



**UNIVERSITY OF SCIENCE AND TECHNOLOGY
OF SOUTHERN PHILIPPINES**

Alubijid | Balubal | Cagayan de Oro | Claveria | Jasaan | Oroquieta | **Panaon** | Villanueva

Calculator App of 3 Conversion Feature and Documentation

(USTP PANAON)

Submitted by:

Reynaldo D. Aquit

Mr. Marlon M. Bajolo

Performance Innovative Task: IT 313 – Mobile Programming

January 2022

Abstract

With flutter and Dart, I able to create an App Calculator this have 3 conversions, (Height Conversion *(Inches – Centimeter)*, Decimal – Binary and Distance Conversion *(from meter to foot)*).

This conversion App will help you calculate from the queue to its certain total number or amount.

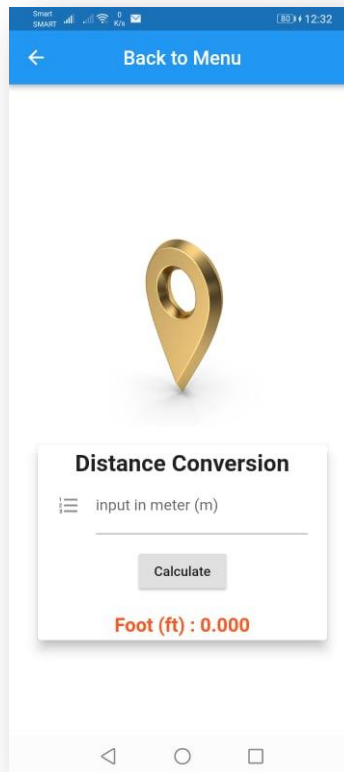
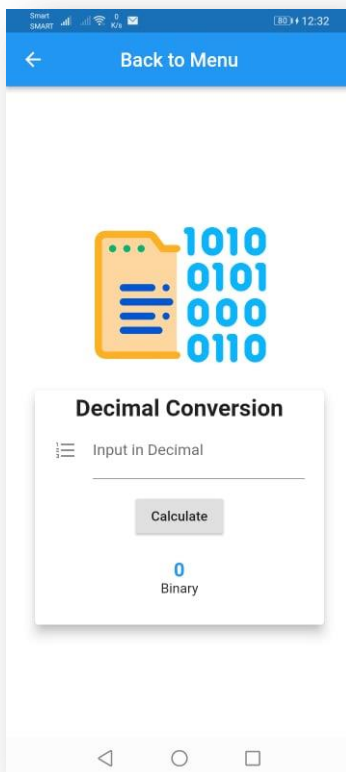
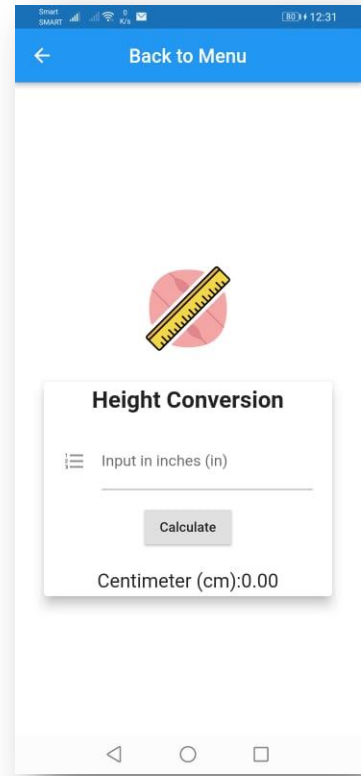
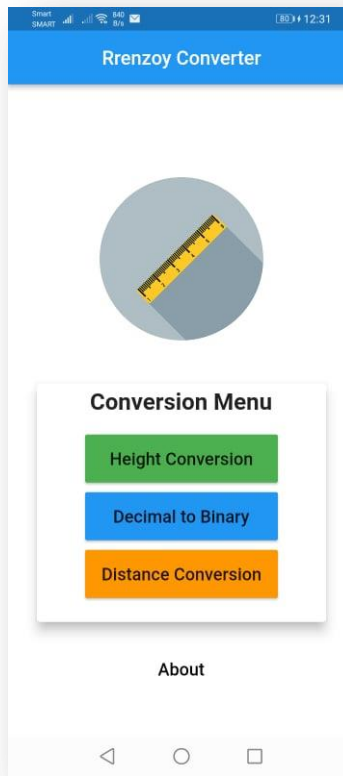
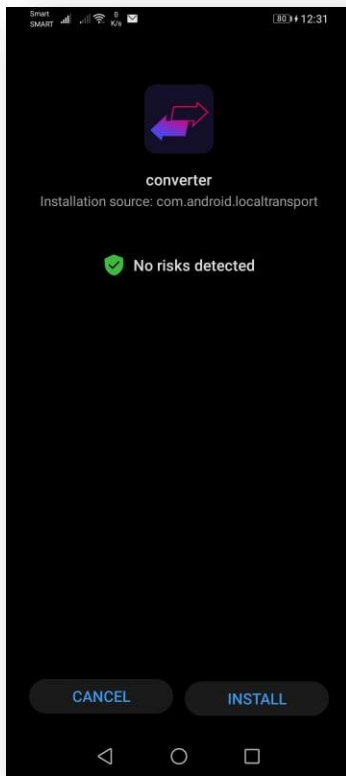


With strong determination and self-encouragement nothing is impossible, struggle and work hard, there you will enjoy success.

Acknowledgement

Special thanks to my parents who are there for me always and support especially with my financial needs, with my younger brother who let me use his laptop because mine doesn't meet the minimum requirement specifications for the said project software to use. I am also thankful to Mr. Marlon M. Bajolo, from the given task I able to exceed my beyond expectation knowledge of programming.

Preview



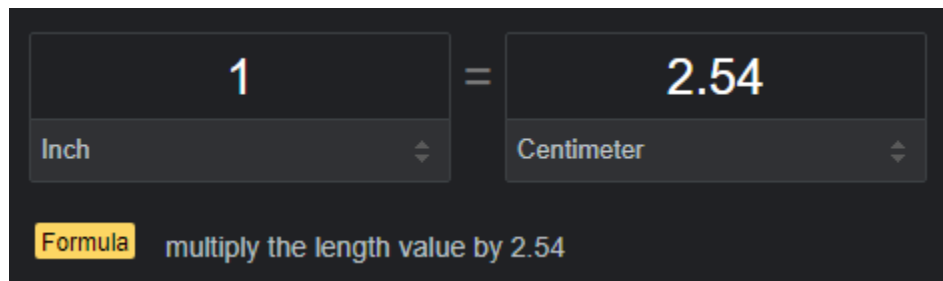
Theoretical Framework

Height conversion which its feature is to calculate the said height from inches (in) to centimeter (cm). There are 2.54 centimeters in 1 inch. Using the conversion number of 2.54, the conversion formula for converting x inches to centimeter is this: **x inches * 2.54 cm.**

1 In = Cm 2.54

In =?

Cm = 2.54



1 = 2.54

Inch Centimeter

Formula multiply the length value by 2.54

For example, to convert 5 inches to centimeter, we multiply 5 by 2.54.

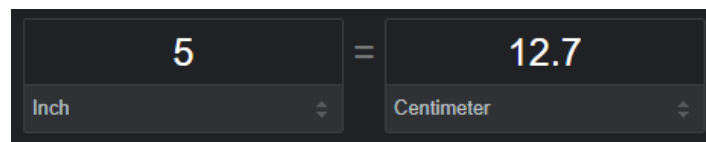
In =5

Cm = 2.54

In x cm = cm

$$\begin{array}{r} 2.54 \\ \times 5 \\ \hline 12.70 \end{array}$$

In 5 inches there is 12.7 centimeter.



5 = 12.7

Inch Centimeter

Decimal to Binary featured. Binary numbers are used for programming and coding in computers. Since a computer understands the language of binary digits, 0 and 1, numbers are converted from decimal to binary. To do the decimal to binary conversions, the base of the decimal numbers changes from 10 to 2.

Step 1: Divide the given number **13** repeatedly by 2 until you get '0' as the quotient

$$\begin{array}{lcl} 13 \div 2 = 6 & \text{(Remainder 1)} & \\ 6 \div 2 = 3 & \text{(Remainder 0)} & \\ 3 \div 2 = 1 & \text{(Remainder 1)} & \\ 1 \div 2 = 0 & \text{(Remainder 1)} & \end{array}$$


The diagram shows four horizontal lines corresponding to the remainders 1, 0, 1, and 1. From each line, an arrow points down to a digit in the sequence '1 1 0 1'. The arrows from the first and third lines point to the first and third digits (1s), while the arrows from the second and fourth lines point to the second and fourth digits (0 and 1).

Step 2: Write the remainders in the reverse order **1 1 0 1**

$$\therefore 13_{10} = 1101_2$$

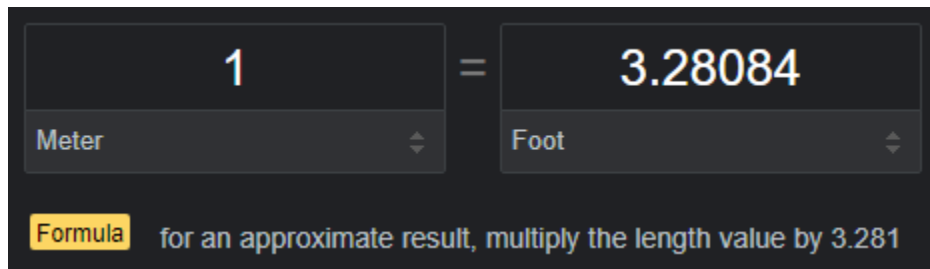
(Decimal) (Binary)

Distance conversion feature is to calculate the meter into foot. We use the formula 1 meter (m) = 3.28084 feet (ft) to convert meters to feet. Here, the conversion factor is 3.28084.

$$1\text{M} = \text{Ft } 3.28084$$

$$\text{M} = ?$$

$$\text{Ft} = 3.28084$$



A screenshot of a digital conversion tool interface. It features two input fields: the left one contains the number '1' and is labeled 'Meter' below it; the right one contains the result '3.28084' and is labeled 'Foot' below it. An equals sign '=' is positioned between the two fields. Below the input fields, there is a yellow button labeled 'Formula' followed by the text 'for an approximate result, multiply the length value by 3.281'.

For example, to convert 7 meters to feet, we multiply 7 by 3.28084.

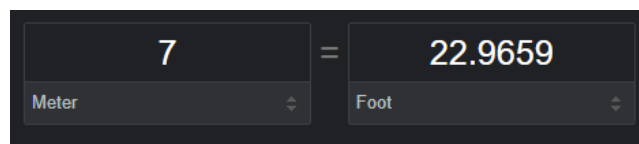
$$\text{M} = 7$$

$$\text{Ft} = 3.28084$$

$$\text{M} \times \text{Ft} = \text{Ft.}$$

$$\begin{array}{r} 3.28084 \\ \times \quad 7 \\ \hline 22.96588 \end{array}$$

In 7 meters there is 22.9659 foot.

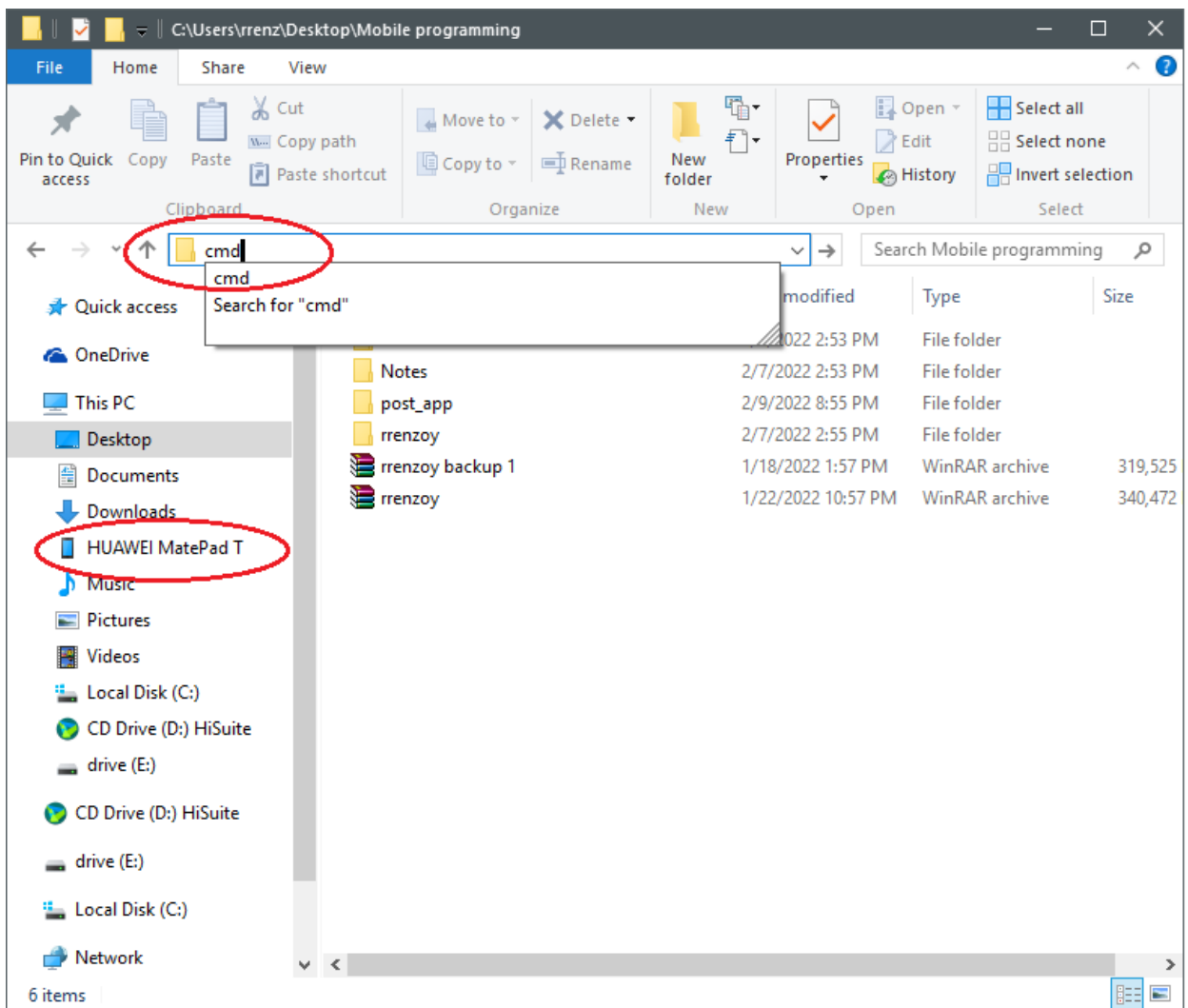


A screenshot of a digital conversion tool interface, similar to the one above. The left input field now contains the number '7' and is labeled 'Meter'. The right input field contains the result '22.9659' and is labeled 'Foot'. An equals sign '=' is between the fields. The 'Formula' button and its text are not visible in this screenshot.

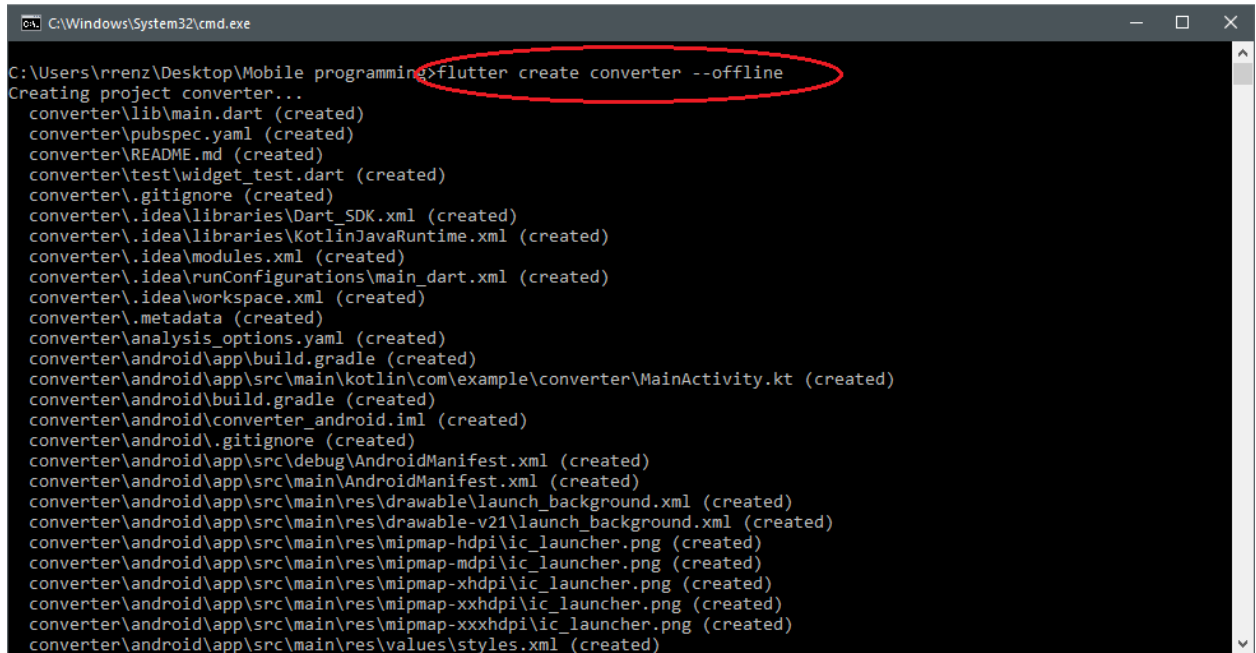
Source Code

Flutter Create

1. First, inside a folder or in my case “*Mobile programming folder*” type *cmd* to re-direct for your project location. I will also be using an external android device instead of android emulator. Device name “*Huawei MatePad T*”.

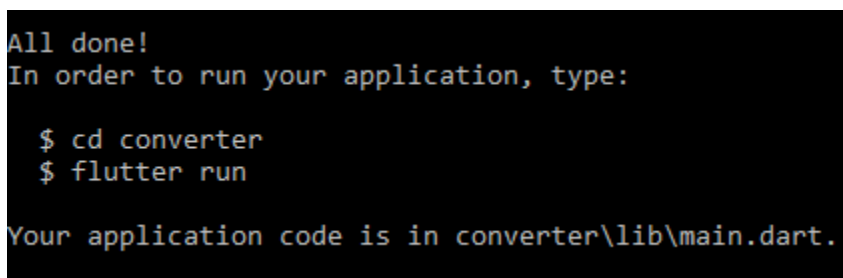


2. In cmd type “*flutter create converter --offline*”. I created a flutter project via offline mode. It is easier and less hassle with offline flutter project because you don’t need to depend on the internet downloading packages to your android device.



```
C:\Windows\System32\cmd.exe
C:\Users\rrenz\Desktop\Mobile programming>flutter create converter --offline
Creating project converter...
converter\lib\main.dart (created)
converter\pubspec.yaml (created)
converter\README.md (created)
converter\test\widget_test.dart (created)
converter\.gitignore (created)
converter\.idea\libraries\Dart_SDK.xml (created)
converter\.idea\libraries\KotlinJavaRuntime.xml (created)
converter\.idea\modules.xml (created)
converter\.idea\runConfigurations\main_dart.xml (created)
converter\.idea\workspace.xml (created)
converter\.metadata (created)
converter\analysis_options.yaml (created)
converter\android\app\build.gradle (created)
converter\android\app\src\main\kotlin\com\example\converter\MainActivity.kt (created)
converter\android\build.gradle (created)
converter\android\converter_android.iml (created)
converter\android\.gitignore (created)
converter\android\app\src\debug\AndroidManifest.xml (created)
converter\android\app\src\main\AndroidManifest.xml (created)
converter\android\app\src\main\res\drawable\launch_background.xml (created)
converter\android\app\src\main\res\drawable-v21\launch_background.xml (created)
converter\android\app\src\main\res\mipmap-hdpi\ic_launcher.png (created)
converter\android\app\src\main\res\mipmap-mdpi\ic_launcher.png (created)
converter\android\app\src\main\res\mipmap-xhdpi\ic_launcher.png (created)
converter\android\app\src\main\res\mipmap-xxhdpi\ic_launcher.png (created)
converter\android\app\src\main\res\mipmap-xxxhdpi\ic_launcher.png (created)
converter\android\app\src\main\res\values\styles.xml (created)
```

3. Now, my App is ready! “*cd converter*” command is to change directory to converter. Once you are on path you can simply command “*flutter run*” to deploy project to your android device.



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

$ cd converter
$ flutter run

Your application code is in converter\lib\main.dart.
```

4. This is how it looks like, if you have an old version of flutter project in your device, flutter will uninstall it and will install the new version of it.

```
C:\Windows\System32\cmd.exe - flutter run

C:\Users\rrenz\Desktop\Mobile programming>cd converter

C:\Users\rrenz\Desktop\Mobile programming\converter>flutter run
Launching lib\main.dart on KOB2 L09 in debug mode...
Parameter format not correct -
Running Gradle task 'assembleDebug'... 183.4s
✓ Built build\app\outputs\flutter-apk\app-debug.apk.
Installing build\app\outputs\flutter-apk\app.apk... 12.3s
Error: ADB exited with exit code 1
Performing Streamed Install

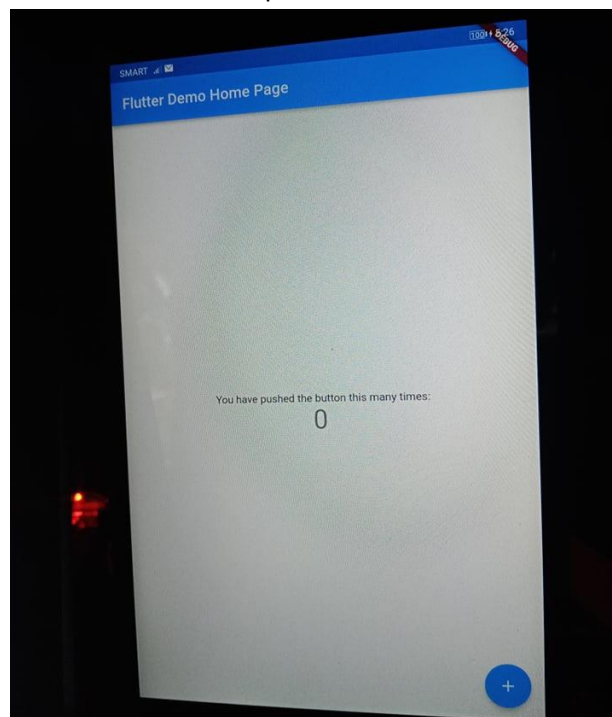
adb: failed to install C:\Users\rrenz\Desktop\Mobile programming\converter\build\app\outputs\flutter-apk\app.apk:
Failure [INSTALL_FAILED_UPDATE_INCOMPATIBLE: Package com.example.converter signatures do not match previously installed
version; ignoring!]
Uninstalling old version...
Installing build\app\outputs\flutter-apk\app.apk... 10.7s
D/AwareBitmapCacher(21843): handleInit switch not opened pid=21843
I/HwViewRootImpl(21843): removeInvalidNode jank list is null
Syncing files to device KOB2 L09... 727ms

Flutter run key commands.
r Hot reload.
R Hot restart.
h List all available interactive commands.
d Detach (terminate "flutter run" but leave application running).
c Clear the screen
q Quit (terminate the application on the device).

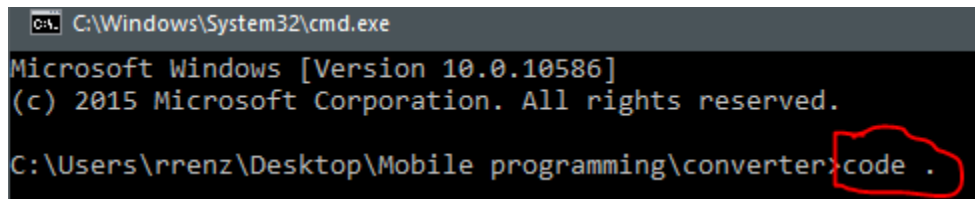
Running with sound null safety

An Observatory debugger and profiler on KOB2 L09 is available at: http://127.0.0.1:60161/qqWVfcNrs1w=/
The Flutter DevTools debugger and profiler on KOB2 L09 is available at:
http://127.0.0.1:9100?uri=http://127.0.0.1:60161/qqWVfcNrs1w=/
```

The output from device.



5. To run a Tool; in my case I am using Visual Studio Code. Therefore, “*code .*” to launch VScode directly to converter path.



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\rrenz\Desktop\Mobile programming\converter>code .
```

6. main.dart. I used *StatelessWidget* here since it is to used booting or where my functions next to splashscreen and homepage page.

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

import 'home.dart';
import 'splashscreen.dart';
import 'home.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      // theme: CustomTheme.ligththeme,
      routes: <String, WidgetBuilder>{
        '/loginScreen': (BuildContext context) => HomePage(),
        // '/registerscreen': (BuildContext context) => new
RegistrationScreen(),
        // '/mainScreen': (BuildContext context) => new MainScreen(),
        // '/gramscreen': (BuildContext context) => new TouringGramScreen(),
      },
      debugShowCheckedModeBanner: false,
      title: 'converter',
      theme: ThemeData(scaffoldBackgroundColor: const Color(0xffffffff)),
      home: SplashScreen();
    }
  }
```

SplashScreen

7. splashscreen.dart

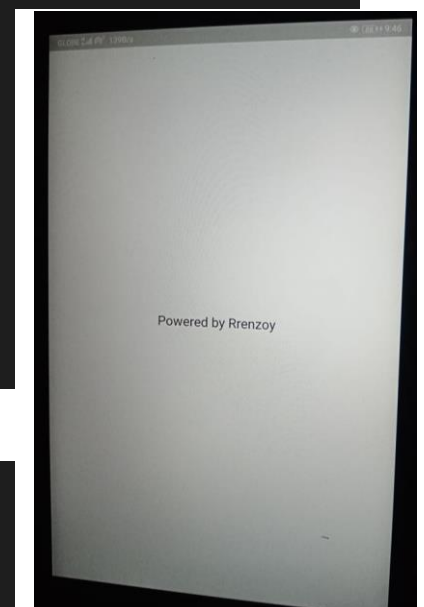
```
import 'dart:async';
import 'package:flutter/material.dart';
import 'home.dart';

class SplashScreen extends StatefulWidget {
  @override
  _SplashScreenState createState() => _SplashScreenState();
}

class _SplashScreenState extends State<SplashScreen> {
  @override
  void initState() {
    super.initState();
    Timer(
      Duration(seconds: 3),
      () => Navigator.pushReplacement(
        context, MaterialPageRoute(builder: (content) => HomePage())));
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Container(
              margin: EdgeInsets.all(50),
              // child:
              Image.asset('assets/images/logo.jpg'),
              child: const Text(
                "Powered by Rrenzoy",
                style: TextStyle(fontSize: 20),
              ),
            ),
          ],
        ),
      ),
    );
  }
}
```

This will be the display text in splashscreen

```
child: const Text(
  "Powered by Rrenzoy",
  style: TextStyle(fontSize: 20),
),
```

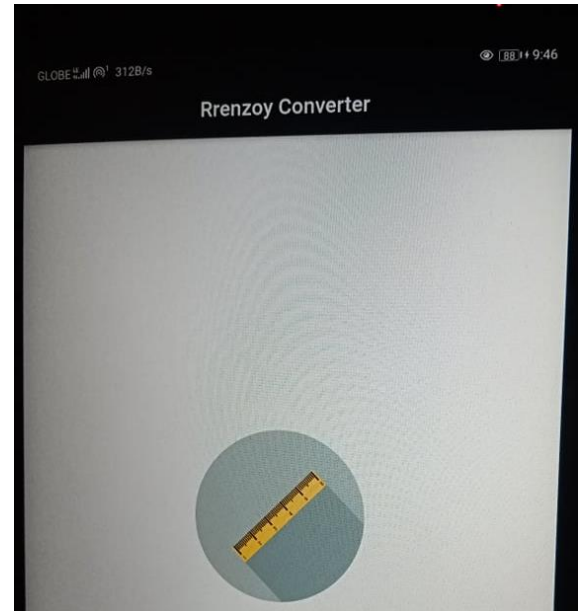


Home

8. home.dart

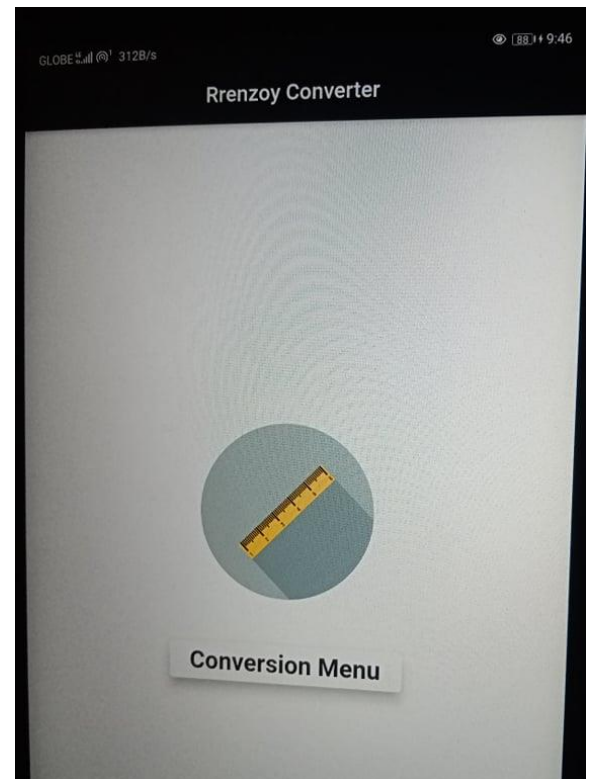
For the image/ logo name ruler an extension type png.

```
Container(  
    margin: EdgeInsets.fromLTRB(70,  
50, 70, 10),  
    child:  
Image.asset('assets/images/ruler.png', scale:  
2)),
```



Applying "Conversion menu" title in pad.

```
Card(  
    margin: EdgeInsets.fromLTRB(30, 5,  
30, 15),  
    elevation: 10,  
    child: Padding(  
        padding: const  
EdgeInsets.fromLTRB(20, 5, 20, 5),  
        child: Column(  
            children: [  
                Text(  
                    'Conversion Menu',  
                    style: TextStyle(  
                        fontWeight:  
FontWeight.bold,  
                        fontSize: 24,  
                    ),  
                ),  
            ],  
        ),  
    ),
```



For space enter or for between two widgets I used sizebox.

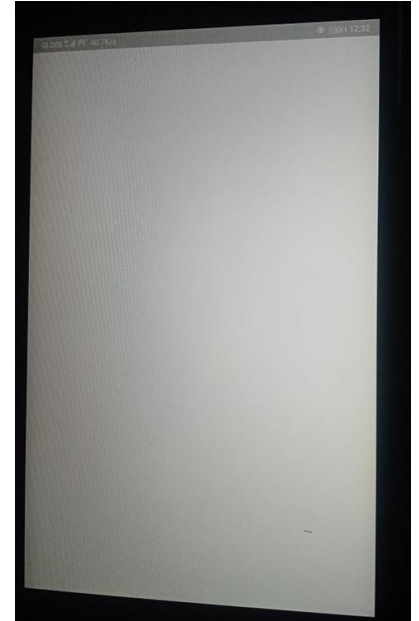
```
SizedBox(height: 30),
```

Height Menu

Create page for the next page named Height Conversion

9. height.dart (blank page)

```
10. import 'package:flutter/material.dart';
11. import 'home.dart';
12.
13. class HeightConversion extends StatefulWidget {
14.   @override
15.   _HeightConversionState createState() =>
16.     _HeightConversionState();
17.
18. class _HeightConversionState extends
19.   State<HeightConversion> {
20.   TextEditingController numztextEditingController =
21.     new TextEditingController();
22.   double results = 0.0;
23.   @override
24.   Widget build(BuildContext context) {
25.     return Scaffold(
26.       body: Center(
27.         child: SingleChildScrollView(),
28.       ),
29.     );
30.   }
31.
```



Home /Conversion Menu

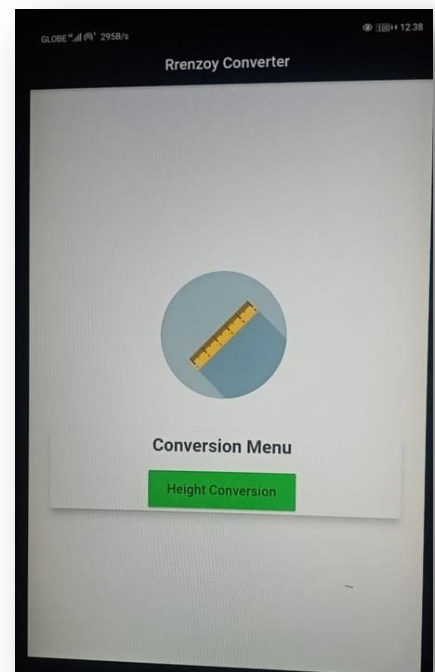
10. home.dart - this row is for the button when press it will navigate to the next page where in Height menu.

```
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: <Widget>[
    SizedBox(
      width: 200,
      height: 50,
      child: RaisedButton(
        child: Text(
          "Height Conversion",
```

```

style: TextStyle(fontSize: 18),
),
color: Colors.green,
onPressed: () => Navigator.push(
context,
MaterialPageRoute(
builder: (content) => HeightConversion()),
),
),
],
)

```



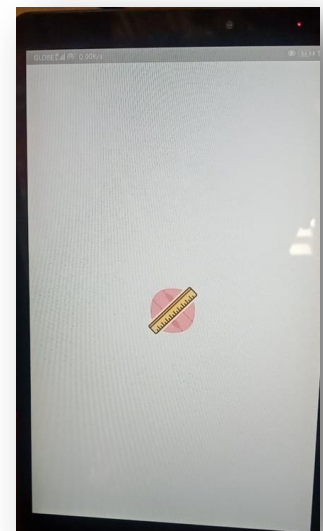
Height Menu

11. height.dart

```

Container(
  margin: EdgeInsets.fromLTRB(70, 50, 70, 10),
  child:
Image.asset('assets/images/height.png', scale: 2),
)

```

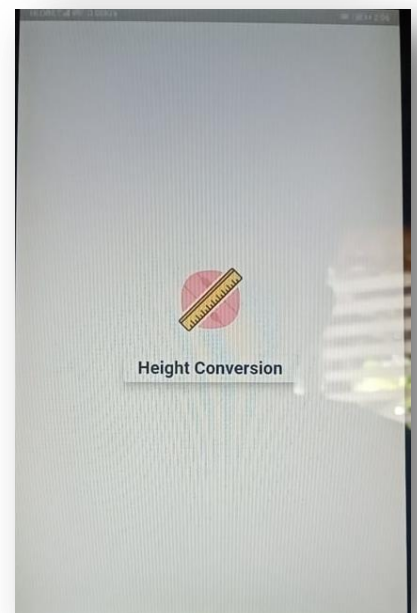


I added a title '**Height Conversion**' with a font size and a font weight as well as its edges.

```

Card(
  margin: EdgeInsets.fromLTRB(30, 5, 30, 15),
  elevation: 10,
  child: Padding(
    padding: const EdgeInsets.fromLTRB(20,
5, 20, 5),
    child: Column(
      children: [
        Text(
          'Height Conversion',

```



```

        style: TextStyle(
          fontWeight: FontWeight.bold,
          fontSize: 24,
        ),
      ),
    ],
  )),
)

```

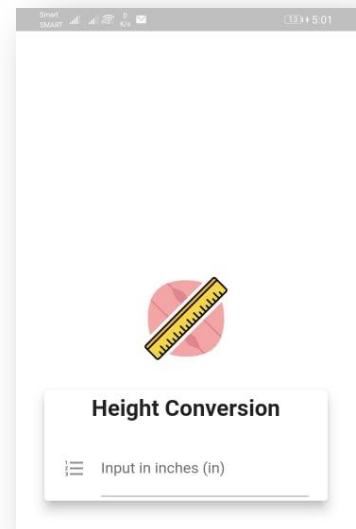
Text field for numbers inputs.

(I used screenshots this time with a phone not in Huawei tablet due for some reasons.)

```

TextField(
  controller: numztextEditingController,
  keyboardType: TextInputType.number,
  decoration: InputDecoration(
    labelText: 'Input in inches (in)',
    icon: Icon(Icons.format_list_numbered),
  ),
)

```

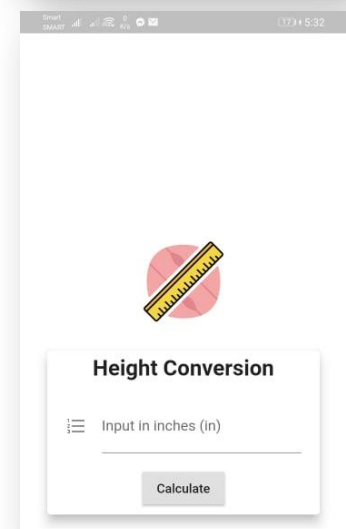


Button and calculation

```

Row(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: [
    RaisedButton(
      child: Text("Calculate"),
      onPressed: () => calculateMe(1),
    ),
  ],
)

```



To display the results calculation from inches to centimeter.

```
Text(
  "Centimeter (cm):" + results.toStringAsFixed(2),
  style: TextStyle(fontSize: 20),
),

void calculateMe(int op) {
  setState(() {
    double numz =
double.parse(numztextEditingController.text);

    if (op == 1) {
      results = numz * 2.54;
      //print(results);
    }
  });
}
```

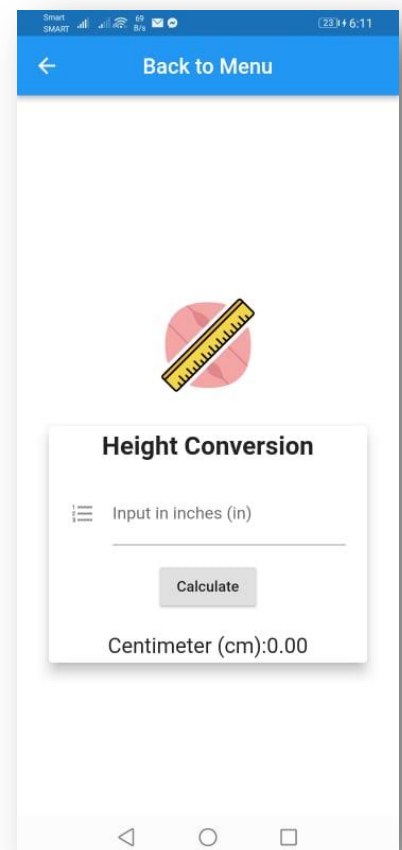


On pressed to call the function void `onPressed: () => calculateMe(1),`

\

I added an AppBar here with back button.

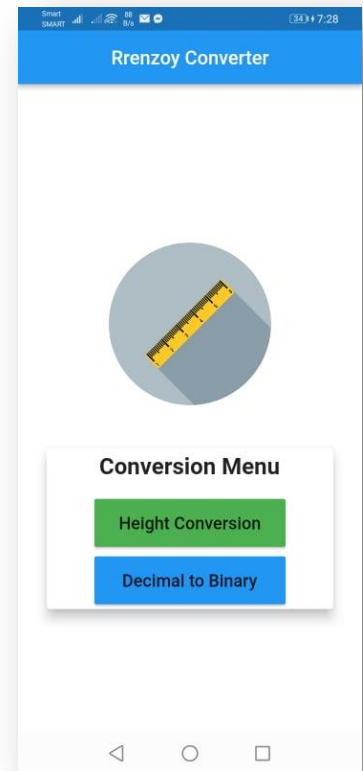
```
appBar: AppBar(
  backgroundColor: Colors.blue,
  iconTheme: IconThemeData(
    color: Colors.white, //change your color here
  ),
  title: Text("Back to Menu"),
  centerTitle: true,
),
```



Home/Conversion menu

12. Home.dart Added "Decimal to Binary" Button

```
Row(  
  mainAxisAlignment: MainAxisAlignment.center,  
  children: <Widget>[  
    SizedBox(  
      width: 200,  
      height: 50,  
      child: RaisedButton(  
        child: Text("Decimal to Binary",  
          style: TextStyle(fontSize: 18),  
        ),  
        color: Colors.blue,  
        onPressed: () => Navigator.push(  
          context,  
          MaterialPageRoute(  
            builder: (content) => DecimalConversion()),  
          ),  
        ),  
      ],  
    ),  
  ),  
)
```



13. decimal.dart

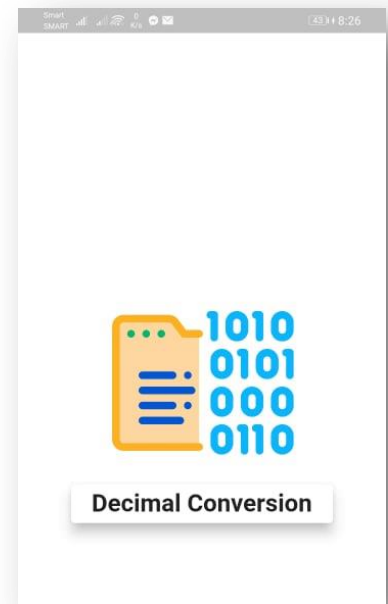
```
body: Center(  
  child: SingleChildScrollView(  
    child: Column(  
      children: [  
        Container(  
          margin: EdgeInsets.fromLTRB(70, 50, 70,  
10),  
          child: Image.asset(  
            'assets/images/binary.jpg',  
            scale: 2,  
          ),  
        ),  
      ],  
    ),  
  ),  
)
```



```

child: Padding(
  padding: const EdgeInsets.fromLTRB(20, 5, 20, 5),
  child: Column(
    children: [
      Text(
        'Decimal Conversion',
        style: TextStyle(
fontWeight: FontWeight.bold, fontSize: 24),
      ),
    ],
  ),
),

```

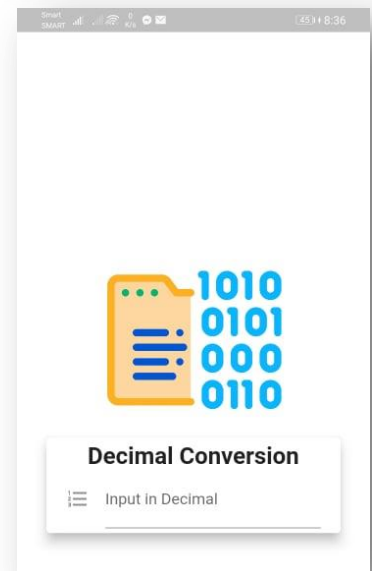


Text Field.

```

TextField(
  controller: numzTextEditingController,
  keyboardType: TextInputType.number,
  decoration: InputDecoration(
    labelText: 'Input in Decimal',
    icon: Icon(Icons.format_list_numbered),
  ),
)

```



Button.

```

Row(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: [
    RaisedButton(
      child: Text("Calculate"),
      onPressed: () => calculateMe(1),
    ),
  ],
)

```

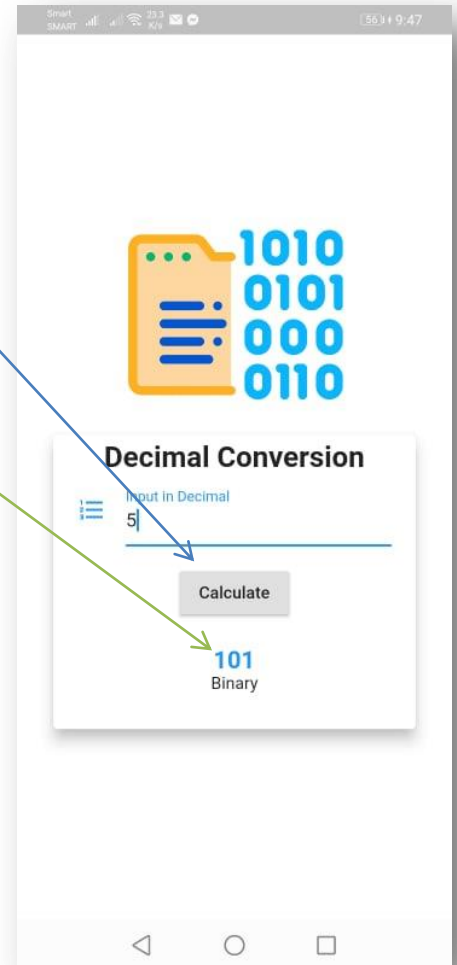
Display Results.

```
Text(
    results.toStringAsFixed(0),
    style: TextStyle(
        color: Colors.blue,
        fontWeight:
FontWeight.bold,
        fontSize: 20),
    ),
    Text("Binary")
```

Binary Calculation.

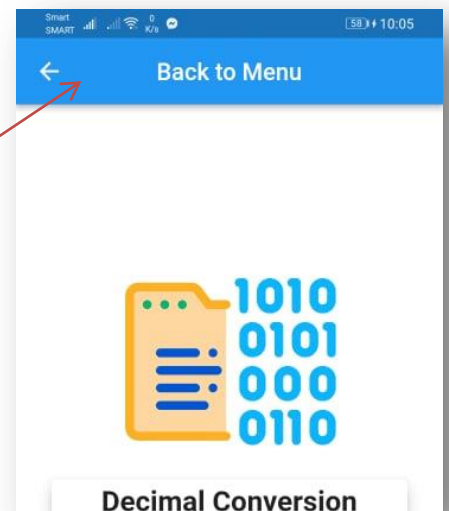
```
void calculateMe(int op) {
    setState(() {
        int numz =
int.parse(numzTextEditingController.text);

        if (op == 1) {
            int bin = 0, i = 1;
            while (numz > 0) {
                bin = bin + (numz % 2) * i;
                numz = (numz / 2).floor();
                i = i * 10;
            }
            results = bin;
        }
    });
}
```



App Bar with Back Button.

```
appBar: AppBar(
    backgroundColor: Colors.blue,
    iconTheme: IconThemeData(
        color: Colors.white, //change your color
here
    ),
    title: Text("Back to Menu"),
    centerTitle: true,
),
```

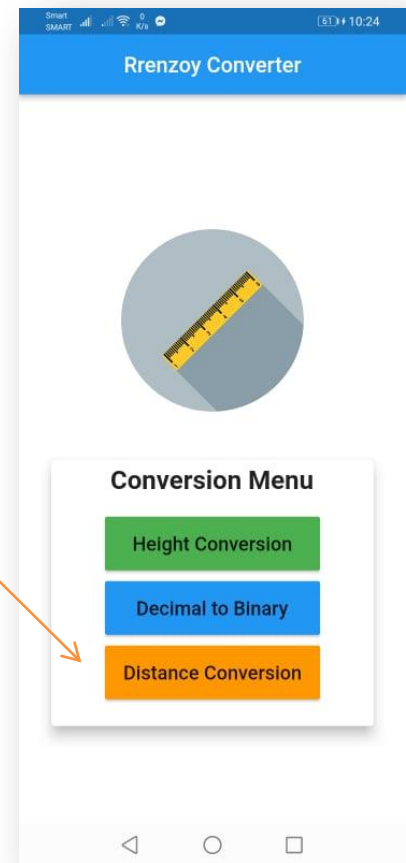


Home

14. Import distance.dart for the function to be functional.

The 'Distance Conversion Menu' button created.

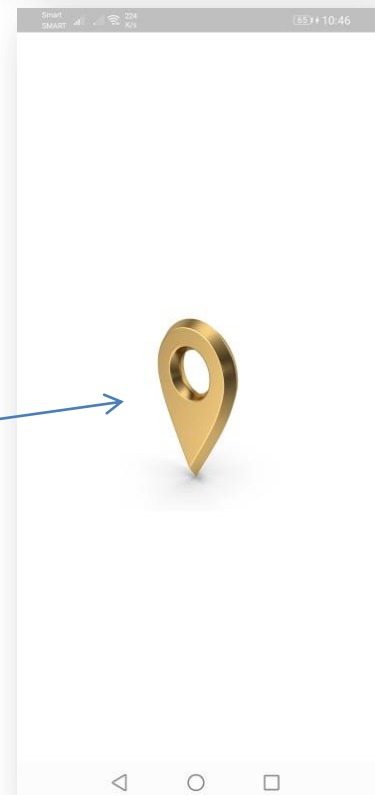
```
Row(  
  mainAxisAlignment: MainAxisAlignment.center,  
  children: <Widget>[  
    SizedBox(  
      width: 200,  
      height: 50,  
      child: RaisedButton(  
        child: Text(  
          "Distance Conversion",  
          style: TextStyle(fontSize: 18),,  
          color: Colors.orange,  
          onPressed: () => Navigator.push(  
            context,  
            MaterialPageRoute(  
              builder: (content) =>  
                DistanceConversion()),  
            ),  
          ),  
        ],  
      ),  
    ],  
  ),  
)
```



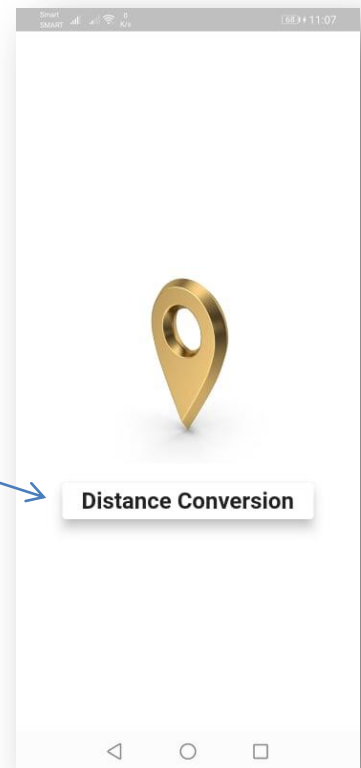
Distance Menu

15. distance.dart

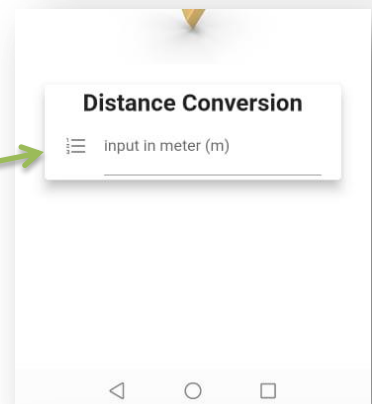
```
Scaffold(  
  body: Center(  
    child: SingleChildScrollView(  
      child: Column(  
        children: [  
          Container(  
            margin: EdgeInsets.fromLTRB(70, 50, 70, 10),  
            child:  
              Image.asset('assets/images/distance.jpg', scale: 2)),  
          SizedBox(  
            height: 5,  
          ),  
        ],  
      ),  
    ),  
  ),  
)
```



```
Card(
  margin: EdgeInsets.fromLTRB(30, 5, 30, 15),
  elevation: 10,
  child: Padding(
    padding: const EdgeInsets.fromLTRB(20, 5, 20, 5),
    child: Column(
      children: [
        Text(
          'Distance Conversion',
          style: TextStyle(
            fontWeight: FontWeight.bold,
            fontSize: 24,
          ),
        ),
      ],
    ),
  ),
)
```

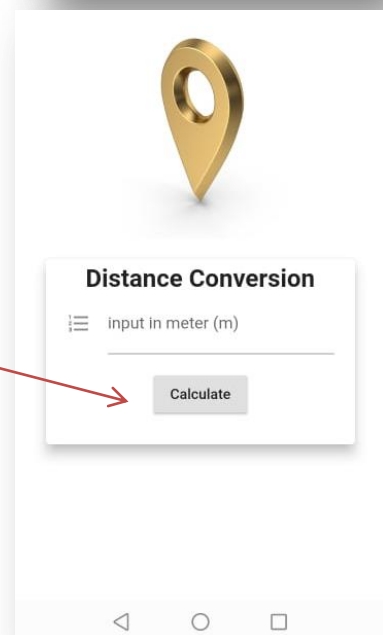


```
TextField(
  controller: numzTextEditingController,
  keyboardType:
TextInputType.number,
  decoration: InputDecoration(
    labelText: 'input in meter (m)',
    icon: Icon(Icons.format_list_numbered),
  ),
)
```



Button

```
Row(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: [
    RaisedButton(
      child: Text("Calculate"),
      onPressed: () => calculateMe(1),
    ),
  ],
)
```

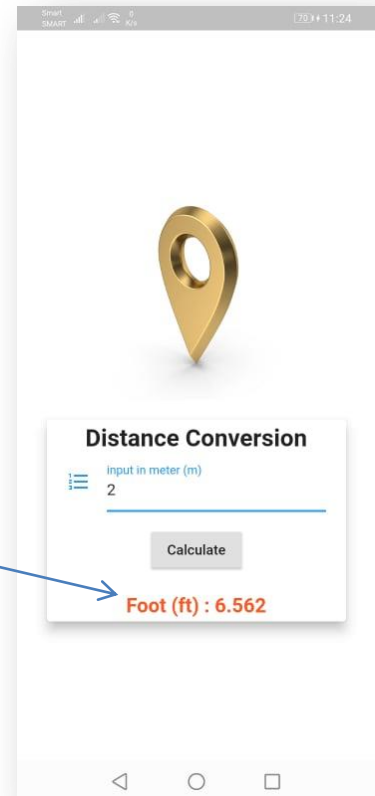


Distance Calculation from meter to foot.

```
void calculateMe(int op) {  
    setState(() {  
        double numz =  
double.parse(numzTextEditingController.text);  
        results = numz * 3.281;  
    });  
}
```

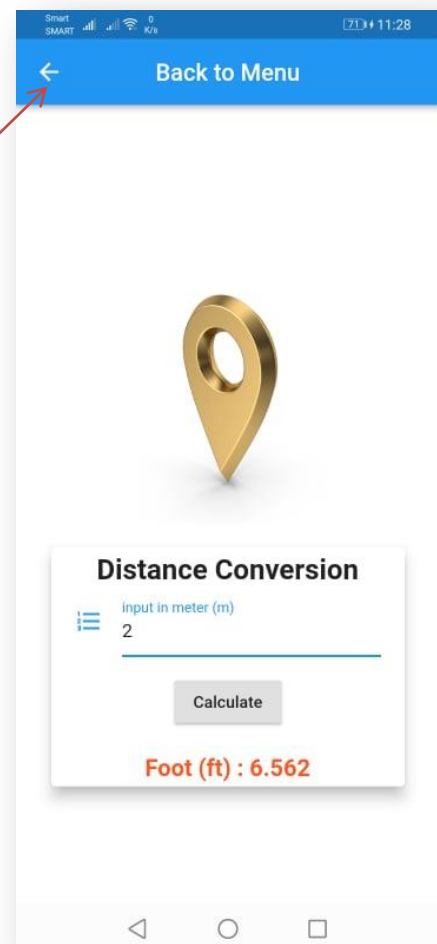
Display Results.

```
Text(  
    "Foot (ft) : " + results.toStringAsFixed(3),  
    style: TextStyle(  
        color: Colors.deepOrange,  
        fontWeight: FontWeight.bold,  
        fontSize: 20,  
    ),  
),
```



App Bar added with back button.

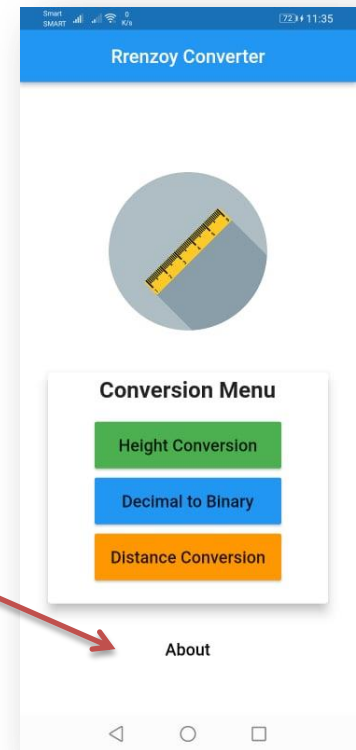
```
appBar: AppBar(  
    backgroundColor: Colors.blue,  
    iconTheme: IconThemeData(  
        color: Colors.white, //change your color  
here  
    ),  
    title: Text("Back to Menu"),  
    centerTitle: true,  
),
```



About

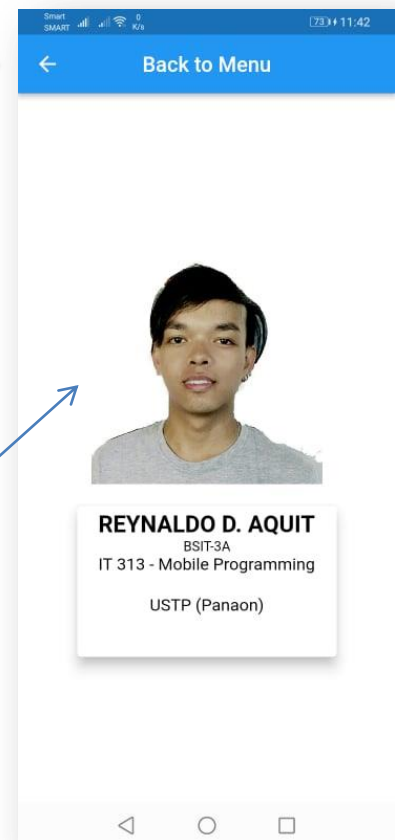
17. about.dart *Button*.

```
TextButton(  
  child: Text(  
    "About",  
    style: TextStyle(fontSize: 18, color:  
Colors.black),  
  ),  
  
  onPressed: () =>  
    Navigator.push(  
      context,  
      MaterialPageRoute(  
        builder: (content) =>  
          MyAbout()),  
    ),  
)
```

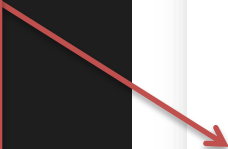


Description

```
return Scaffold(  
  appBar: AppBar(  
    backgroundColor: Colors.blue,  
    iconTheme: IconThemeData(  
      color: Colors.white, //change your color here  
    ),  
    title: Text("Back to Menu"),  
    centerTitle: true,  
  ),  
  body: Center(  
    child: SingleChildScrollView(  
      child: Column(  
        children: [  
          Container(  
            margin: EdgeInsets.fromLTRB(70, 50, 70, 10),  
            child: Image.asset('assets/images/renz.jpg', scale:  
2)),  
          SizedBox(height: 5),  
          Card(  
            margin: EdgeInsets.fromLTRB(30, 5, 30, 15),  
            elevation: 10,  
            child: Padding(  
              padding: const EdgeInsets.fromLTRB(20, 5, 20, 5),  
            ),  
          ),  
        ],  
      ),  
    ),  
  ),  
)
```




```
t.bold,  
  
normal,  
  
gramming',  
  
t.normal,
```

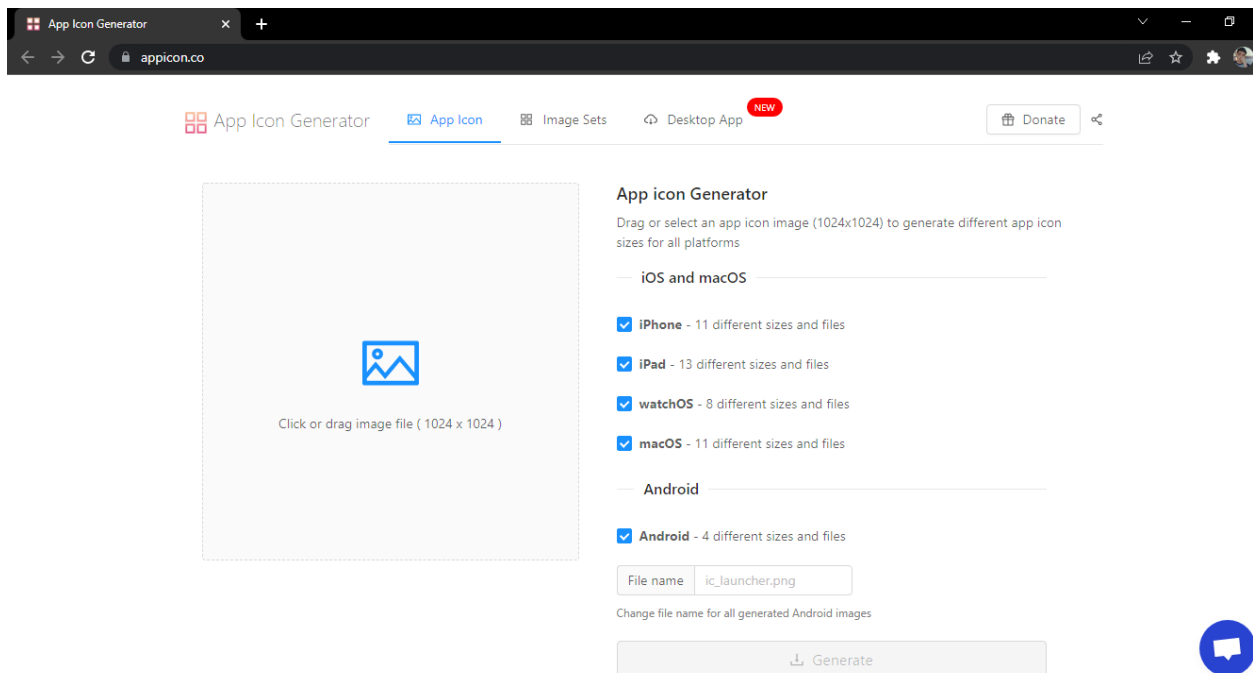


Changing App Icon

There are two ways of how to change an App Icon. In my pubspec.yaml using package appcarch doesn't work so switch to Manual Approach changing icon.

Step 1. Generating different sized icons.

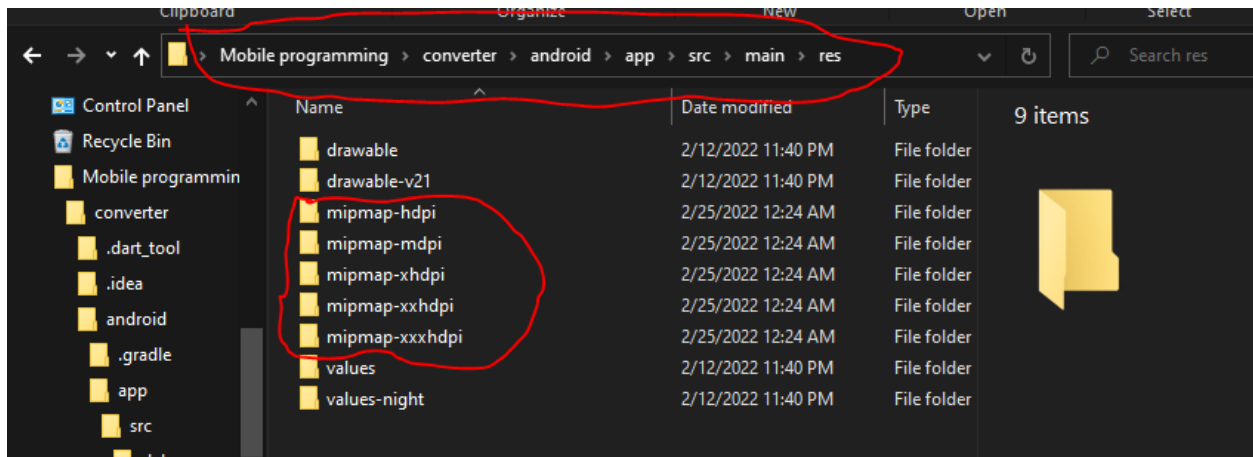
Upload image to <https://appicon.co/> and tick and iphone and android options and click on generate. The site generates different sized icons for both Android and IOS at the same time.



It will Download the Zip file named **AppIcons** with the android and **Assets.xcassets** named folders along with images for appstore and playstore which can be directly uploaded as an icon in both the stores

Step 2. Open project folder.

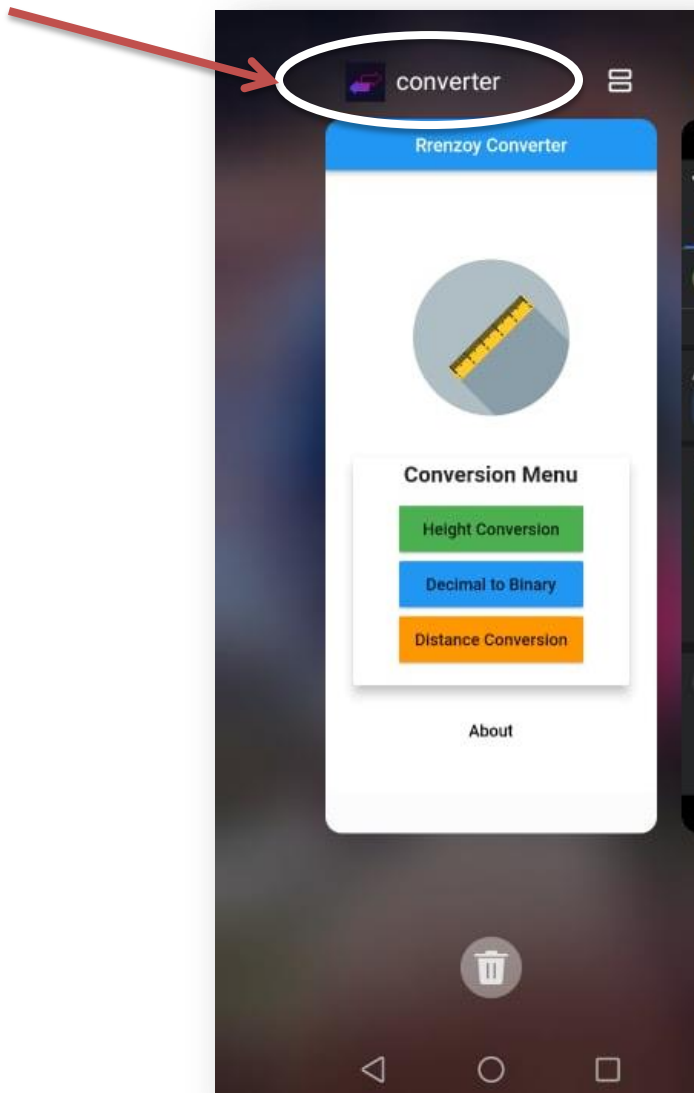
Navigate to **android/app/src/main/res** and right-click on res folder and click “reveal in Explorer”. Now delete all the **mipmap** folders in res folder and paste the mipmap folders from **AppIcon/android** folder which you have downloaded.



Step 4: Run the Application

After manually changing the images in android and IOS folders now go to **lib/main.dart** and run the flutter project in the flutter console.

- ✓ Source: <https://www.geeksforgeeks.org/flutter-changing-app-icon/>



Thank You!