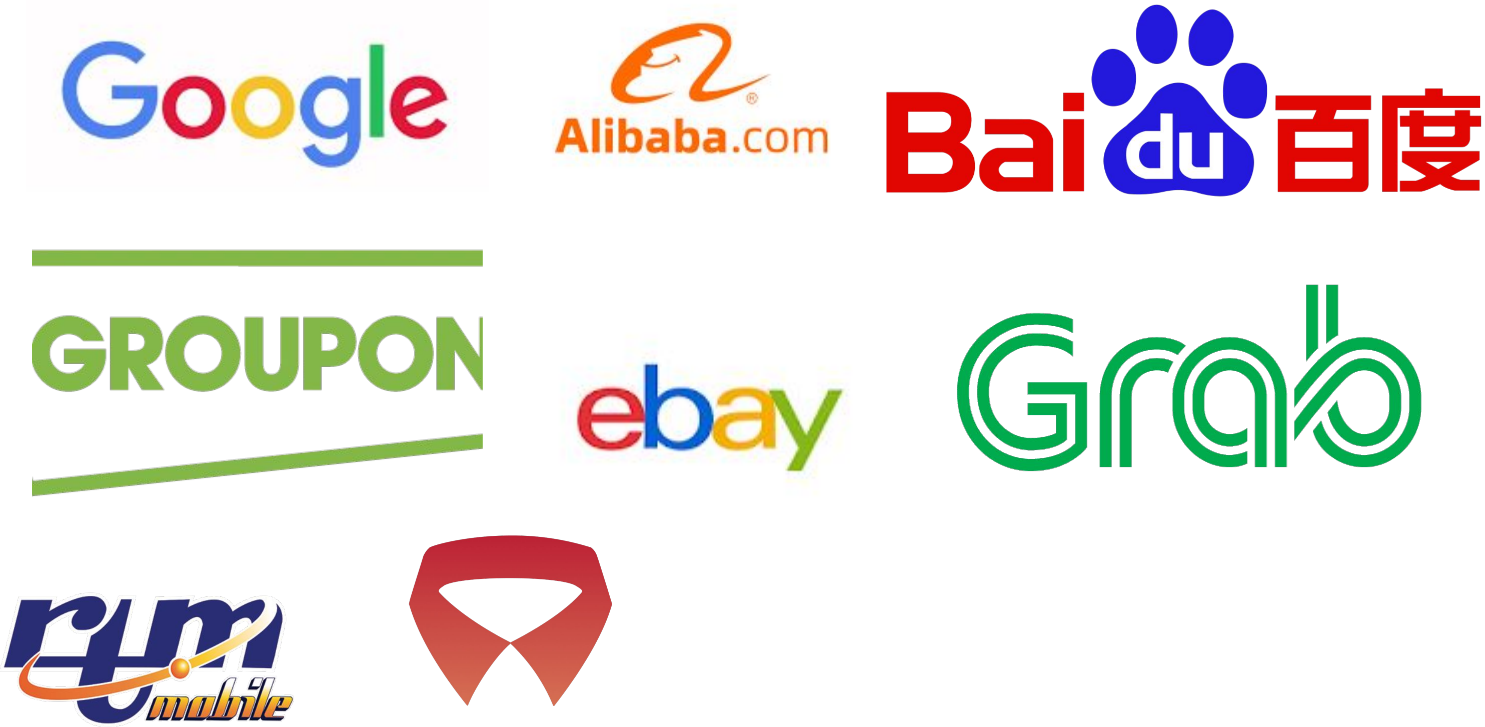
Dart is a programming language developed by Google

Learning it isn’t hard if you have experience with Java or JavaScript. You will quickly get it.

You can use dartpad as an online compiler of Dart

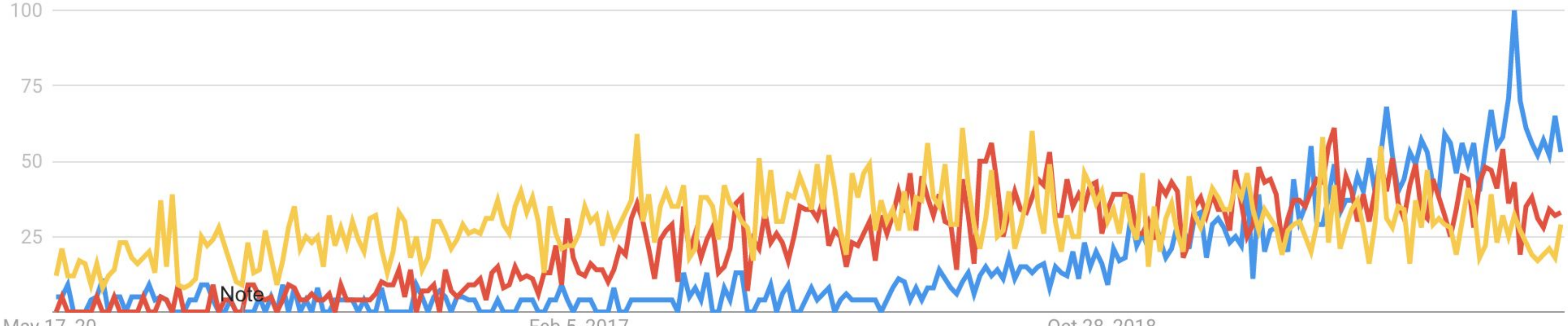
https://dartpad.dev/

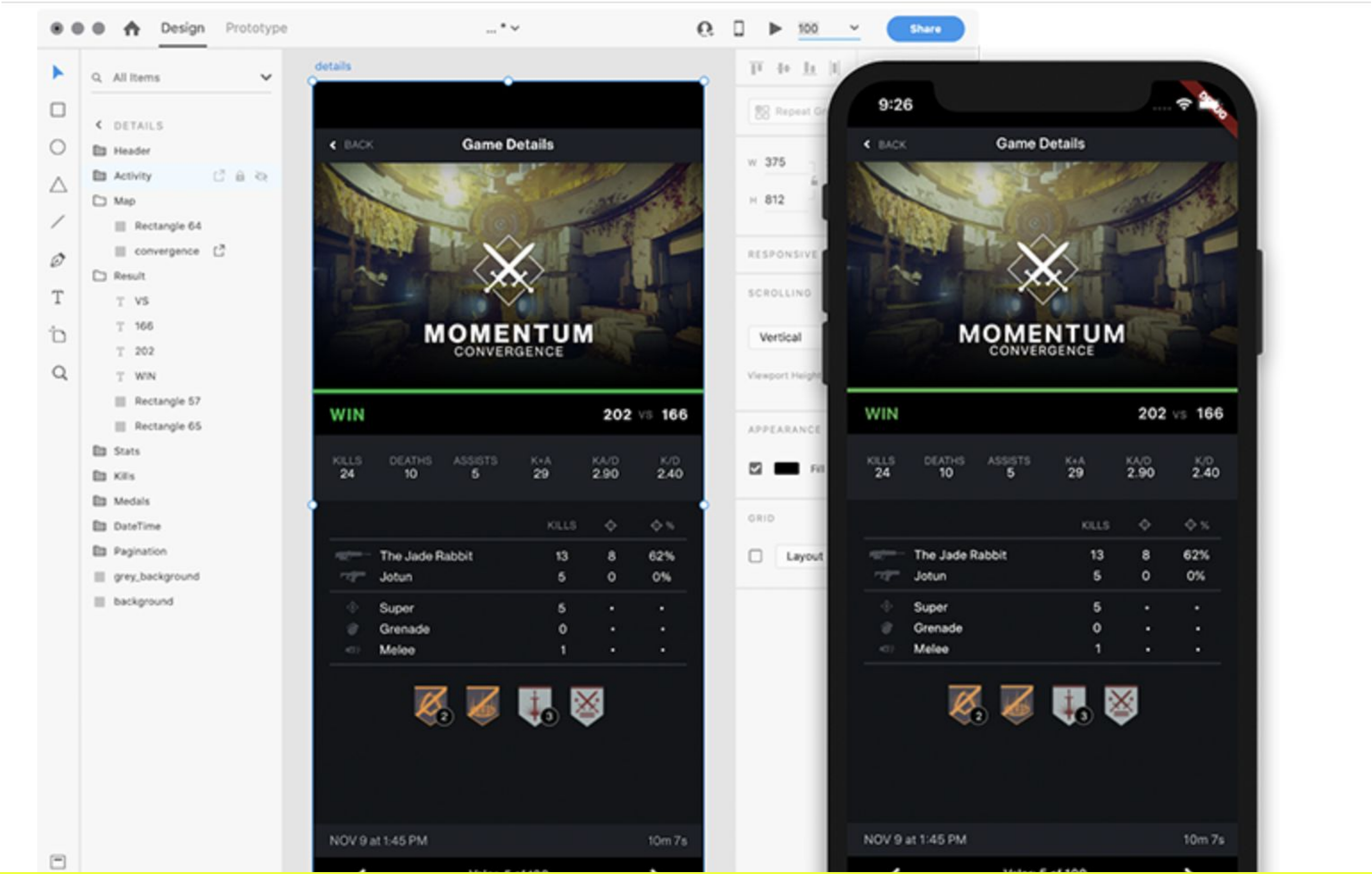
Who uses Flutter



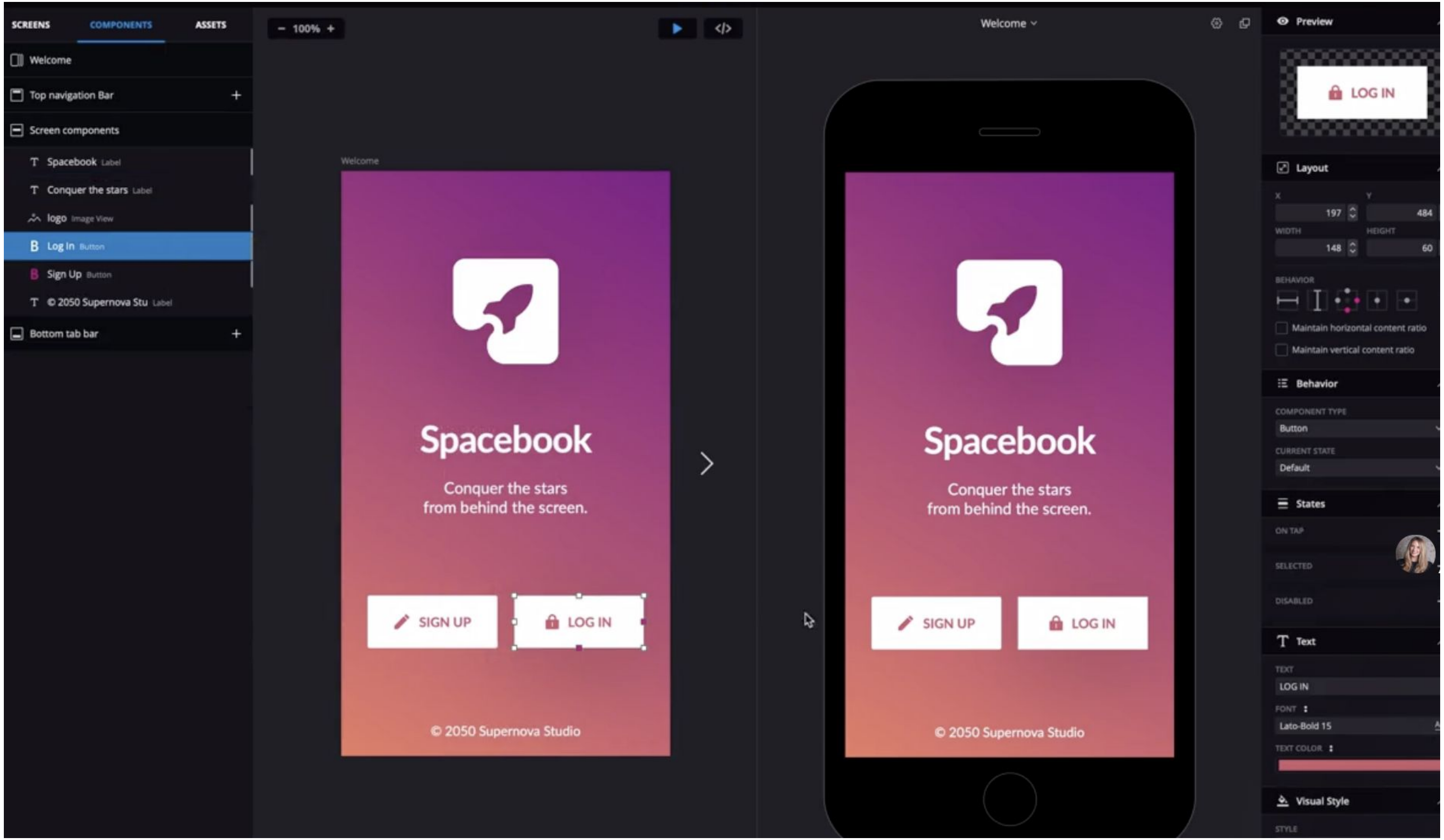
https://flutter.dev/showcase

Malaysia Google Trend (over 5 years)





Bridge gap between designer and developer - XD Flutter integration



X 197
Y 484
WIDTH 148
HEIGHT 60
BEHAVIOR
□ Maintain horizontal content ratio
□ Maintain vertical content ratio
COMPONENT TYPE Button
CURRENT STATE Default
ON TAP
SELECTED
DISABLED
TEXT LOG IN
FONT Lato-Bold 15
TEXT COLOR

Bridge gap between designer and developer - Supernova io

Setup your Editor

https://flutter.dev/docs/get-started/editor

You will need to configure an emulator after setting up the SDK.

Setting up Android Studio

1) Create New Project -> Select empty project -> Next -> Finish. Wait until gradle sync successfully
2) Select AVD Manager, Select a device (with Google Play logo), Download & Install OS (recommended Q and Above) -> Next (Finish)
3) Once AVD created, press Play
4) Select Run

Setting up Flutter

1) Go to Flutter.dev -> Docs -> Getting started
2) Select your OS and Download the installer file
3) Unzip the installer folder to a proper folder
4) Install Android Studio Flutter plugin
    a) File -> Settings -> Plugins -> Flutter
    b) A pop up will appear for confirmation to install Dart as well. Select Yes
    c) Restart IDE

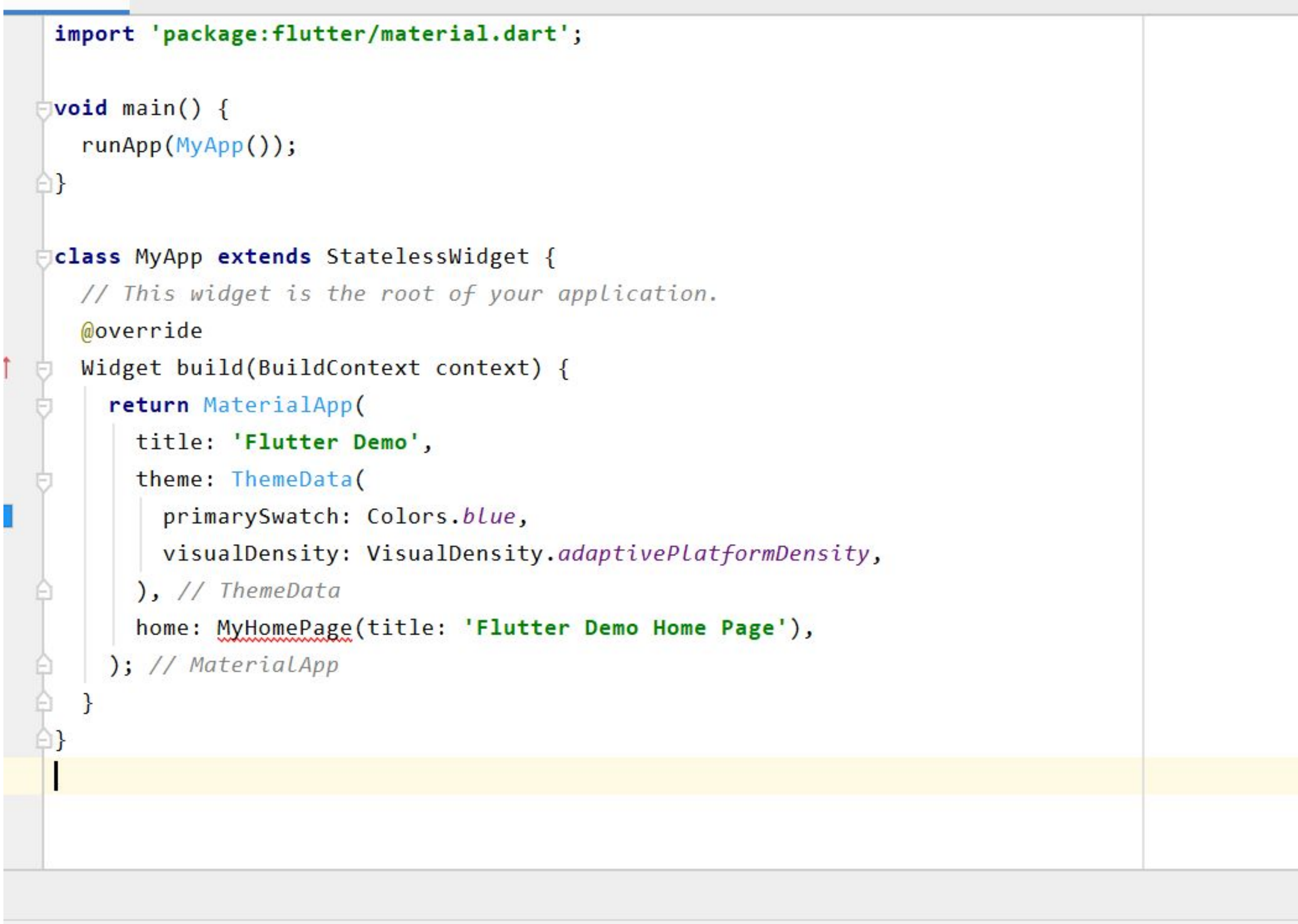
Upon restart new menu will appear Start new Flutter project

Native vs Crossplatform

Native	Crossplatform
2 code base - iOS - Android	1 codebase - Use the same code to compile in iOS and Android respectively
4 bulan	⅔ to ¾ of native time..
Stable.. This is the main source of truth	***Not as stable as native..*** A bridge to native.. If ios…., if android ... (library) Android -> file folder, ios no file..
Matured.. Already there since 2009 A lot of people and advocate, more questions answered in Stackoverflow Eg: Android Certified Developer, Google Expert (Android), Google Advocate. (work	New, between 3-5 years Not as much user as Native...

Getting started with Flutter

1) Create Flutter project
2) Point to flutter sdk
3) Add package name **= *reverse DNS* + *application name***



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

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

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
        visualDensity: VisualDensity.adaptivePlatformDensity,
      ), // ThemeData
      home: MyHomePage(title: 'Flutter Demo Home Page'),
    ); // MaterialApp
  }
}
```

The boilerplate code of an app - Scaffold

```
Scaffold(

    appBar: AppBar(

      title: const Text('Sample Code'),

    ),

    body: Center(child: Text('Hello World')),

    floatingActionButton: FloatingActionButton(

      onPressed: () => {},

      tooltip: 'Increment Counter',

      child: const Icon(Icons.add),

    ),

  );
```

Scaffold

A scaffold is a basic structure of an application having the following property by default:

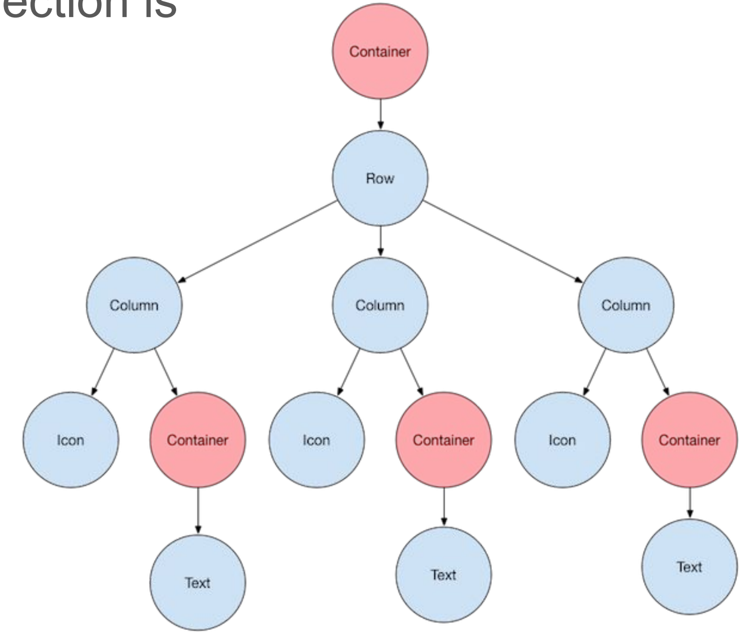
- appbar
- body
- floatingActionButton
- bottomNavigationBar
- drawer

Everything is a widget

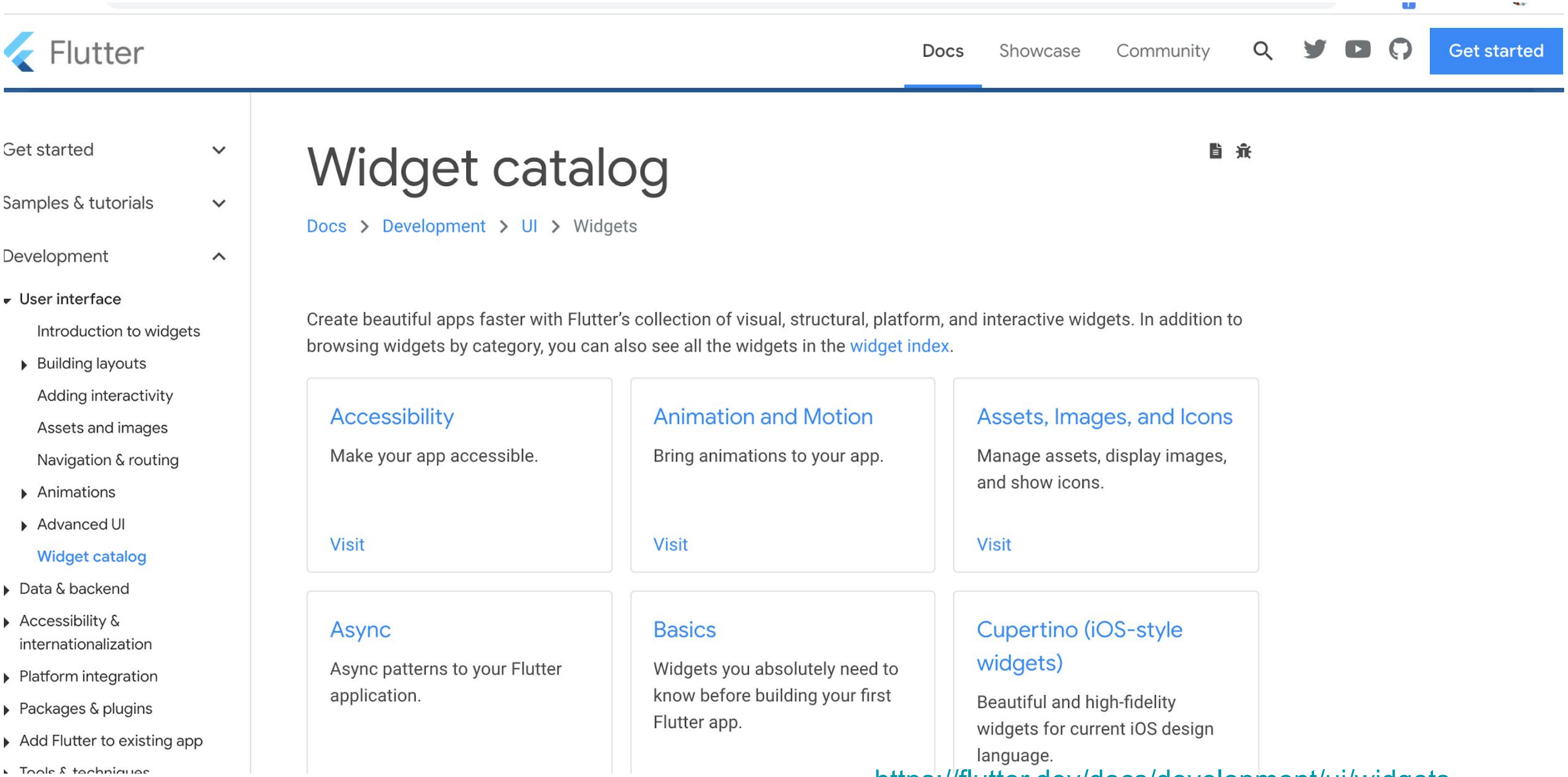
You build widget upon widget.

Your screen, a section in a screen, a tiny little section is also a Widget.

You create and customize your own widget.



Widget catalog



Widget catalog

Docs > Development > UI > Widgets

Create beautiful apps faster with Flutter's collection of visual, structural, platform, and interactive widgets. In addition to browsing widgets by category, you can also see all the widgets in the widget index.

Accessibility
Make your app accessible.
Visit

Animation and Motion
Bring animations to your app.
Visit

Assets, Images, and Icons
Manage assets, display images, and show icons.
Visit

Async
Async patterns to your Flutter application.

Basics
Widgets you absolutely need to know before building your first Flutter app.

Cupertino (iOS-style widgets)
Beautiful and high-fidelity widgets for current iOS design language.

https://flutter.dev/docs/development/ui/widgets

Widgets for layouting

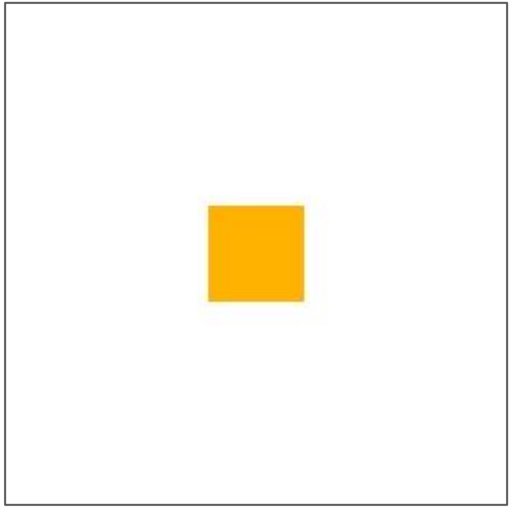
We will discover the widgets that are used to position items within a page. Here are some important/main widgets:

- Container
- Center
- Column
- Row
- SingleChildScrollView

Container

A container is a box! You can specify the width, height, color, padding and margin. In the below example, EdgeInsets.all means all direction (top, bottom, left, right)

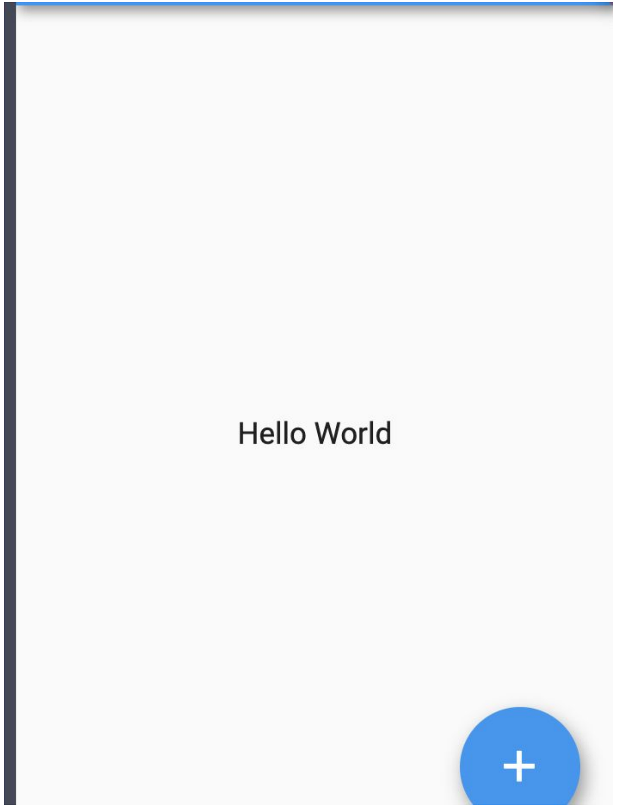
```
Center(
  child: Container(
    margin:  EdgeInsets.all(10.0),
    color: Colors.amber[600],
    width: 48.0,
    height: 48.0,
   padding:EdgeInsets.all(10.0)
  ),
)
```



Center

A widget that centers its child within itself.

```
Center(child: Text('Hello World')),
```

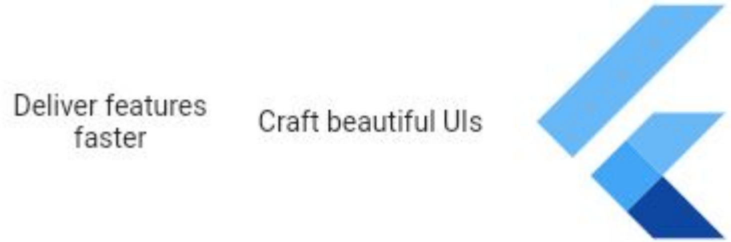


Hello World

Row

A widget that displays its children in a horizontal array.

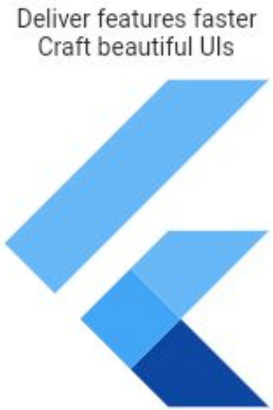
```
Row(
  children: <Widget>[
    Expanded(
      child: Text('Deliver features faster', textAlign:
TextAlign.center),
    ),
    Expanded(
      child: Text('Craft beautiful UIs', textAlign:
TextAlign.center),
    ),
    Expanded(
      child: FittedBox(
        fit: BoxFit.contain,
        child: const FlutterLogo(),
      ),
    ),
  ],
```



Column

A widget that displays its children in a vertical array.

```
Column(
  children: <Widget>[
    Text('Deliver features faster'),
    Text('Craft beautiful UIs'),
    Expanded(
      child: FittedBox(
        fit: BoxFit.contain,
        child: const FlutterLogo(),
      ),
    ),
  ],
)
```



Deliver features faster
Craft beautiful UIs

SingleChildScrollView

A box which allows a single widget to be scrolled.

You will use this when you have a single box that will normally be entirely visible, for example a clock face in a time picker, but you need to make sure it can be scrolled if the container gets too small in one axis

Center, Container	**Row and Column**
Have 1 child only	Can have more than one child (children)
child	children
Call Widget directly	Put widget inside Array []
	<Widget> - of type

Visible widget in Flutter

Once you know how to position items on a page, we will see some of the widgets that you can use in your application. Here are some important/main widgets:

- Text
- Image
- Button
- Icon
- Slider

Text

This widget is used to displays a text with single style.

You might need to use TextStyle widget as well with this widget to add styling to the text, for example to add color, set to bold

```
Text(

  'Hello World',

  textAlign: TextAlign.center,

  style: TextStyle(fontWeight: FontWeight.bold, color:Colors.red),

),
```

Image

To show an image. You may show an image from:

- Downloaded from a URL (Image.network)
- Stored locally in assets folder (Image.assets)

```
Image(

  image:
NetworkImage('https://flutter.github.io/assets-for-api-docs/assets/widgets/owl.jpg'),

)
```

RaisedButton

A ***raised button,*** follows Material design principle is a button that raises slightly, configurable via elevation property.

You will need to declare what should happen when the button is pressed via it’s onPress property.

Other type of button includes ***FlatButton***

```
RaisedButton(
                child: Text('Color Changed'),
                color: Colors.green,
                onPressed: () {
print(“Hello World”)},
              ),
```

Icon

As per its name, an icon is a widget that is predefined, and can be used directly within your application.

You may refer to Icon documentation, to see all available icon ready to be used in your application

```
Icon(

      Icons.audiotrack,

      color: Colors.green,

      size: 30.0,

    ),
```

https://api.flutter.dev/flutter/material/Icons-class.html

Slider

A slider can be used to select from either a continuous or a discrete set of values.

We will use onChanged property to update the value of item, once the value of slider changed.

```
Slider(
value: _value.toDouble(),
min: 1.0,
max: 10.0,
onChanged: (double newValue) {
setState(() {
_value = newValue.round();
});
},
```

Visible widget in Flutter

Once you know how to position items on a page, we will see some of the widgets that you can use in your application. Here are some important/main widgets:

- Text
- Image
- Button
- Icon
- Slider

Text

This widget is used to displays a text with single style.

You might need to use TextStyle widget as well with this widget to add styling to the text, for example to add color, set to bold

```
Text(

  'Hello World',

  textAlign: TextAlign.center,

  style: TextStyle(fontWeight: FontWeight.bold,
color:Colors.red),

),
```

Image

To show an image. You may show an image from:

- Downloaded from a URL
- Stored locally in assets folder

```
Image(

  image:
NetworkImage('https://flutter.github.io/assets-for-api
-docs/assets/widgets/owl.jpg'),

)
```

Icon

As per its name, an icon is a widget that is predefined, and can be used directly within your application.

You may refer to Icon documentation, to see all available icon ready to be used in your application

```
Icon(
    Icons.audiotrack,
    color: Colors.green,
    size: 30.0,
  ),
```

https://api.flutter.dev/flutter/material/Icons-class.html

RaisedButton

A raised button, follows Material design principle is a button that raises slightly, configurable via elevation property.

You will need to declare what should happen when the button is pressed via it’s onPress property.

Other type of button includes FlatButton

```
RaisedButton(
                child: Text('Color Changed'),
                color: Colors.green,
                onPressed: () {
print(“Hello World”)},
              ),
```

Slider

A slider can be used to select from either a continuous or a discrete set of values.

We will use onChanged property to update the value of item, once the value of slider changed.

```
Slider(
value: _value.toDouble(),
min: 1.0,
max: 10.0,
onChanged: (double newValue) {
setState(() {
_value = newValue.round();
});
},
```

Styling attributes

- Text - > style : TextStyle(color , fontSize, fontFamily)
- FlatButton, RaisedButton -> color, textColor
- Scaffold - > backgroundColor -> Change background color of the page

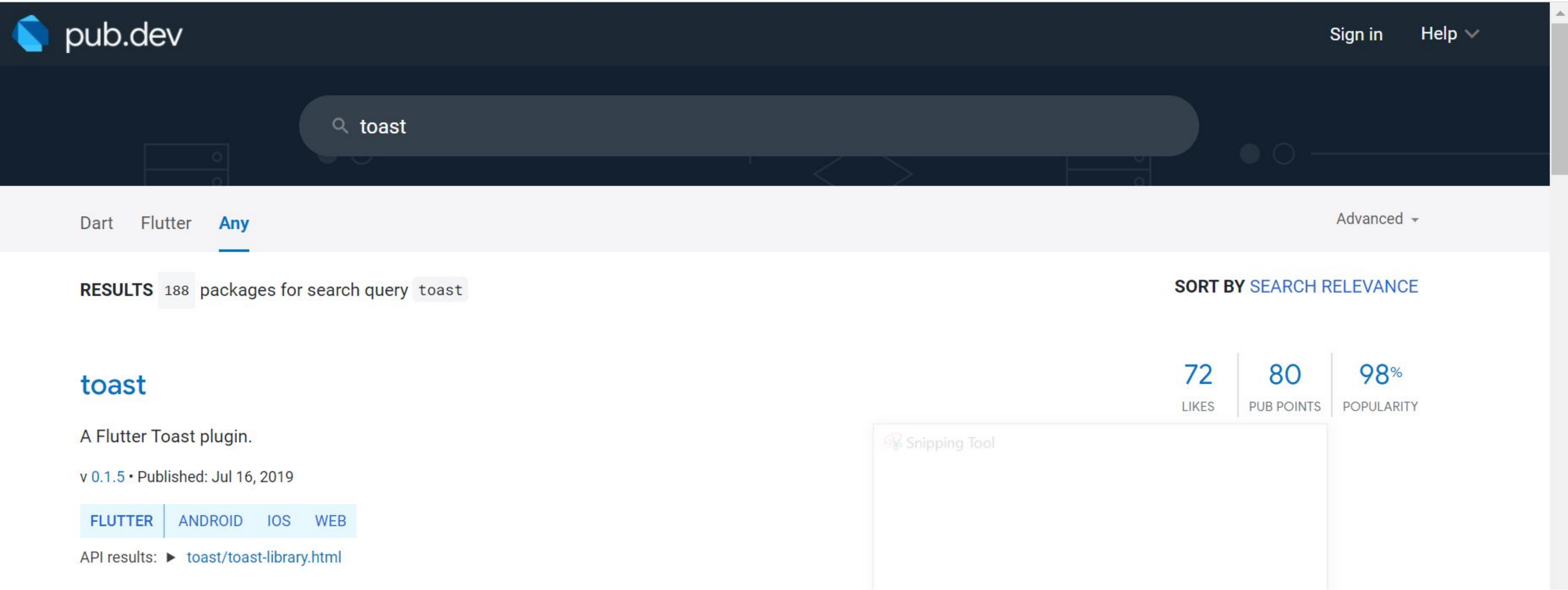
You can use Colors.green (Color name) or use ARGB Color.fromARGB() when defining color in the style.

For changing font, refer to the manual, there is an example to load font from Google Font. It will involve changing pubspec.yaml

pubspec.yaml

Define the sdk version (no need to change)

Define the dependencies (get it from pub.dev)



RESULTS `188` packages for search query `toast`

SORT BY SEARCH RELEVANCE

toast

72 LIKES | 80 PUB POINTS | 98% POPULARITY

A Flutter Toast plugin.

v 0.1.5 • Published: Jul 16, 2019

FLUTTER | ANDROID IOS WEB

API results: ▶ toast/toast-library.html

Add new library

1) Go to pub.dev
2) Look for the library of choice (Make sure support iOS/Android) and good rating + maintained
3) Copy the code
4) And put inside pubspec.yaml under dependencies. Verify the identaton is correct.

Stateless

Only to show UI and constant - About Us page..

stless + tab (Create a stateless widget)

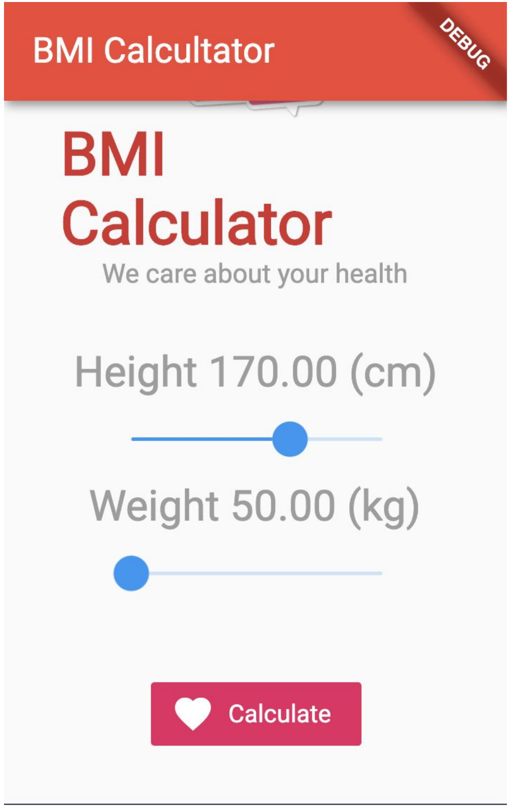
Stateful

Widget that manipulate data:

1) Normally page with ***form*** (TextInput, Slider..) is stateful, if you need to use setState in that page
2) page with API call is stateful, unless you are using StreamBuilder.

stful + tab -> Create a stateful widget

Demo - BMI Calculator



BMI Calcultator

BMI Calculator

We care about your health

Height 170.00 (cm)

Weight 50.00 (kg)

Demo

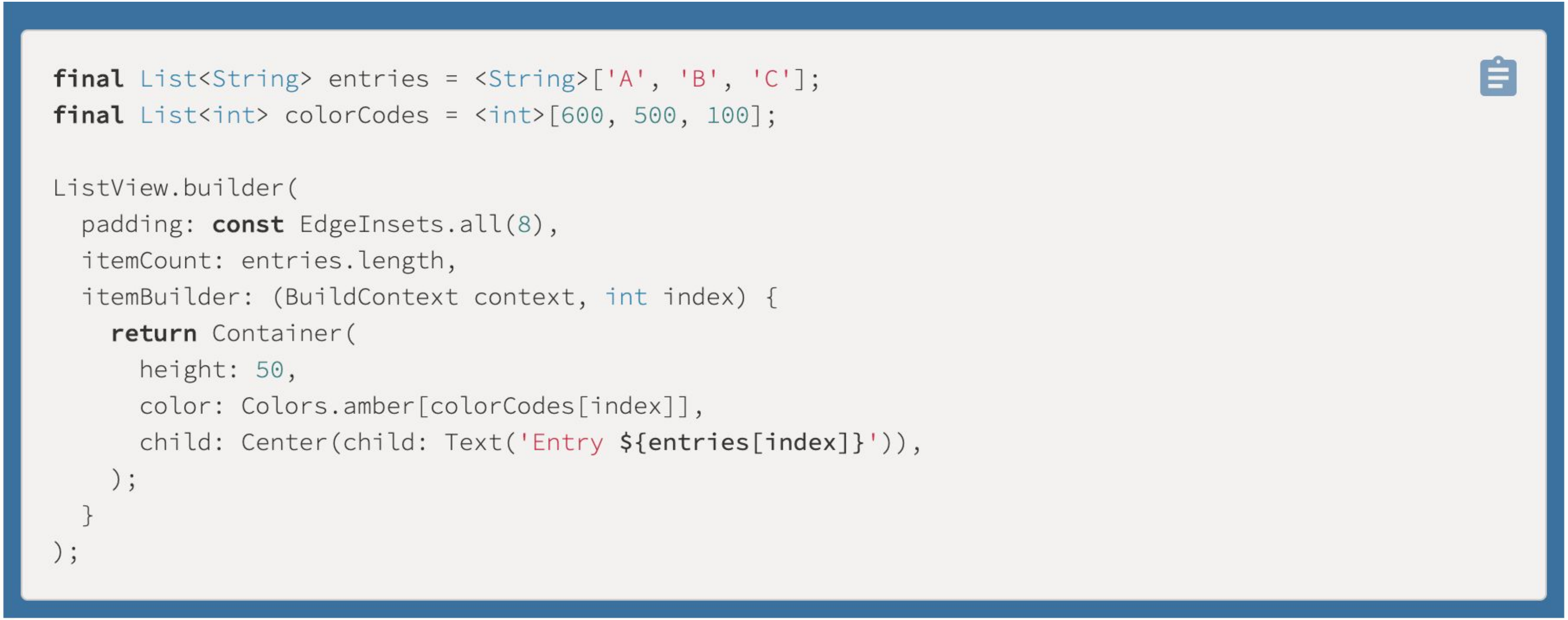
We will create a simple BMI calculator app that will calculate BMI based on height and weight entered by user.

- An application using stateful widget since we are storing height, weight and bmi
- Create the structure using scafffold
- Add Scrollview and container
- The container will contain a Column with:
    - Image (logo of our app)
    - App title and subtitle
- Two containers containing slider for user to choose height and weight
- Button when the button is pressed you will do the BMI calculation

Best practice when creating files

1) Create a new package, call it widgets
2) Create all widgets file (UI) inside components folder ..
3) You may also have different package for each widgets as you might have multiple files in one page

Creating a ListView

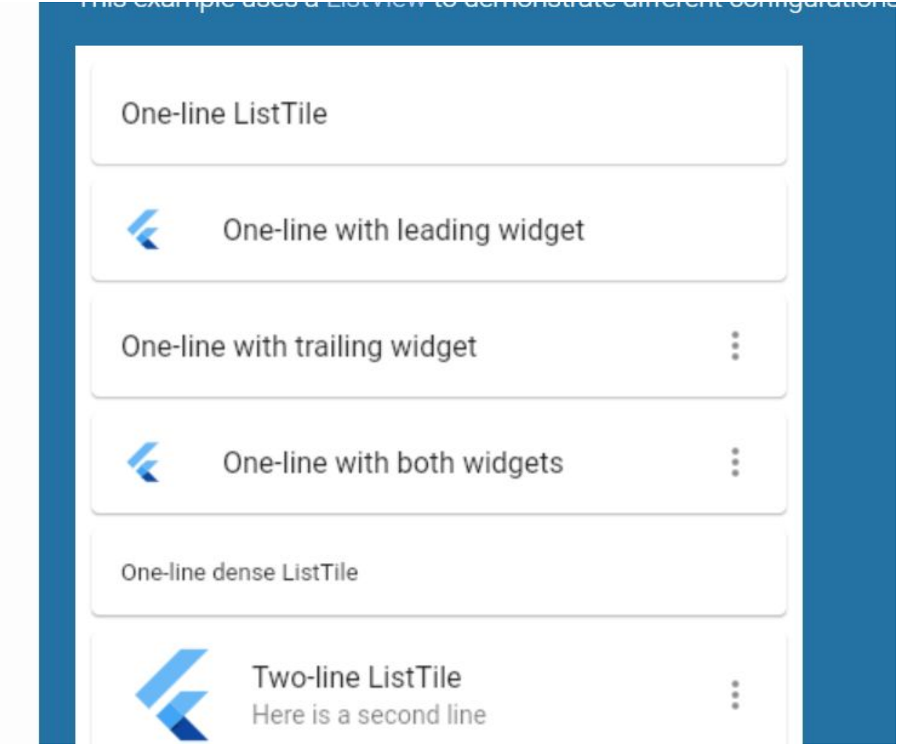


```dart
final List<String> entries = <String>['A', 'B', 'C'];
final List<int> colorCodes = <int>[600, 500, 100];

ListView.builder(
  padding: const EdgeInsets.all(8),
  itemCount: entries.length,
  itemBuilder: (BuildContext context, int index) {
    return Container(
      height: 50,
      color: Colors.amber[colorCodes[index]],
      child: Center(child: Text('Entry ${entries[index]}')),
    );
  }
);
```

Step creating a ListView

1) Get the reference from : https://api.flutter.dev/flutter/widgets/ListView-class.html
2) Create the Data source , in our case, we built a List of names
3) Get the second code from documentation, there are two important items:
    a) itemCount : How many rows are there? Normally it is the length of your List created in #2
    b) itemBuilder : What to show on each row

ListTile



This example uses a ListView to demonstrate different configurations

One-line ListTile

One-line with leading widget

One-line with trailing widget

One-line with both widgets

One-line dense ListTile

Two-line ListTile
Here is a second line

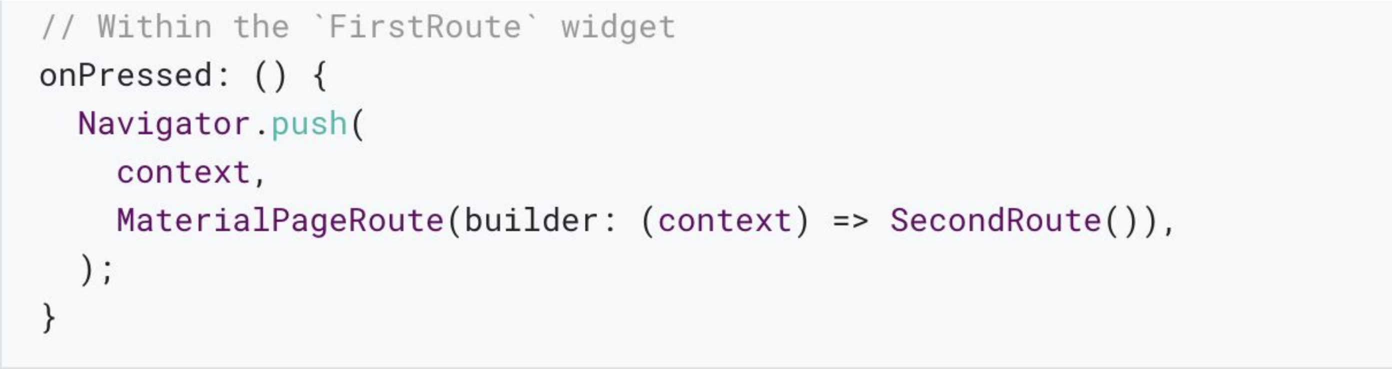
To facilitate the creation of List, you hava access to ListTile, a component that help you to create row , and by default will have:

- Title
- Subtitle
- Leading
- Trailing
- onTap

https://api.flutter.dev/flutter/material/ListTile-class.html

Navigation to a new screen

1) Create a new page, inside widget folders, call it add.dart and detail.dart
2) Inside the new page create a simple UI (Scaffold and Container body)
3) In the first page, for example upon button pressed create the code to open the second page as follows:



```
// Within the `FirstRoute` widget
onPressed: () {
  Navigator.push(
    context,
    MaterialPageRoute(builder: (context) => SecondRoute()),
  );
}
```

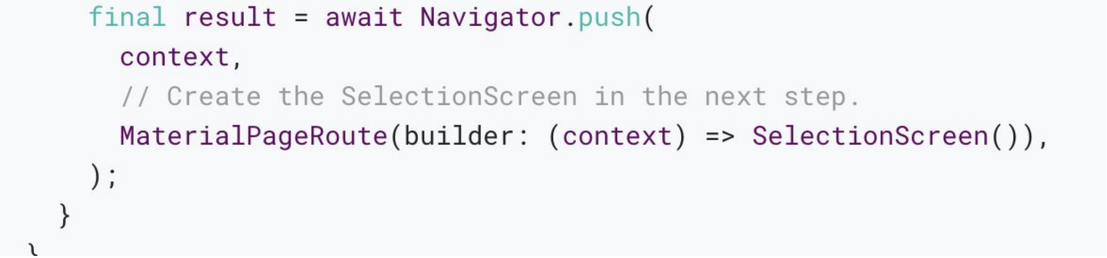
https://flutter.dev/docs/cookbook/navigation/navigation-basics

Passing data to second page

1) In second page (receiver) create a variable where you will retrieve the data
2) Create constructor with data retrieved in parameter.
3) Go back to first page, pass the data

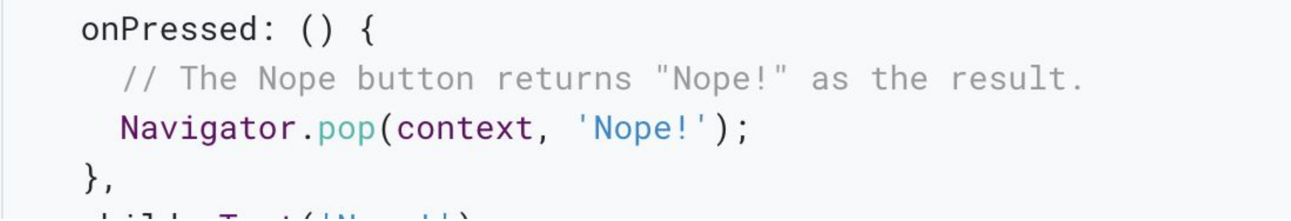
Passing data from second page to first page (1)

1) In the first page, (receiver) open the second page but this time you will add keyword async await indicating that you are waiting result from second page.



```
    final result = await Navigator.push(
      context,
      // Create the SelectionScreen in the next step.
      MaterialPageRoute(builder: (context) => SelectionScreen()),
    );
  }
```

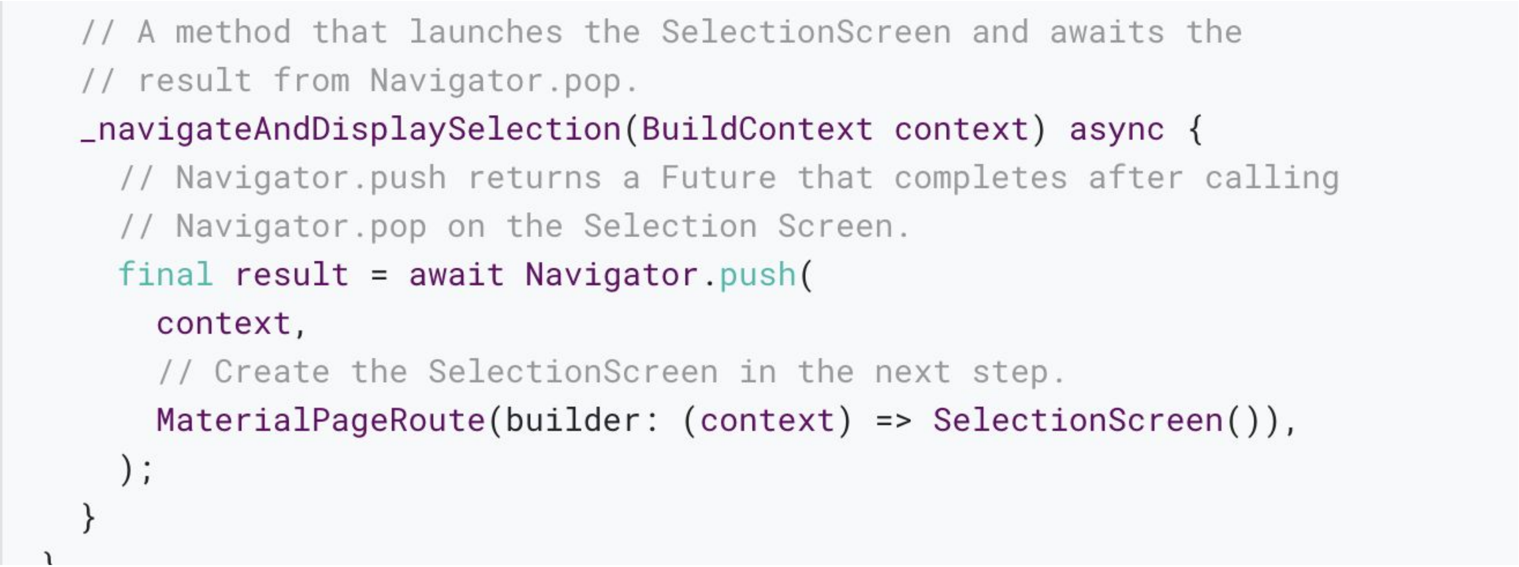
2) On second page, (sender), you will pass back the item using navigation.pop method, and pass it in the second parameter.



```
onPressed: () {
  // The Nope button returns "Nope!" as the result.
  Navigator.pop(context, 'Nope!');
},
```

Passing data from second page to first page (2)

3) Retrieve the data and perform operation with the data.



```
  // A method that launches the SelectionScreen and awaits the
  // result from Navigator.pop.
  _navigateAndDisplaySelection(BuildContext context) async {
    // Navigator.push returns a Future that completes after calling
    // Navigator.pop on the Selection Screen.
    final result = await Navigator.push(
      context,
      // Create the SelectionScreen in the next step.
      MaterialPageRoute(builder: (context) => SelectionScreen()),
    );
  }
}
```

Adding form in Flutter (TextField)

1) Create a TextField, and customized the textfield , Eg : Adding InputDecoration hint
2) For each textField, add TextEditingController and link it to each textfields.
3) Upon onPress for Example, retrieve the text entered by user by referring to the text property of the TextEditingController

https://api.flutter.dev/flutter/material/TextField-class.html

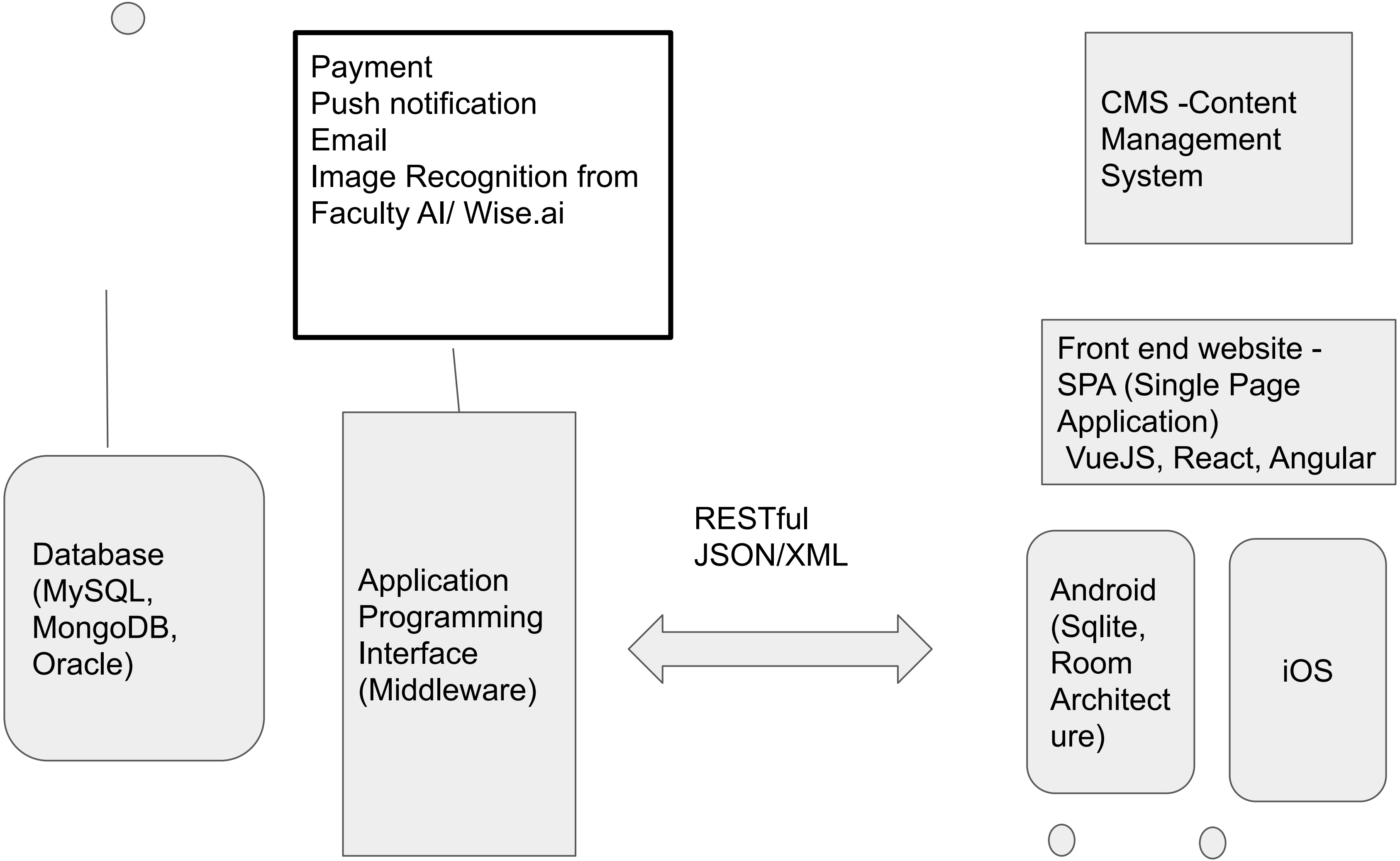
Storing Data

Locally

1) File -https://flutter.dev/docs/cookbook/persistence/reading-writing-files
2) Shared Preference : https://flutter.dev/docs/cookbook/persistence/key-value
3) Sqlite: https://flutter.dev/docs/cookbook/persistence/sqlite
4) Sqlite via Moor : https://medium.com/flutterdevs/moor-database-in-flutter-6a78d91b10e5

Remotely

1) REST API
2) Firebase



Payment
Push notification
Email
Image Recognition from Faculty AI/ Wise.ai

CMS -Content Management System

Front end website - SPA (Single Page Application)
VueJS, React, Angular

RESTful
JSON/XML

Database (MySQL, MongoDB, Oracle)

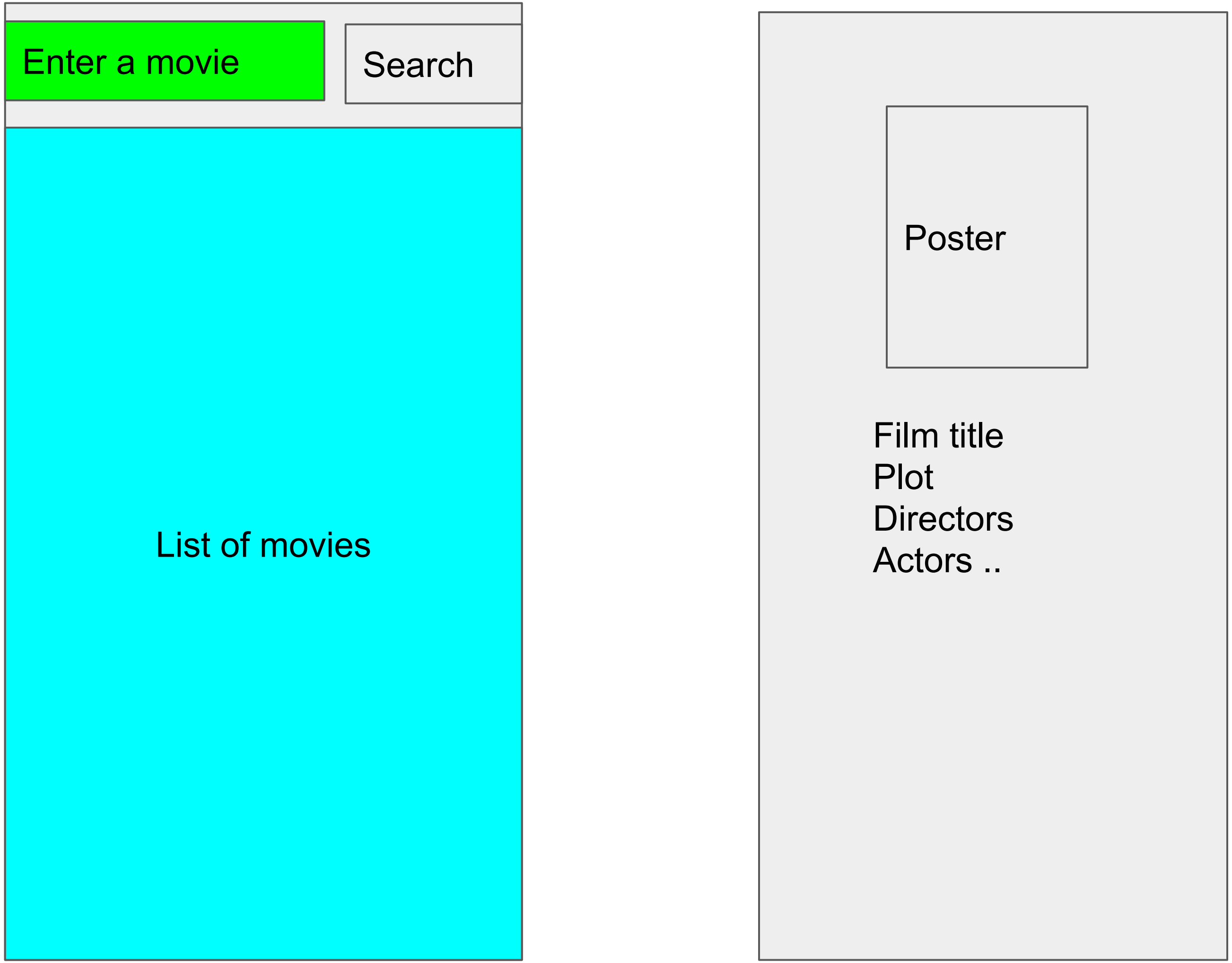
Application Programming Interface (Middleware)

Android (Sqlite, Room Architecture)

iOS

http://www.omdbapi.com/?s=Harry&apikey=87d10179

http://www.omdbapi.com/?i=tt1201607&apikey=87d10179



Enter a movie

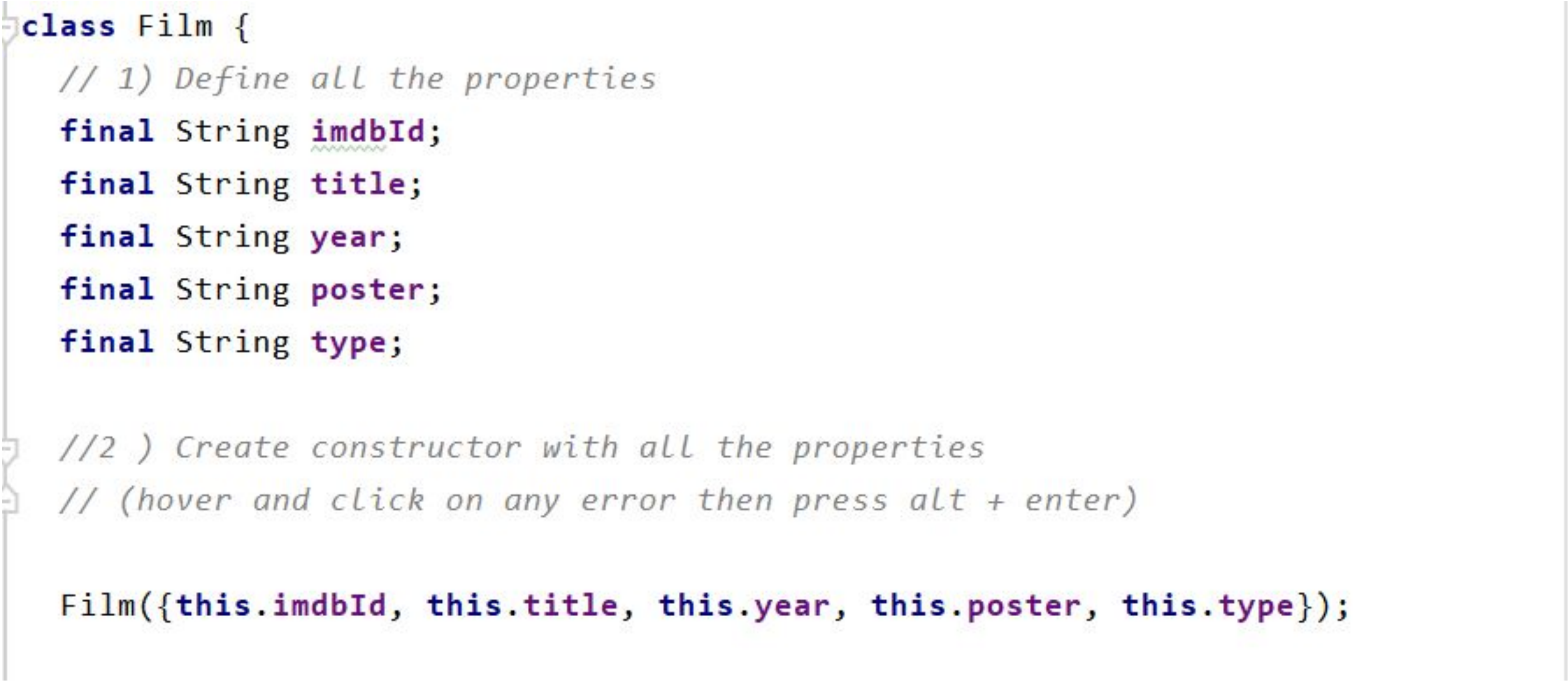
Search

List of movies

Poster

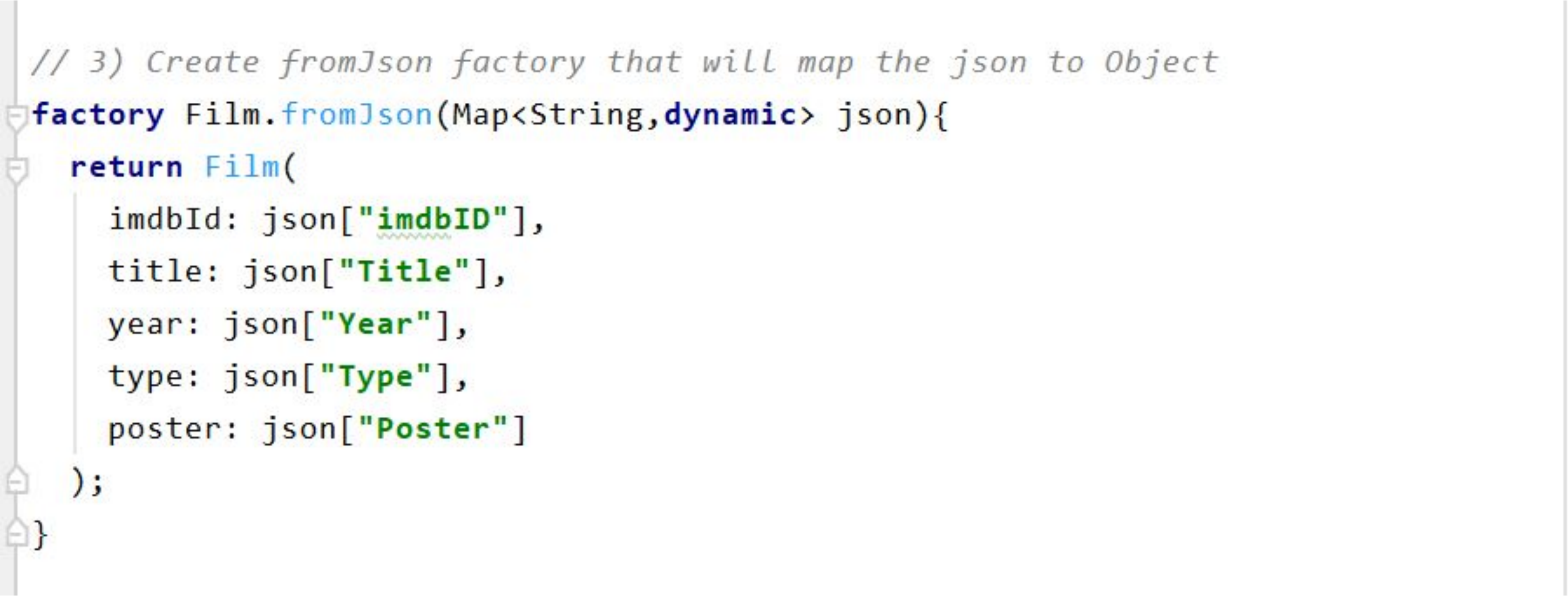
Film title
Plot
Directors
Actors ..

Create Model class (representing JSON)



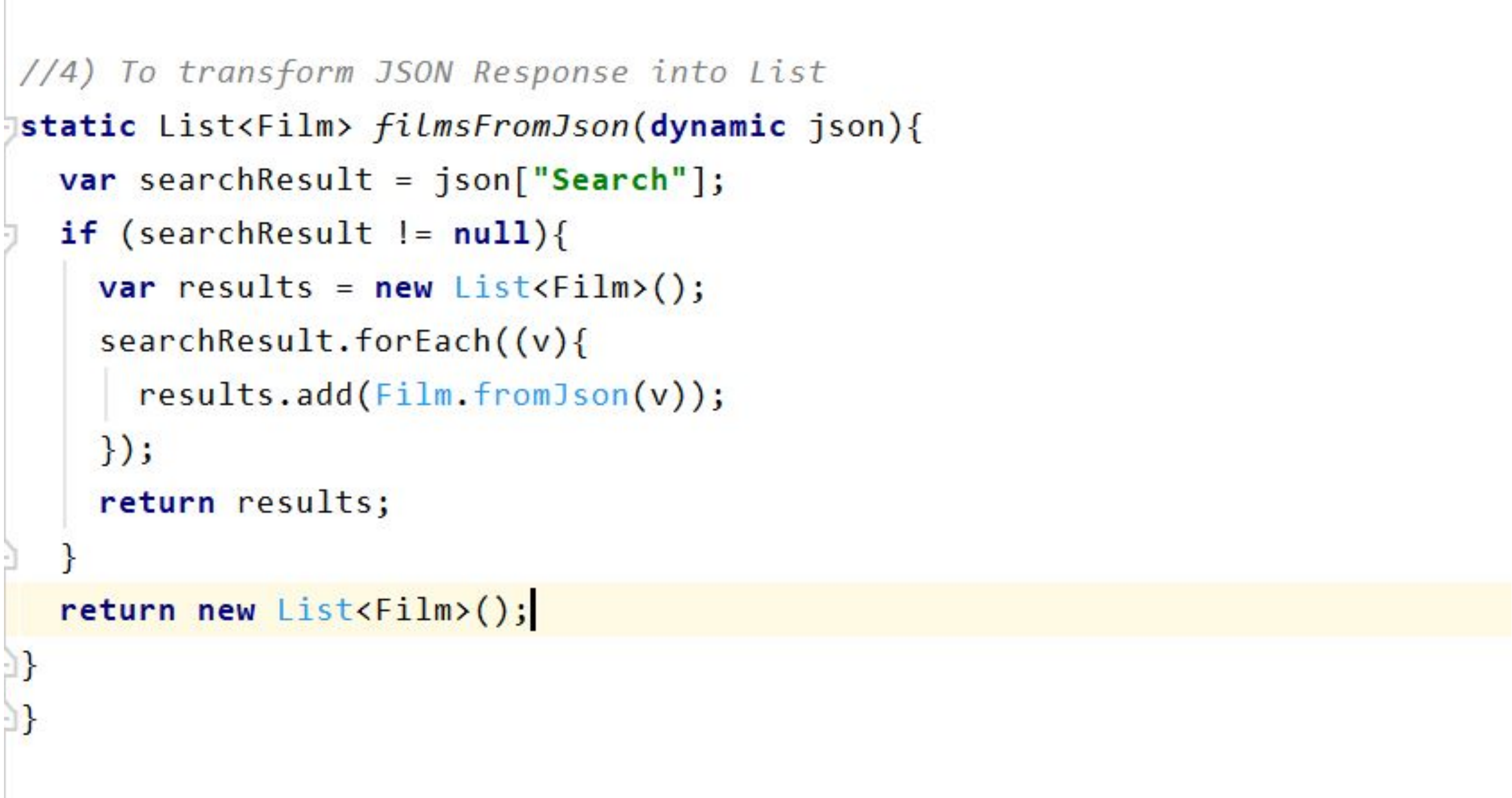
```
class Film {
  // 1) Define all the properties
  final String imdbId;
  final String title;
  final String year;
  final String poster;
  final String type;

  //2 ) Create constructor with all the properties
  // (hover and click on any error then press alt + enter)

  Film({this.imdbId, this.title, this.year, this.poster, this.type});
```

Create fromJSON function to transform JSON to Model



```
// 3) Create fromJson factory that will map the json to Object
factory Film.fromJson(Map<String,dynamic> json){
  return Film(
    imdbId: json["imdbID"],
    title: json["Title"],
    year: json["Year"],
    type: json["Type"],
    poster: json["Poster"]
  );
}
```

Create List<Array> transformation methods



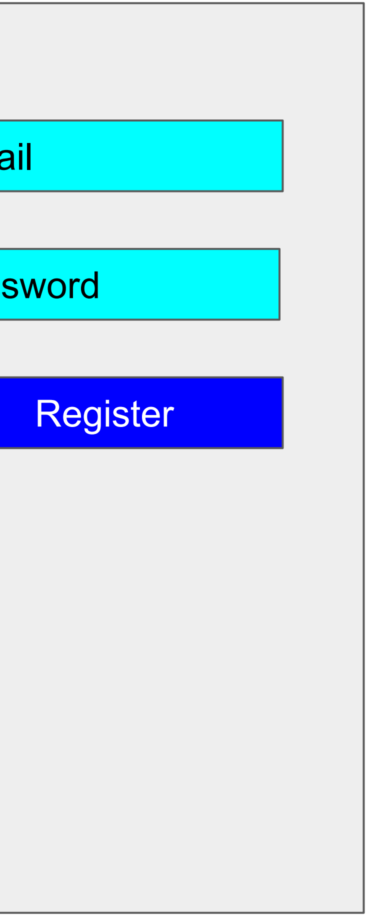
```
//4) To transform JSON Response into List
static List<Film> filmsFromJson(dynamic json){
  var searchResult = json["Search"];
  if (searchResult != null){
    var results = new List<Film>();
    searchResult.forEach((v){
      results.add(Film.fromJson(v));
    });
    return results;
  }
  return new List<Film>();
}
}
```

FetchFilms function

```
Future<List<Film>> fetchFilms() async {

 final response = await http.get('https://www.omdbapi.com/?s=Inception&apikey=87d10179');

 if (response.statusCode == 200) {

   return Film.filmsFromJson(json.decode(response.body));

 } else {

   // If the server did not return a 200 OK response,

   // then throw an exception.

   throw Exception('Failed to load album');

 }

}
```

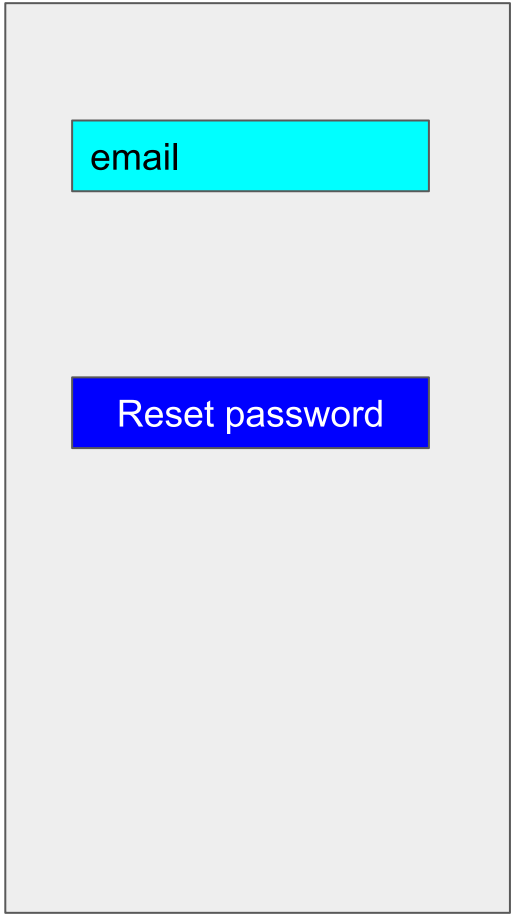
	Python	JS
Computer Science	x	x
Web	X (Flask, Django)	X (React JS, VueJS
App / Mobile		X (React Native)
Data Science	x	
IOT	X (Rasperi Pi/ Arduino)	
DB		X (Mongo DB)

https://insights.stackoverflow.com/survey/2020

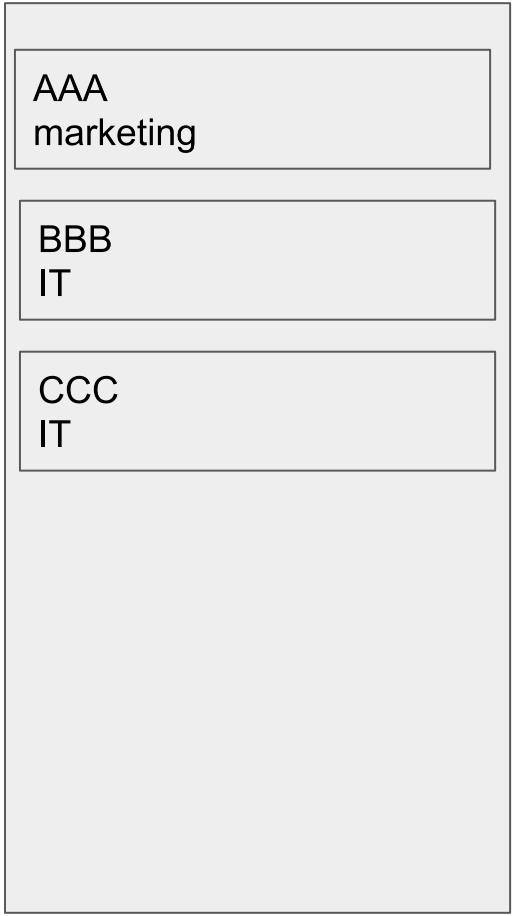

ail

sword

Register

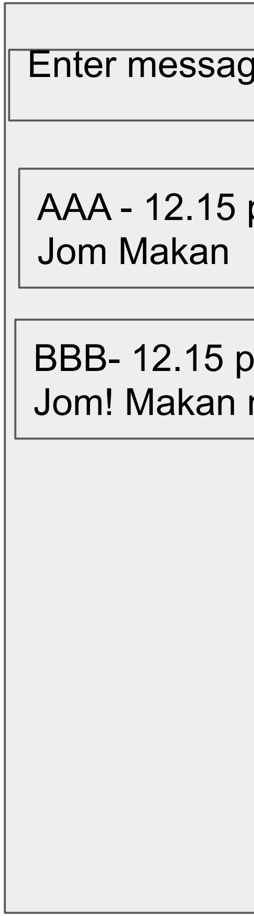

email

Reset password


AAA
marketing

BBB
IT

CCC
IT

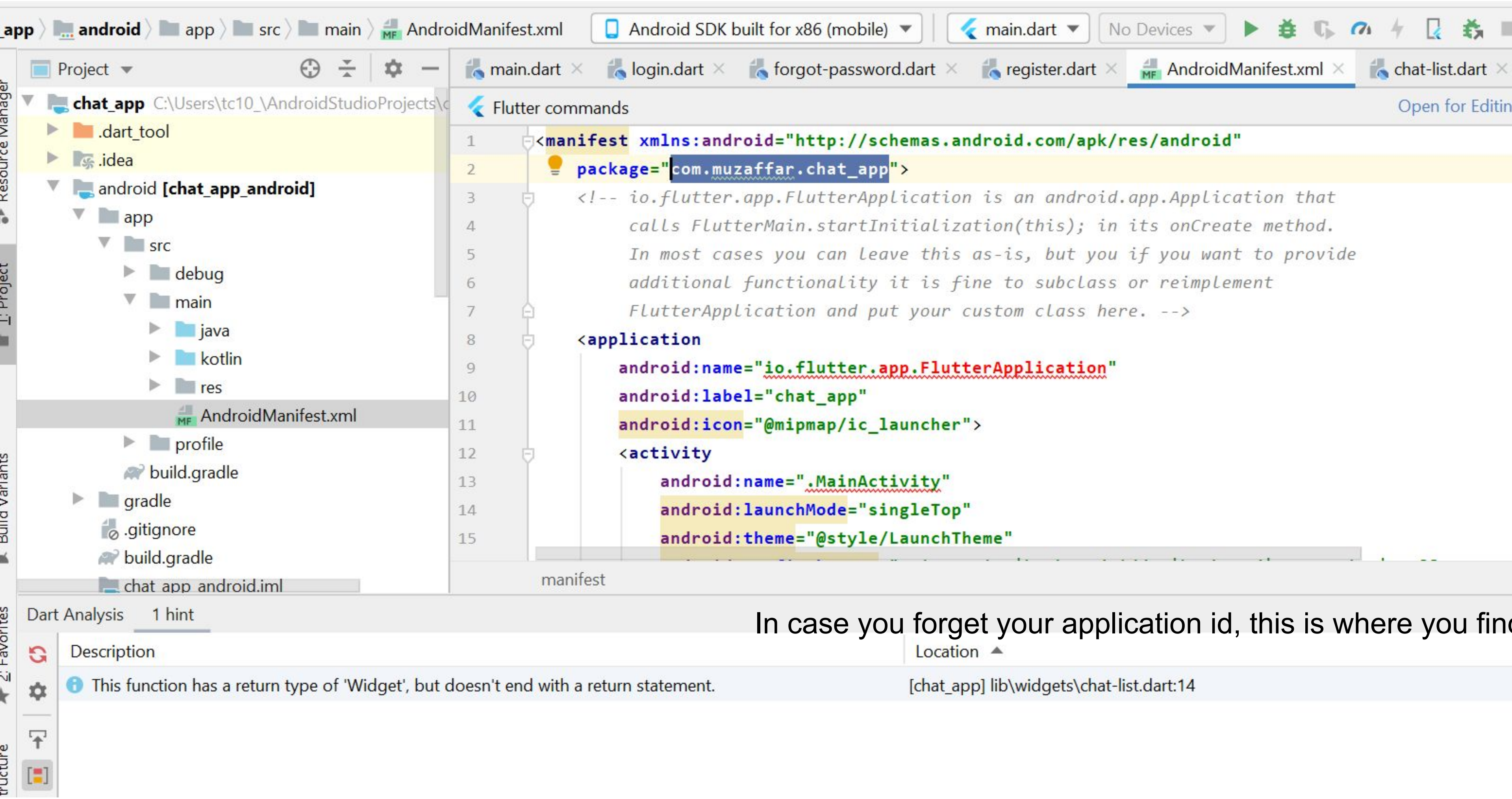

Enter messag

AAA - 12.15 p
Jom Makan

BBB- 12.15 p
Jom! Makan r

Adding firebase in project

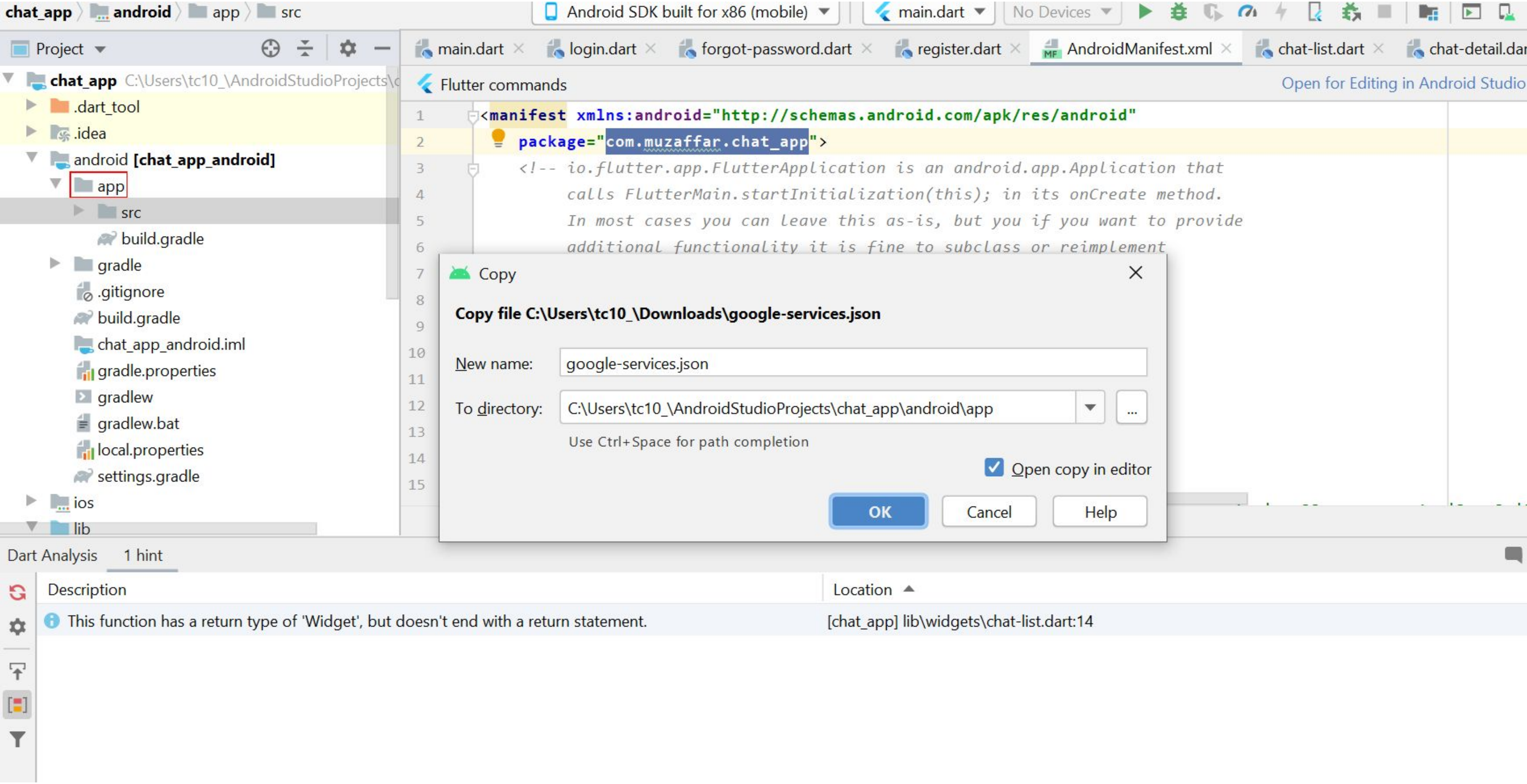
1) Go to firebase.google.com, add a project there, press add (on or off google analytics)
2) Add the google-services.json file into the app root folder
3) Follow the instruction



```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.muzaffar.chat_app">
    <!-- io.flutter.app.FlutterApplication is an android.app.Application that
         calls FlutterMain.startInitialization(this); in its onCreate method.
         In most cases you can leave this as-is, but you if you want to provide
         additional functionality it is fine to subclass or reimplement
         FlutterApplication and put your custom class here. -->
    <application
        android:name="io.flutter.app.FlutterApplication"
        android:label="chat_app"
        android:icon="@mipmap/ic_launcher">
        <activity
            android:name=".MainActivity"
            android:launchMode="singleTop"
            android:theme="@style/LaunchTheme"
```

Description	Location ▲
This function has a return type of 'Widget', but doesn't end with a return statement.	[chat_app] lib\widgets\chat-list.dart:14

In case you forget your application id, this is where you fin



```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.muzaffar.chat_app">
    <!-- io.flutter.app.FlutterApplication is an android.app.Application that
         calls FlutterMain.startInitialization(this); in its onCreate method.
         In most cases you can leave this as-is, but you if you want to provide
         additional functionality it is fine to subclass or reimplement
```

Copy

Copy file C:\Users\tc10_\Downloads\google-services.json

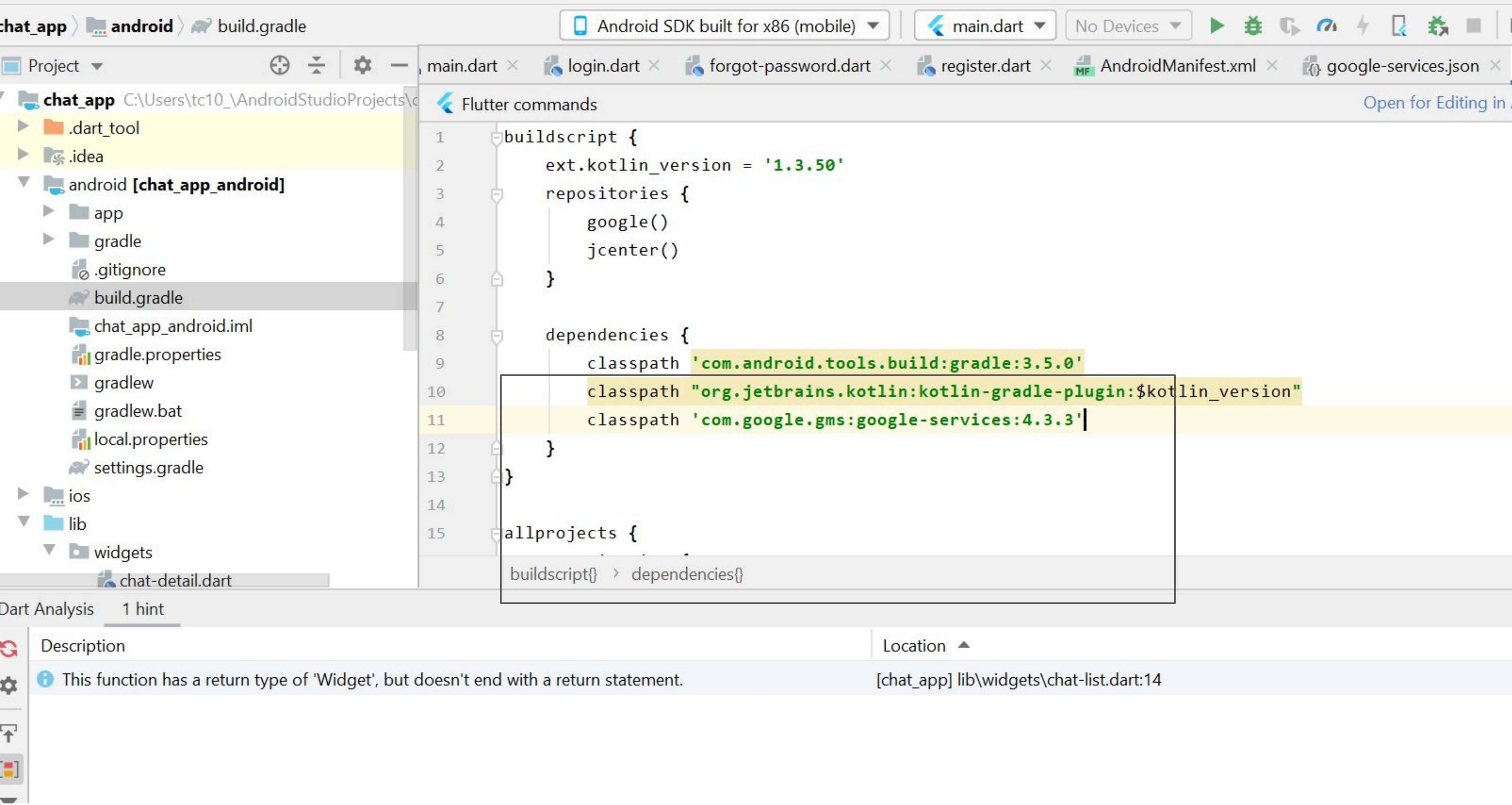
New name: google-services.json

To directory: C:\Users\tc10_\AndroidStudioProjects\chat_app\android\app

Use Ctrl+Space for path completion

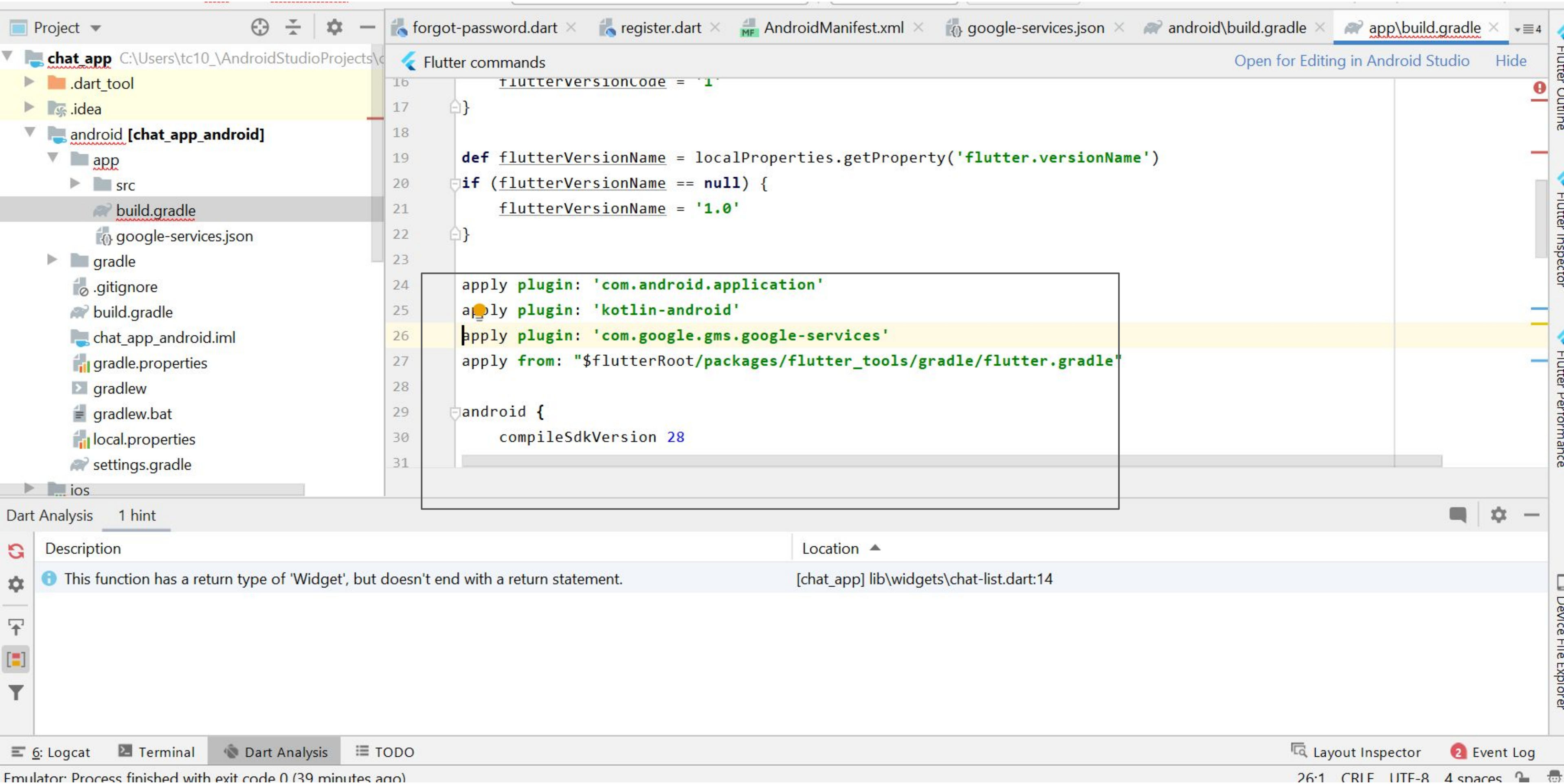
☑ Open copy in editor

This function has a return type of 'Widget', but doesn't end with a return statement. [chat_app] lib\widgets\chat-list.dart:14



```
buildscript {
    ext.kotlin_version = '1.3.50'
    repositories {
        google()
        jcenter()
    }

    dependencies {
        classpath 'com.android.tools.build:gradle:3.5.0'
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"
        classpath 'com.google.gms:google-services:4.3.3'
    }
}

allprojects {
```

This function has a return type of 'Widget', but doesn't end with a return statement. [chat_app] lib\widgets\chat-list.dart:14



```
    flutterVersionCode = '1'
}

def flutterVersionName = localProperties.getProperty('flutter.versionName')
if (flutterVersionName == null) {
    flutterVersionName = '1.0'
}

apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'
apply plugin: 'com.google.gms.google-services'
apply from: "$flutterRoot/packages/flutter_tools/gradle/flutter.gradle"

android {
    compileSdkVersion 28
```

Dart Analysis 1 hint

Description	Location
This function has a return type of 'Widget', but doesn't end with a return statement.	[chat_app] lib\widgets\chat-list.dart:14

```
Emulator: Process finished with exit code 0 (39 minutes ago)          26:1  CRLF  UTF-8  4 spaces
```

Pub dev

1) **Install Firebase Authentication, cloud firestore and Firebase Core**

Initalize Firebase app (add this in main file)

```
// Add the import

import 'package:firebase_core/firebase_core.dart';



void main() async {

// Add these two lines

 WidgetsFlutterBinding.ensureInitialized();

 await Firebase.initializeApp();

 runApp(MyApp());

}
```

Firebase Authentication

For all the page that is going to use Firebase Authentication , : login, register, forget password, you will add the 1st line everytime

1) Initialize Firebase Authentication Instance

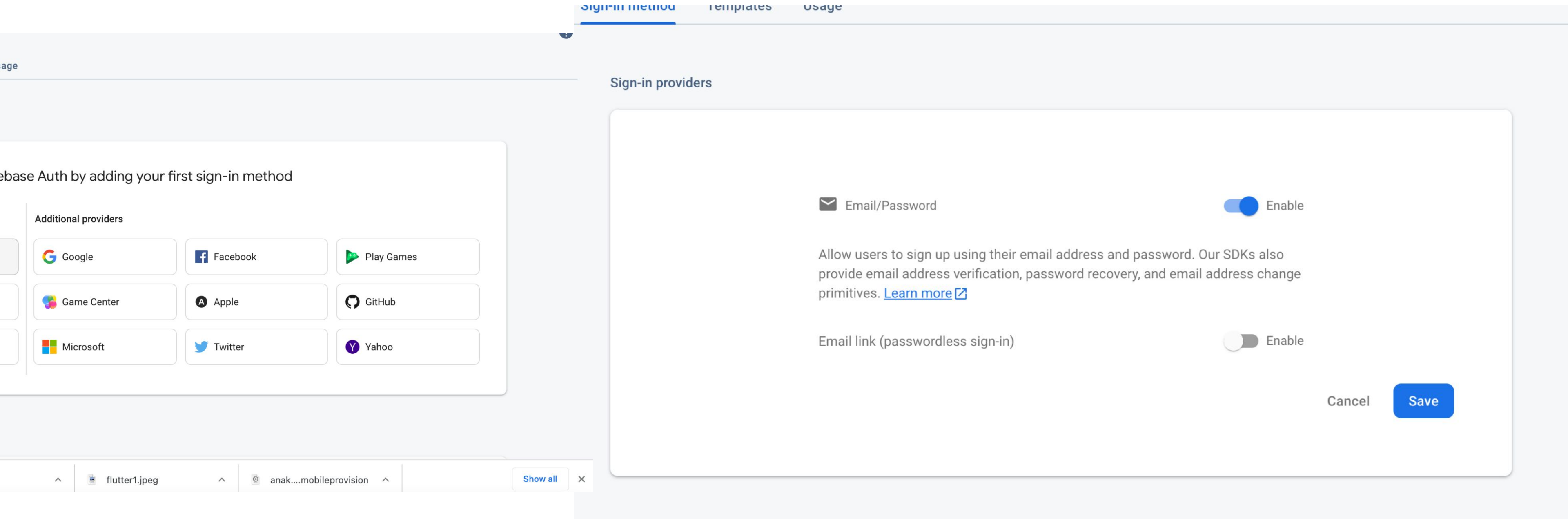
```
var _auth = FirebaseAuth.instance;
```

2) Add code to createUser (as next page)

Code to createUser

```
User user =

   (await _auth.createUserWithEmailAndPassword(email: email,

       password: password))

       .user;

if (user != null){

 print("Succesfully logged in");

}

else {

 print("Something is wrong!");

}
```

Before testing you need to go to firebase authentication console, and enable firebase authentication



ebase Auth by adding your first sign-in method

Additional providers

Sign-in providers

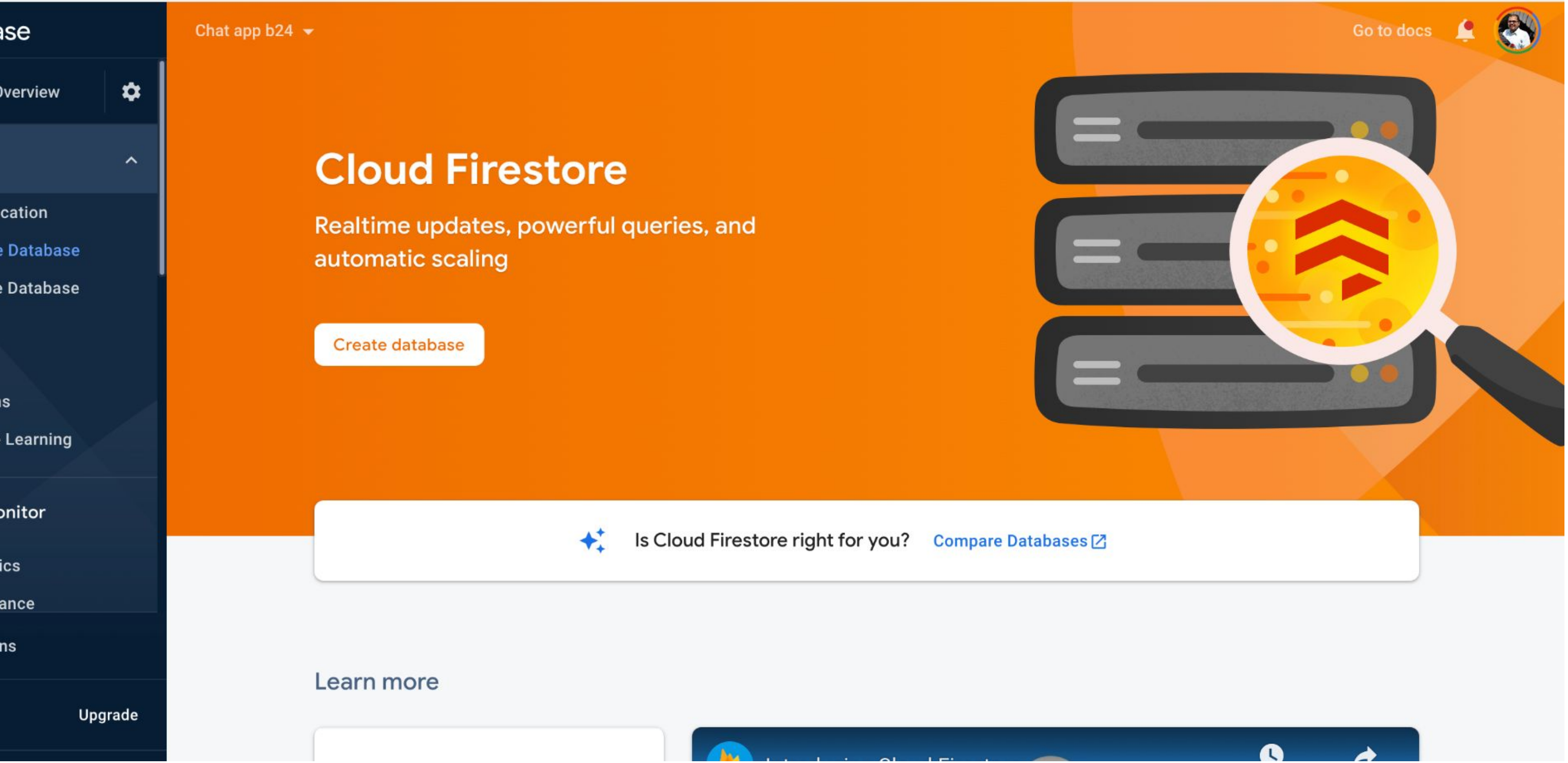
Email/Password ● Enable

Allow users to sign up using their email address and password. Our SDKs also provide email address verification, password recovery, and email address change primitives. Learn more

Email link (passwordless sign-in) ○ Enable

Sign in code

```
User user = (await _auth.signInWithEmailAndPassword(
       email: emailEditingController.text,
       password: passwordEditingController.text))
   .user;
print(user);
if (user != null) {
 print("Succesfully logged in!");
 Navigator.push(context, MaterialPageRoute(builder: (context)=>ChatList()));
} else {
 print("Error");
}
```



Cloud Firestore

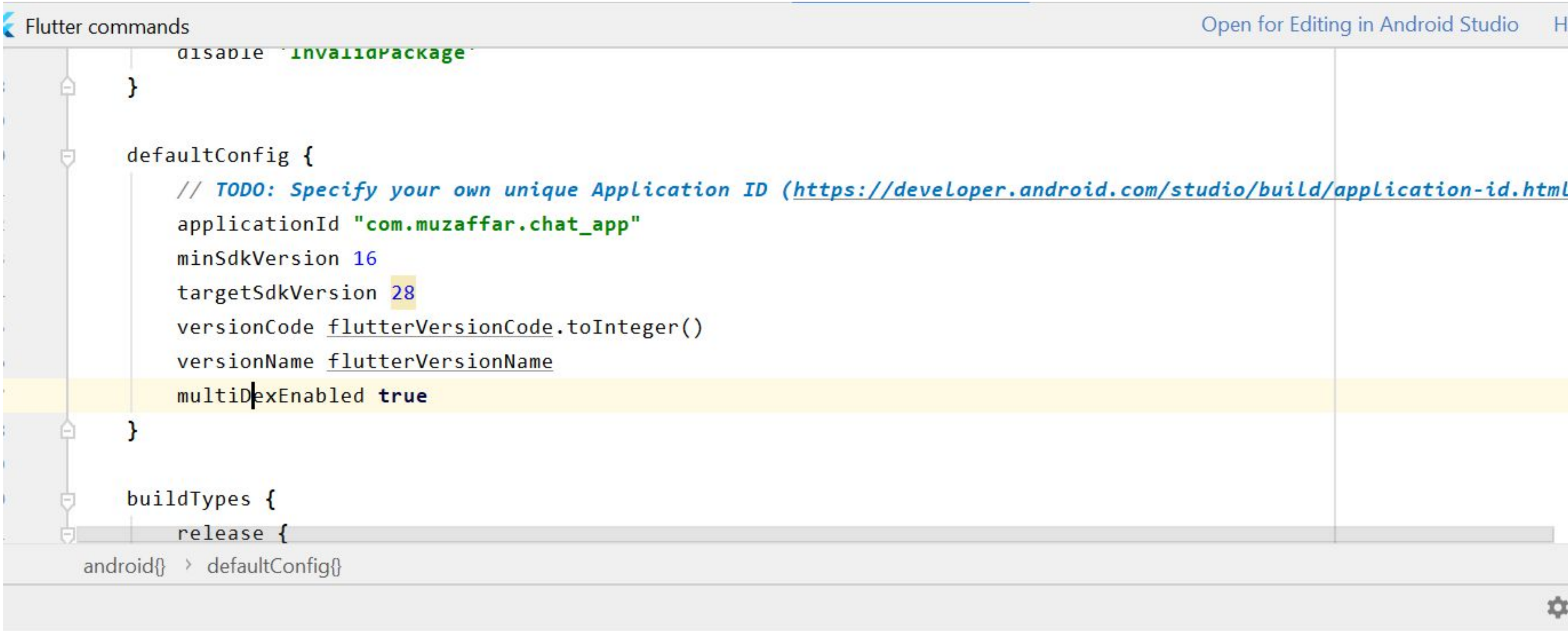
Realtime updates, powerful queries, and automatic scaling

Is Cloud Firestore right for you? Compare Databases

Learn more

Setup firestore

1) Initialize Firestore Instance
2) Add multiDexEnabled inside app level build.gradle (refer next page)
3) Call the methods :
   a) Add/ Create = setData


Flutter commands
Open for Editing in Android Studio
H
android{} › defaultConfig{}

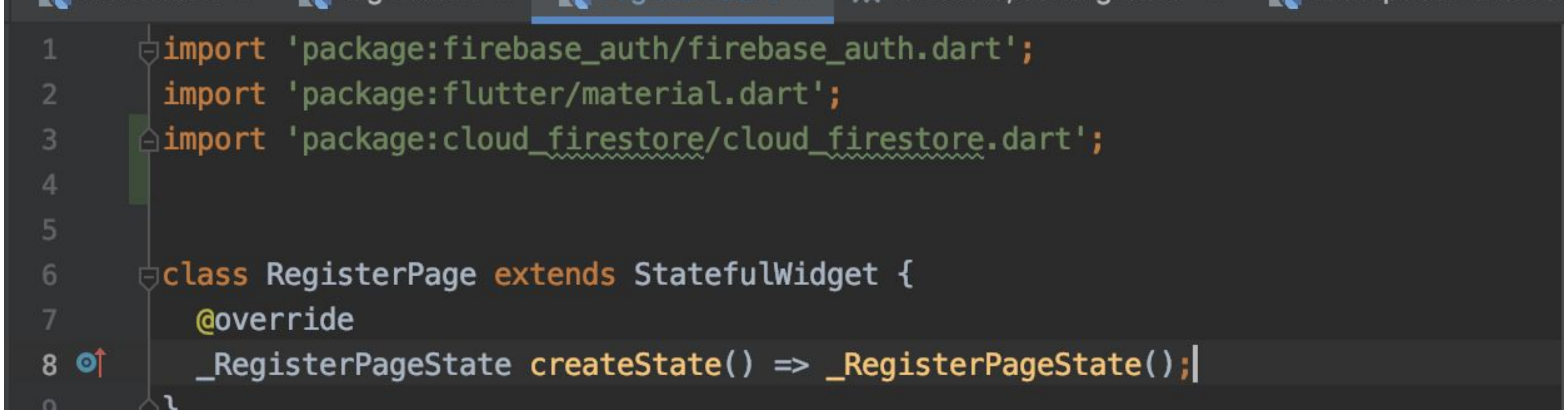
```
        disable 'InvalidPackage'
    }

    defaultConfig {
        // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html
        applicationId "com.muzaffar.chat_app"
        minSdkVersion 16
        targetSdkVersion 28
        versionCode flutterVersionCode.toInteger()
        versionName flutterVersionName
        multiDexEnabled true
    }

    buildTypes {
        release {
```

Firestore permission

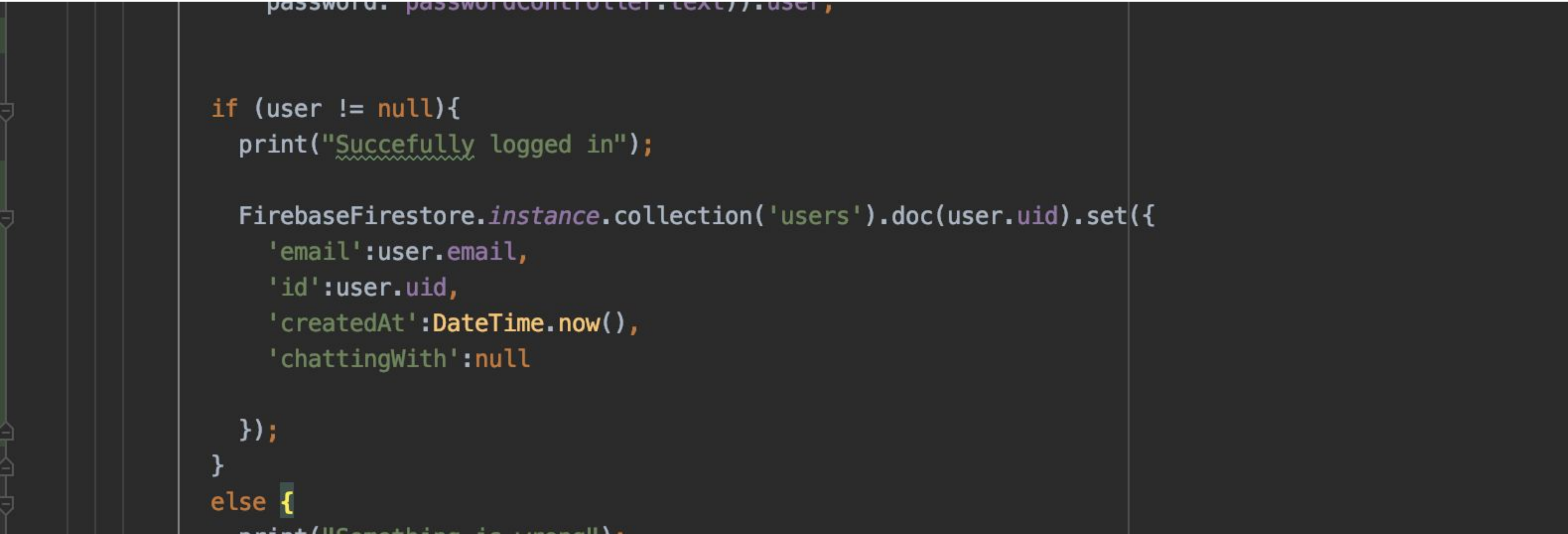
```
rules_version = '2';

service cloud.firestore {

  match /databases/{database}/documents {

    match /{document=**} {

      allow read, write: if

          request.time < timestamp.date(2020, 10, 3);

    }

  }
```

Example - Creating a users collection after user succesfully registered

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

..

var user = value.user!;

FirebaseFirestore.instance.collection('users').doc(user.uid).set({

 'email':user.email,

 'id':user.uid,

 'createdAt':DateTime.now(),

 'chattingWith':null

});
```

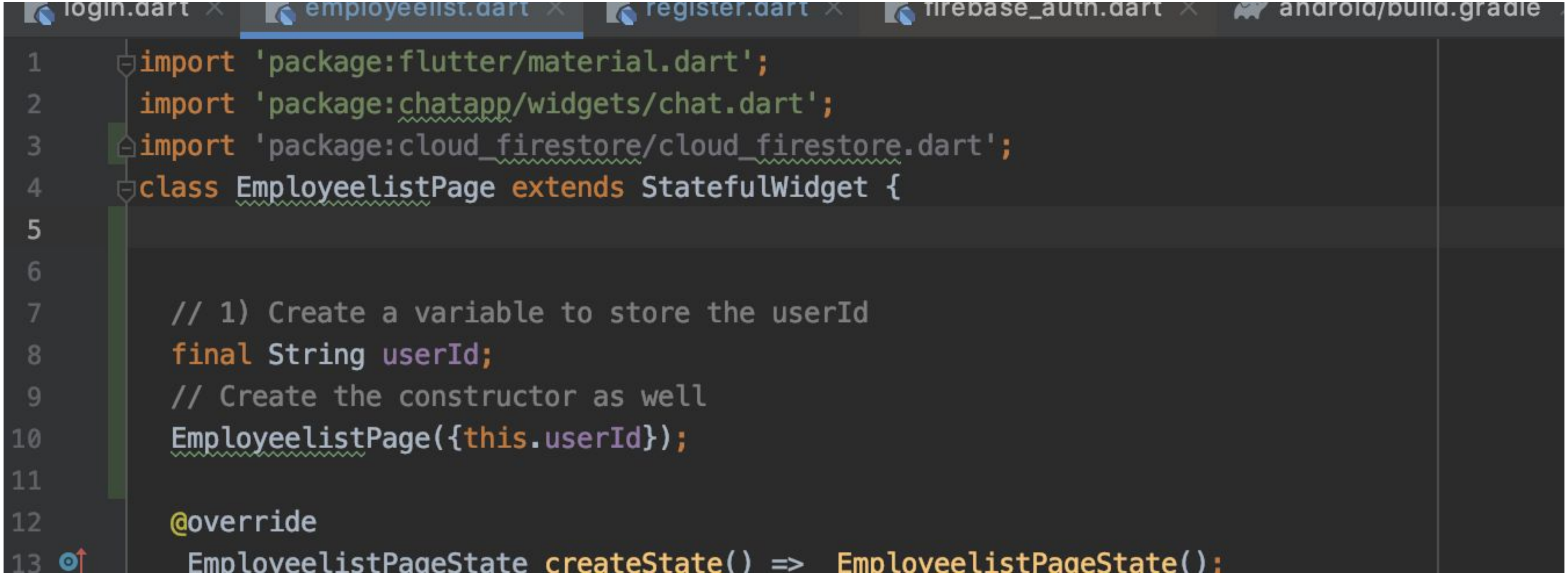


```dart
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';


class RegisterPage extends StatefulWidget {
  @override
  _RegisterPageState createState() => _RegisterPageState();
}
```



```dart
          password: passwordController.text)).user;


      if (user != null){
        print("Succefully logged in");

        FirebaseFirestore.instance.collection('users').doc(user.uid).set({
          'email':user.email,
          'id':user.uid,
          'createdAt':DateTime.now(),
          'chattingWith':null

        });
      }
      else {
        print("Something is wrong");
```

1) Create the userId and constructor, I need to know who is currently logging in



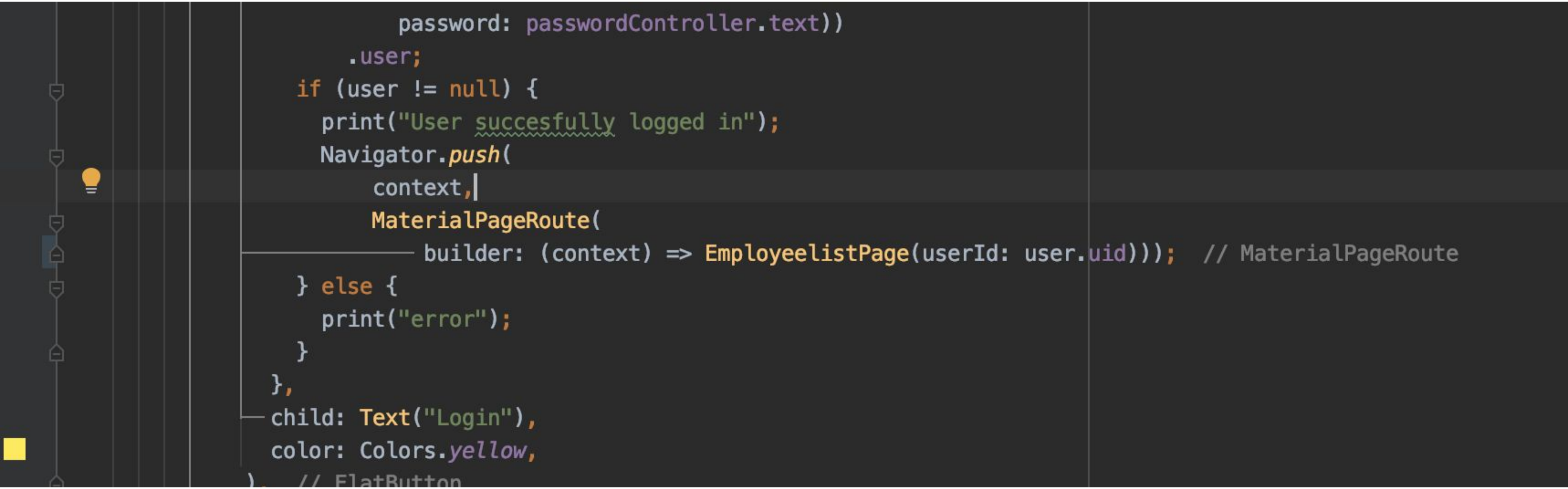
```
import 'package:flutter/material.dart';
import 'package:chatapp/widgets/chat.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
class EmployeelistPage extends StatefulWidget {


  // 1) Create a variable to store the userId
  final String userId;
  // Create the constructor as well
  EmployeelistPage({this.userId});

  @override
  _EmployeelistPageState createState() => _EmployeelistPageState();
```

2) Get the data and show it in the ListView

https://dev.to/kazuhideoki/how-to-get-data-from-firestore-and-show-it-on-flutterbuilder-or-streambuilder-e05

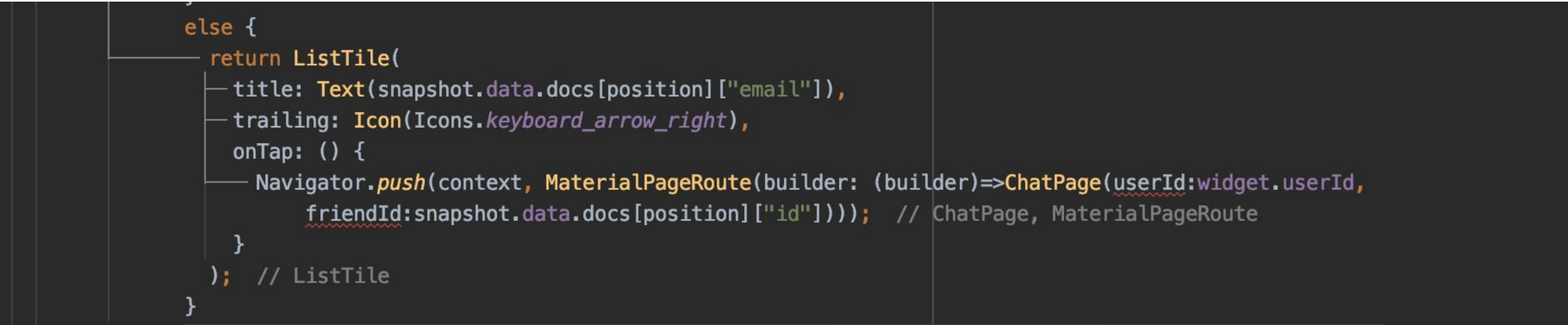
```
StreamBuilder<QuerySnapshot>(
   stream:
FirebaseFirestore.instance.collection('users').
snapshots(),
   builder: (context, snapshot) {
     if (!snapshot.hasData) {
       return CircularProgressIndicator();
     } else {
       final List<DocumentSnapshot> documents =
           snapshot.requireData.docs;
       return ListView.builder(
           itemCount: documents.length,
           itemBuilder: (context, position) {
```

3) Pass the userId from Login page to Employee List

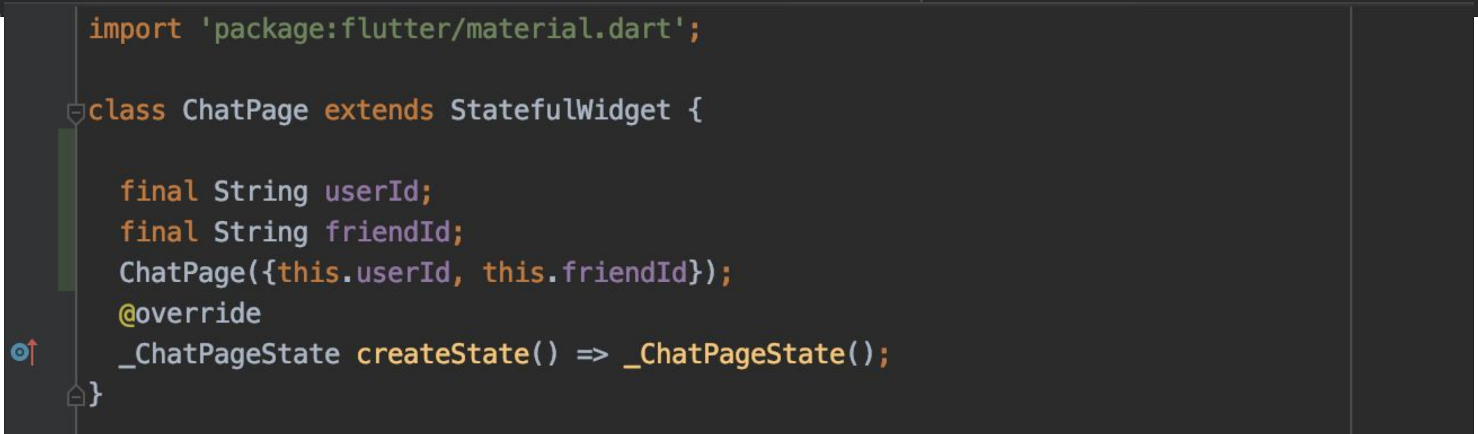


```
            password: passwordController.text))
        .user;
    if (user != null) {
      print("User succesfully logged in");
      Navigator.push(
          context,
          MaterialPageRoute(
              builder: (context) => EmployeelistPage(userId: user.uid)));  // MaterialPageRoute
    } else {
      print("error");
    }
  },
  child: Text("Login"),
  color: Colors.yellow,
),  // FlatButton
```

4) Create an onTap that will open the new page

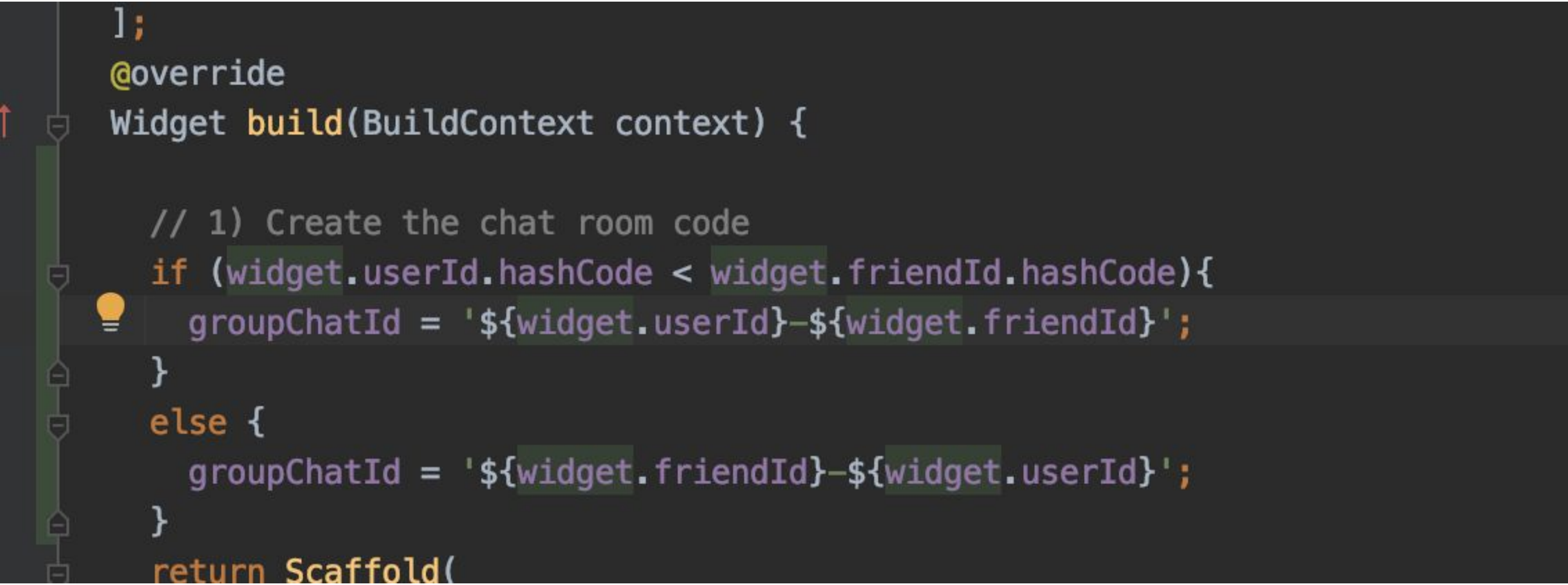


```
else {
  return ListTile(
    title: Text(snapshot.data.docs[position]["email"]),
    trailing: Icon(Icons.keyboard_arrow_right),
    onTap: () {
      Navigator.push(context, MaterialPageRoute(builder: (builder)=>ChatPage(userId:widget.userId,
          friendId:snapshot.data.docs[position]["id"])));  // ChatPage, MaterialPageRoute
    }
  );  // ListTile
}
```



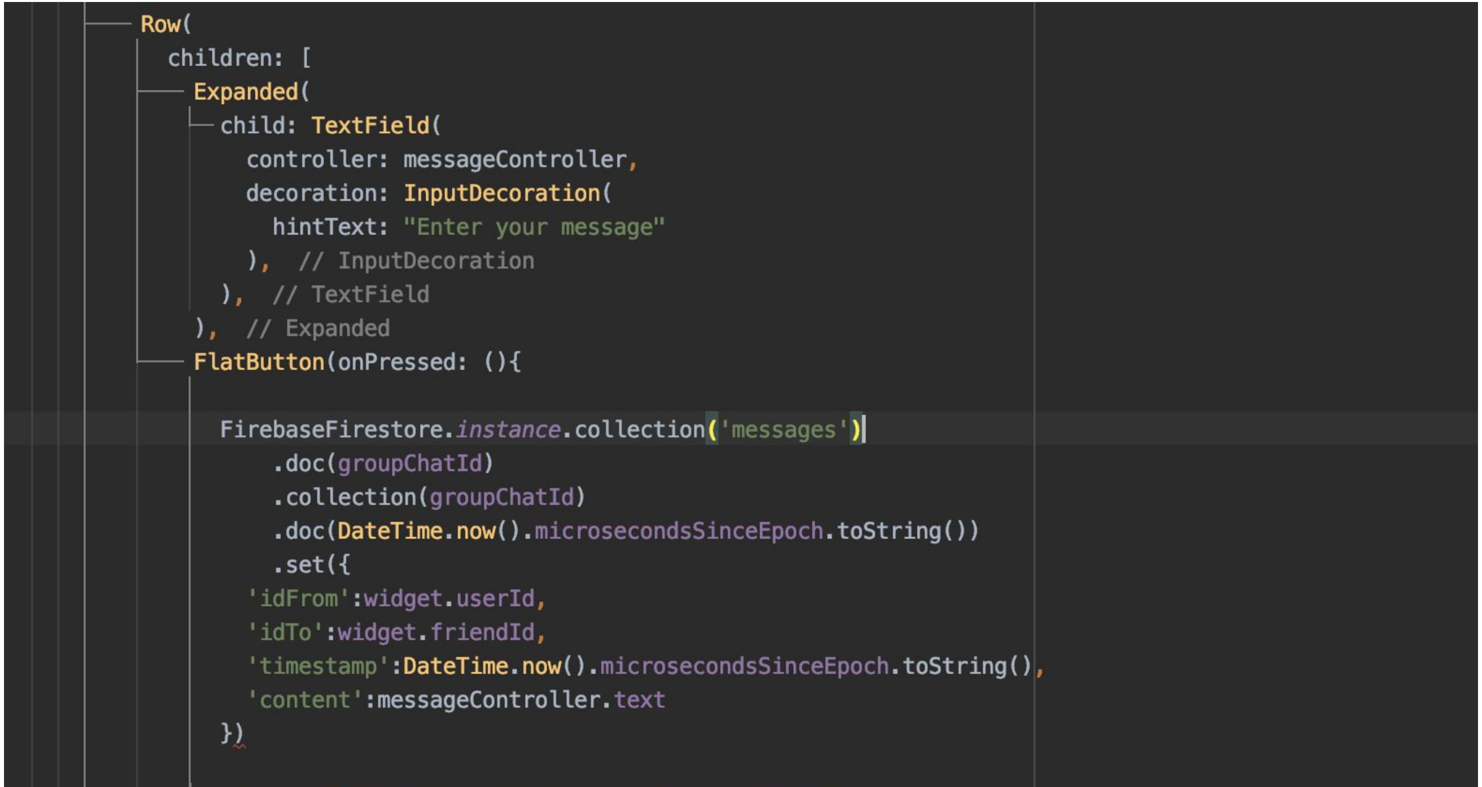
```
import 'package:flutter/material.dart';

class ChatPage extends StatefulWidget {

  final String userId;
  final String friendId;
  ChatPage({this.userId, this.friendId});
  @override
  _ChatPageState createState() => _ChatPageState();
}
```

1) Create the channel ID

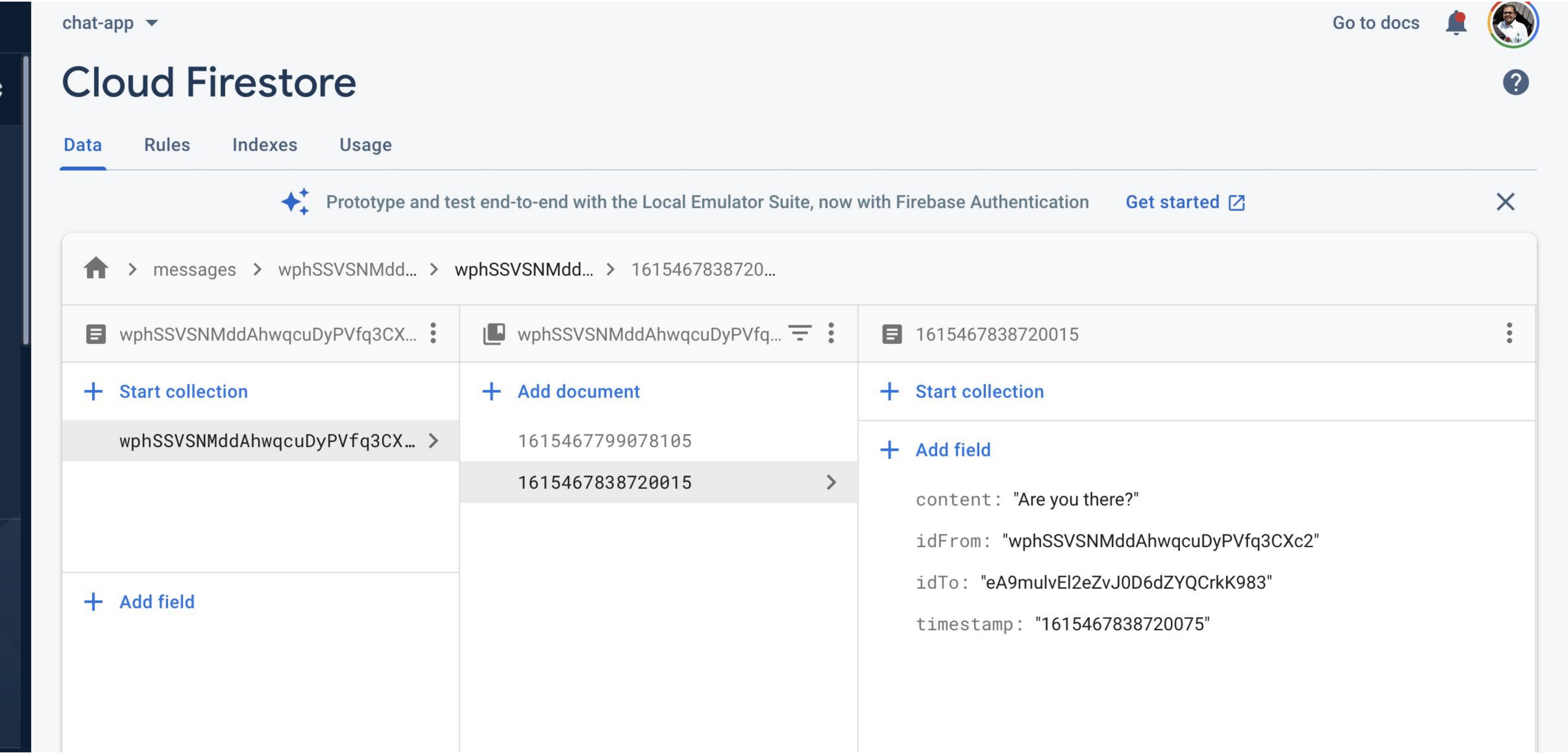


```
];
@override
Widget build(BuildContext context) {

  // 1) Create the chat room code
  if (widget.userId.hashCode < widget.friendId.hashCode){
    groupChatId = '${widget.userId}-${widget.friendId}';
  }
  else {
    groupChatId = '${widget.friendId}-${widget.userId}';
  }
  return Scaffold(
```

2) Code to chat (save/pass data to firebase)



```
Row(
  children: [
    Expanded(
      child: TextField(
        controller: messageController,
        decoration: InputDecoration(
          hintText: "Enter your message"
        ),  // InputDecoration
      ),  // TextField
    ),  // Expanded
    FlatButton(onPressed: (){

      FirebaseFirestore.instance.collection('messages')
          .doc(groupChatId)
          .collection(groupChatId)
          .doc(DateTime.now().microsecondsSinceEpoch.toString())
          .set({
        'idFrom':widget.userId,
        'idTo':widget.friendId,
        'timestamp':DateTime.now().microsecondsSinceEpoch.toString(),
        'content':messageController.text
      })
```



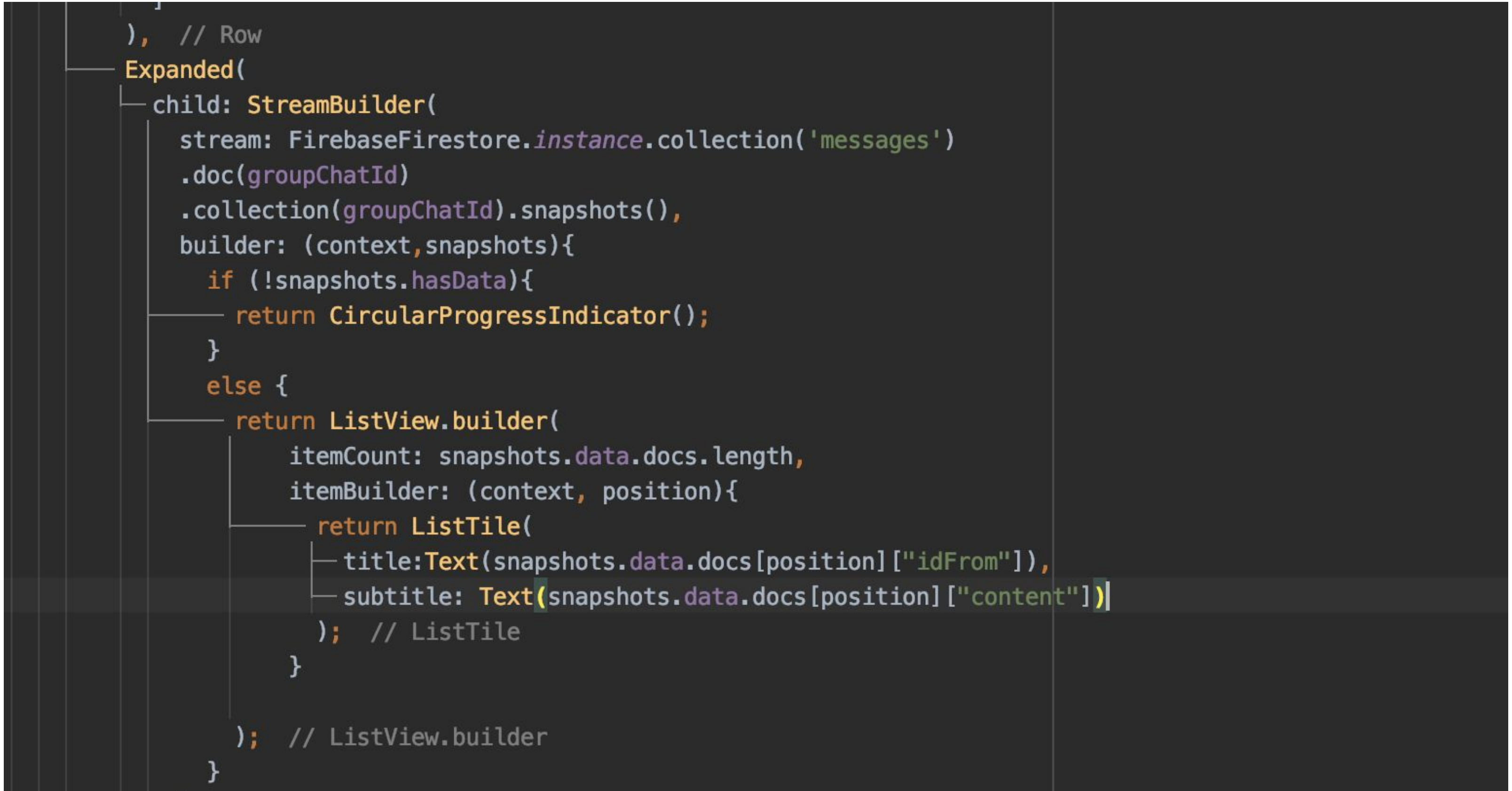
Cloud Firestore

Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication

messages > wphSSVSNMdd... > wphSSVSNMdd... > 1615467838720...

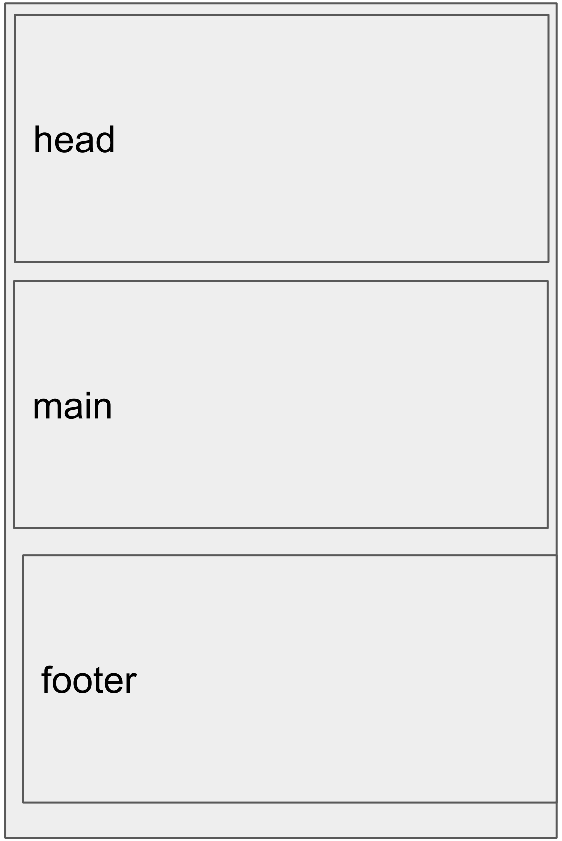
wphSSVSNMddAhwqcuDyPVfq3CX...	wphSSVSNMddAhwqcuDyPVfq...	1615467838720015
wphSSVSNMddAhwqcuDyPVfq3CX...	1615467799078105 1615467838720015	content: "Are you there?" idFrom: "wphSSVSNMddAhwqcuDyPVfq3CXc2" idTo: "eA9mulvEl2eZvJ0D6dZYQCrkK983" timestamp: "1615467838720075"

5) Retrieve the chat, and show the message



```
 ]
),  // Row
Expanded(
 child: StreamBuilder(
  stream: FirebaseFirestore.instance.collection('messages')
  .doc(groupChatId)
  .collection(groupChatId).snapshots(),
  builder: (context,snapshots){
   if (!snapshots.hasData){
    return CircularProgressIndicator();
   }
   else {
    return ListView.builder(
      itemCount: snapshots.data.docs.length,
      itemBuilder: (context, position){
       return ListTile(
        title:Text(snapshots.data.docs[position]["idFrom"]),
        subtitle: Text(snapshots.data.docs[position]["content"])
       );  // ListTile
      }

    );  // ListView.builder
   }
```

/main

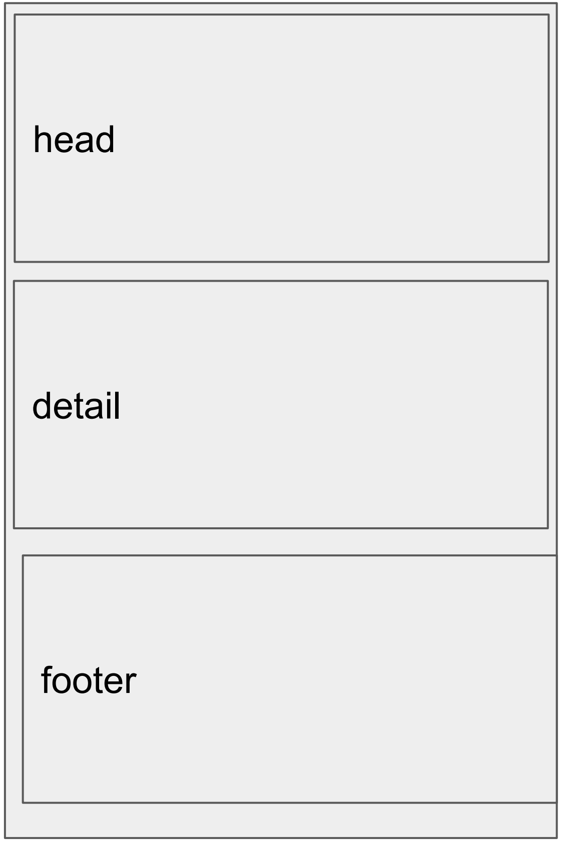


head

main

footer

/detail



head

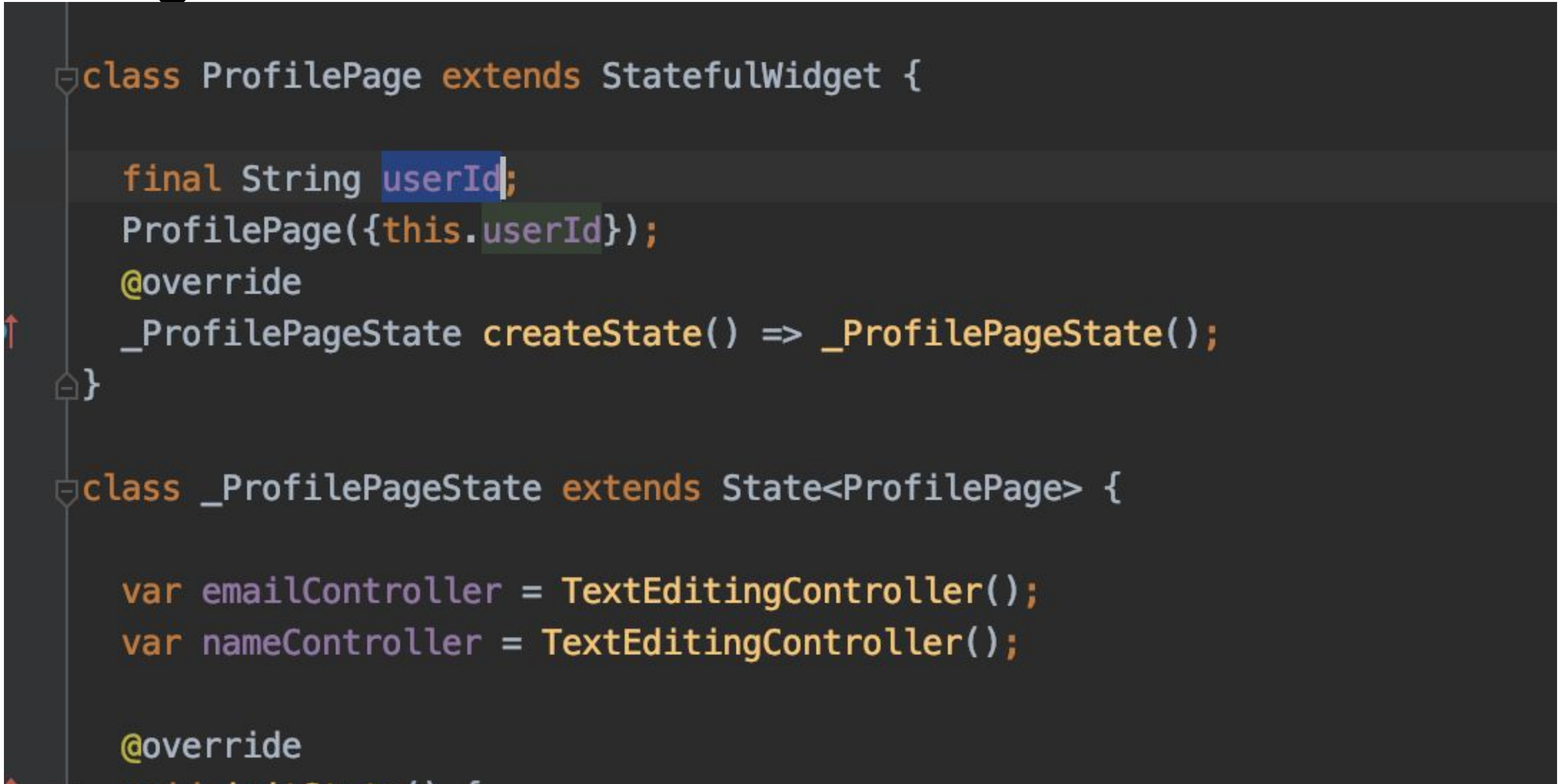
detail

footer

6) Modify your code to show the sender email ...

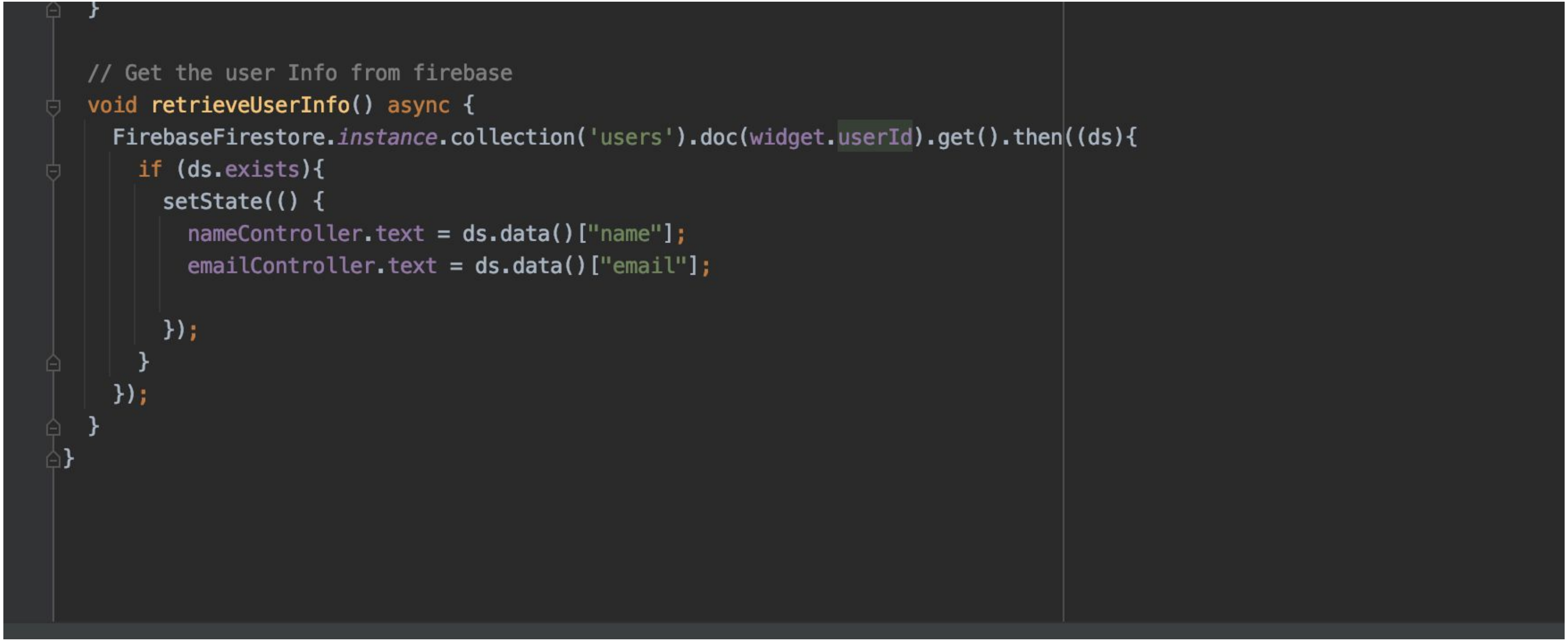
Later verify with my source code

Inside profile, you need current UserId, create the textediting controller as well



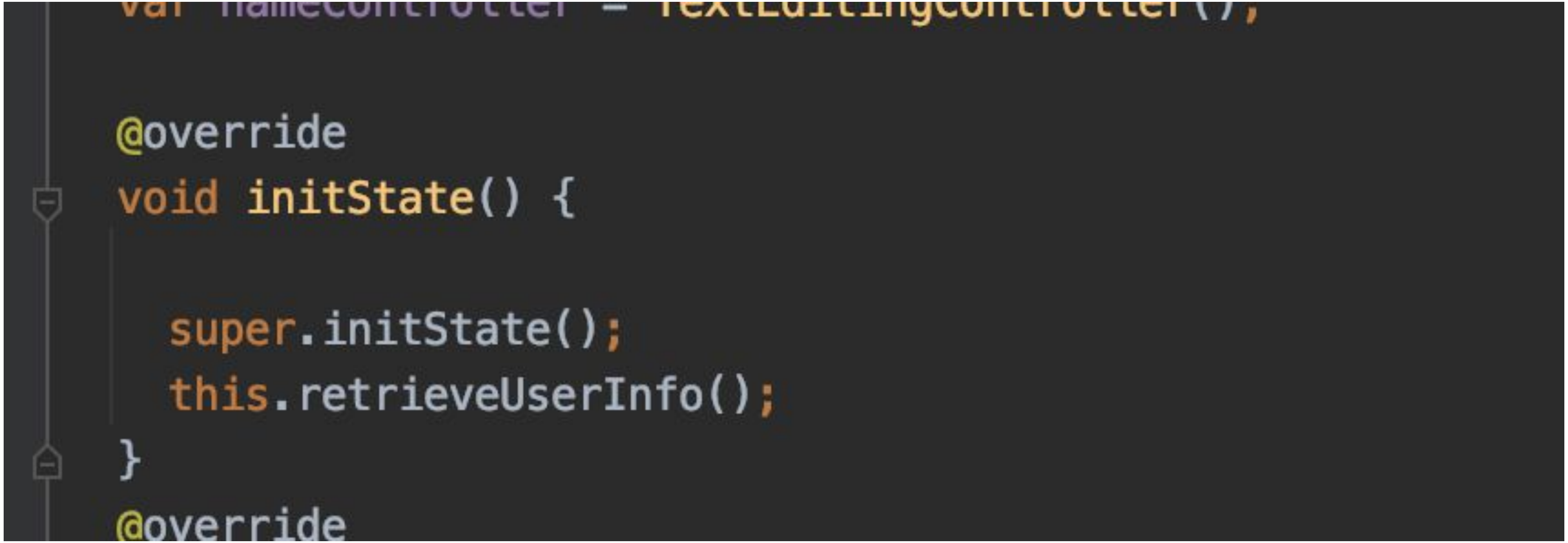
```
class ProfilePage extends StatefulWidget {

  final String userId;
  ProfilePage({this.userId});
  @override
  _ProfilePageState createState() => _ProfilePageState();
}

class _ProfilePageState extends State<ProfilePage> {

  var emailController = TextEditingController();
  var nameController = TextEditingController();

  @override
```

This is the code to get the data and assign it to the controller

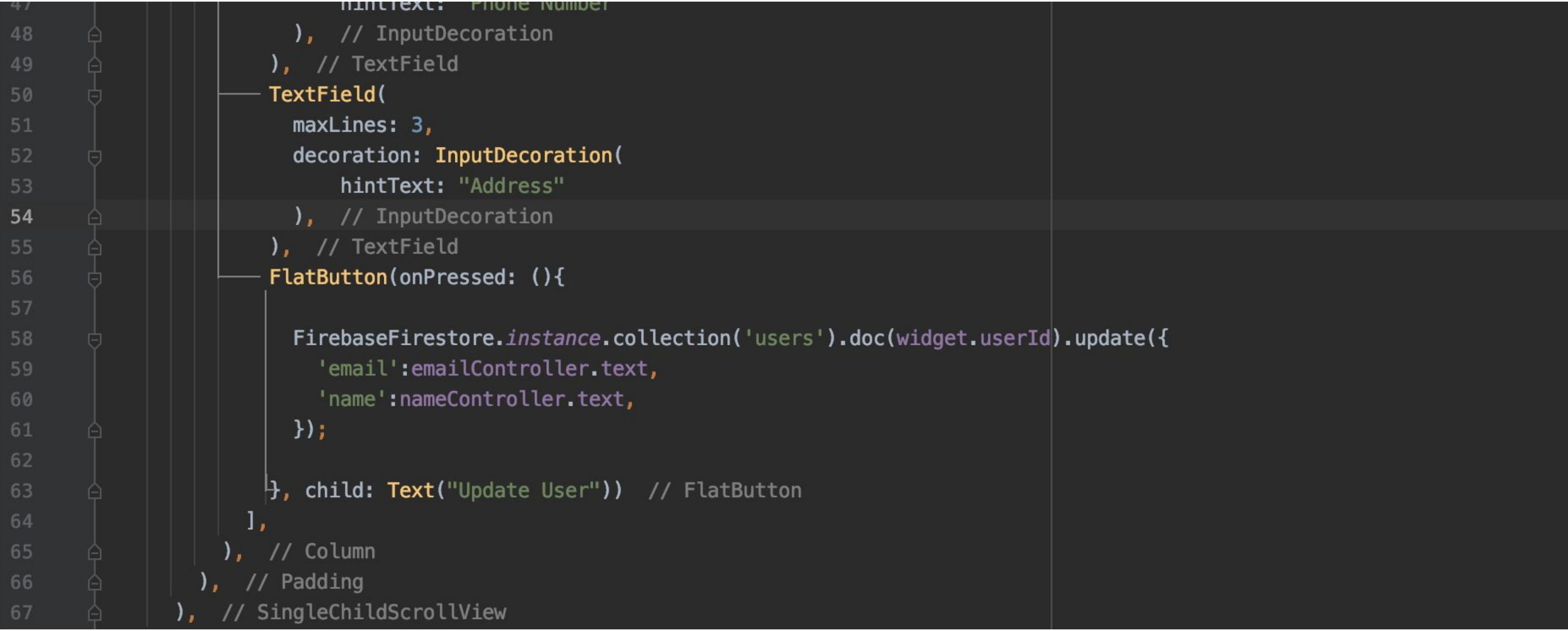


```
  }

  // Get the user Info from firebase
  void retrieveUserInfo() async {
    FirebaseFirestore.instance.collection('users').doc(widget.userId).get().then((ds){
      if (ds.exists){
        setState(() {
          nameController.text = ds.data()["name"];
          emailController.text = ds.data()["email"];

        });
      }
    });
  }
}
```

Call the method created on initState



```
var nameController = TextEditingController();

@override
void initState() {

  super.initState();
  this.retrieveUserInfo();
}
@override
```



```
       hintText: "Phone Number"
     ),  // InputDecoration
    ),  // TextField
    TextField(
     maxLines: 3,
     decoration: InputDecoration(
       hintText: "Address"
     ),  // InputDecoration
    ),  // TextField
    FlatButton(onPressed: (){

     FirebaseFirestore.instance.collection('users').doc(widget.userId).update({
      'email':emailController.text,
      'name':nameController.text,
     });

    }, child: Text("Update User"))  // FlatButton
   ],
  ),  // Column
 ),  // Padding
),  // SingleChildScrollView
```

1) Complete the rest code for update profile, include phone number and address profile update (compulsary)
2) Add a View Profile page, for example on the top right of chat pagem you can have a view profile page that will bring you to your friend’s profile page showing his or her info

If you want to do more, use this tutorial, which I simplified to teach you:

https://medium.com/flutter-community/building-a-chat-app-with-flutter-and-firebase-from-scratch-9eaa7f41782e