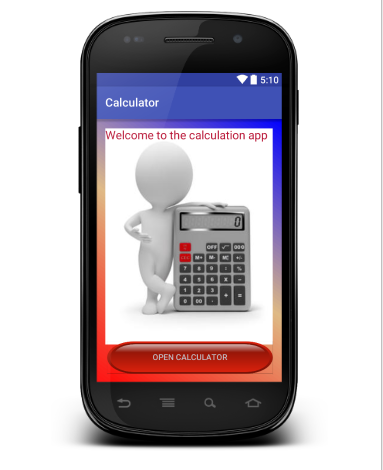
**Calculator**

The app was designed to simply perform some basic calculations such as addition, subtraction, multiplication and division. It provides the user the facility to reset and delete the numbers.

**Features of the application:**

1. Layout
2. Coding
3. Result
4. Layout
5. Welcome Layout



The layout of this section was designed using the TextView and Button views. TextView is used to display the welcome text. A button is used, indicating the user to press it so that he/she can go to the calculator portion. The xml file has some of the additional background images and the gradient resources used.

<gradient  
 android:startColor="#0000ff"  
 android:centerColor="#deb984"  
 android:endColor="#ff0000"  
 android:type="linear"  
 android:angle="225"/>

The above coding was done to create the gradient effect in the background. startColor, centerColor, endColor ,type and angle must be defined in this section.

1. Calculation section



The layout of this section was designed using the TextView and Buttons where two different layout styles were used i.e. Relative layout and Linear Layout. The buttons are created using the images.

1. Coding
2. Welcome Layout

For this part of coding, a different activity was created called “WelcomeActivity.java”. We make use of the intent object. The code using intent can send the user to the system defining the components that we are targeting. Using the startActivity method we can define the intent to start the activity.

Button btnOpenCalculator = (Button) findViewById(R.id.btnOpenCalculator);  
  
btnOpenCalculator.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
  
 Intent intent = **new** Intent(WelcomeActivity.**this**, MainActivity.**class**);  
 startActivity(intent);

In the above code, the intent is created that defines the targeted components. The startActivity method defines which activity to start.

1. Calculation app

Here, the objects are defined for the TextView and Buttons views. A string global variable and a float global variable are defined in the MainActivity java file. Then all the views are assigned to the object by the findViewById function. Then using the setOnClickListener for the defined object creates an interface so that, when the button is pressed the code written inside it will be executed..

**btnAc**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
  
 **value** = **""**;  
 **tvDisplay**.setText(**"0"**);  
 **tvDisplaySymbol**.setText(**""**);  
 }  
});

Here is an example for the use of the setOnClickListener. When the button having object btnAc is pressed the codes written inside are executed.

1. Result

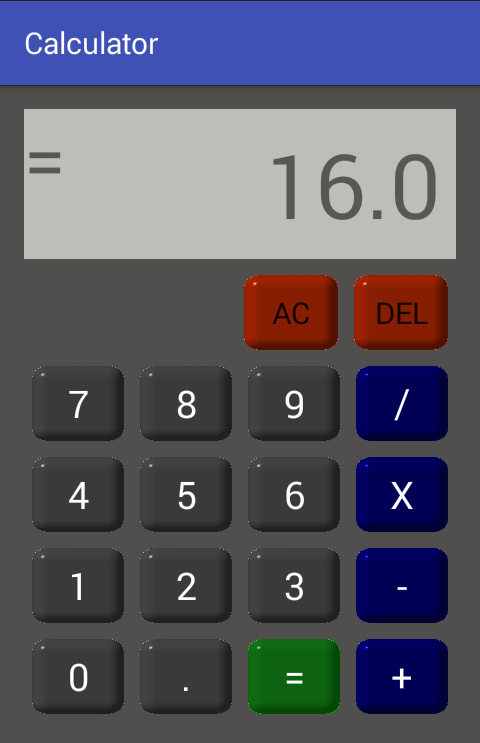


Figure 1: Welcome page layout Figure 2: Calculator

Two figures are listed above as figure 1 and figure 2. Figure 1 shows the welcome page layout of the app calculator. It consists of the gradient background with a image. A text is written to welcome the app users to the app itself and a button is placed at the bottom of the screen. The button is so created that when pressed it will lead the user to the actual calculator section i.e. figure 2. Figure 2 is the calculator section of the app where the users can actually enter the values and perform the basic calculation of addition, subtraction, multiplication and division. This section of the app also contains the delete and the reset buttons as well so that users can delete the numbers or reset the value back to empty so that new values can be entered for some other calculation.