Aim :

Write an code of implementation of Capturing an Image using Camera & display it.

Theory :

1.**Camera Manager**: This is used to get all the cameras available in the device like front camera back camera each having the camera id.

2.**CameraDevice**: You can get it from Camera Manager class by its id.

3.**CaptureRequest**: You can create a capture request from camera device to capture images.

4.**CameraCaptureSession**: To get capture request's from Camera Device create a CameraCaptureSession.

5.**CameraCaptureSession.CaptureCallback**: This is going to provide the Capture session results.

Algorithm :

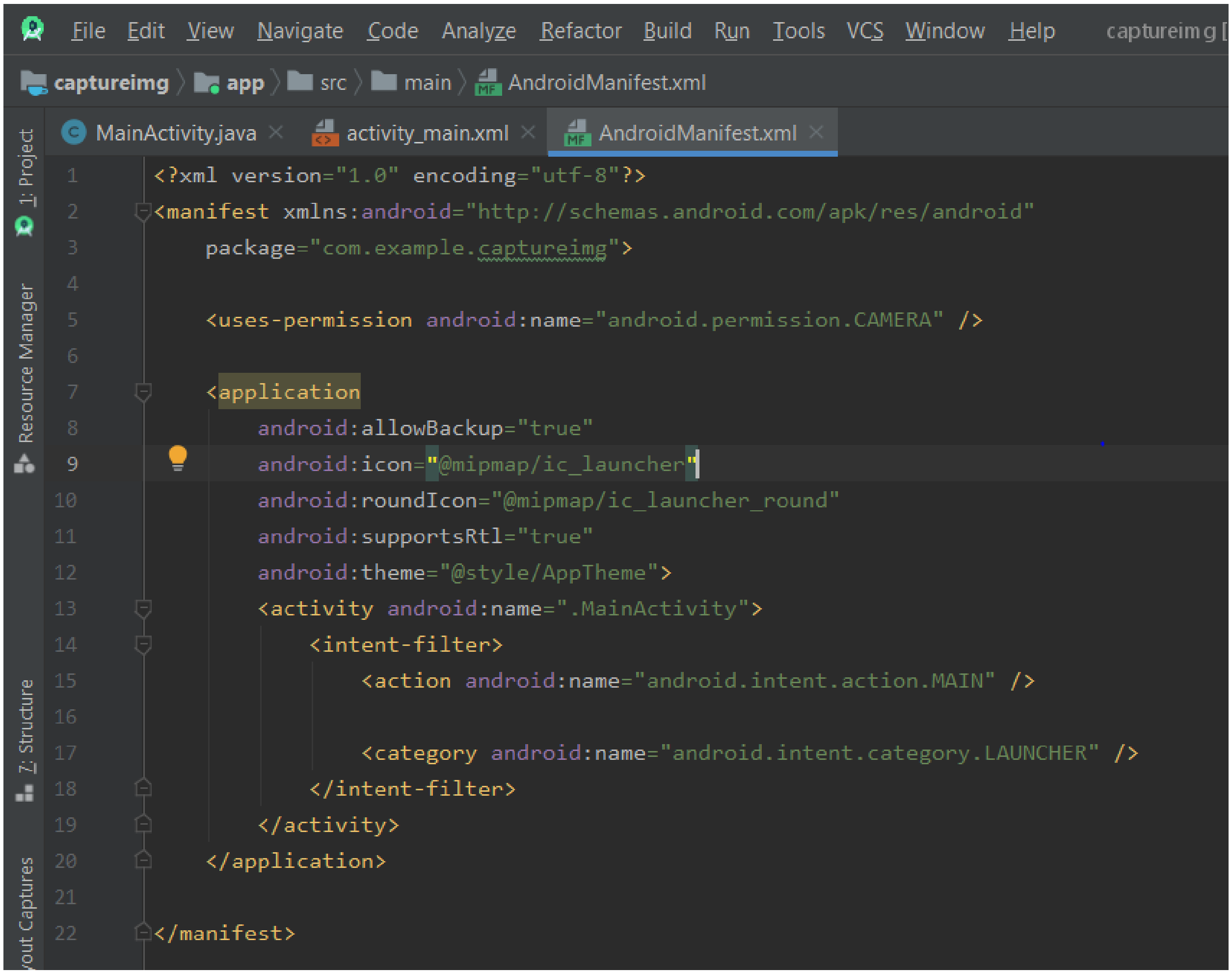
1. Create a new project in **Android Studio** from **File ⇒ New Project** by filling the required details. When it prompts you to select the activity, choose **Empty Activity** and continue.
2. Open **AndroidManifest.xml** and add the following permissions :
o advertise that your application depends on having a camera, put a **<uses-feature>** tag of **name="android.hardware.camera"** in your manifest file.By adding **android.hardware.camera**, Play Store detects and prevents installing the application on devices with no camera.
By setting **android:required** to **false** ,Google Play will allow devices without a camera to download your application.
3. Open **activity_main.xml** and write the below code.It contains one **imageView** and two buttons **Camera** and **Gallery** ,**btn_camera** is to capture image from camera and **btn_gallery** is to pick image from gallery.
4.Intent is the standard way to delegate actions to another application.

- To start the native camera the Intent requires **android.provider.MediaStore.ACTION_IMAGE_CAPTURE**.
- To pick an image from gallery, the Intent requires the following argument : **Intent.ACTION_PICK**.

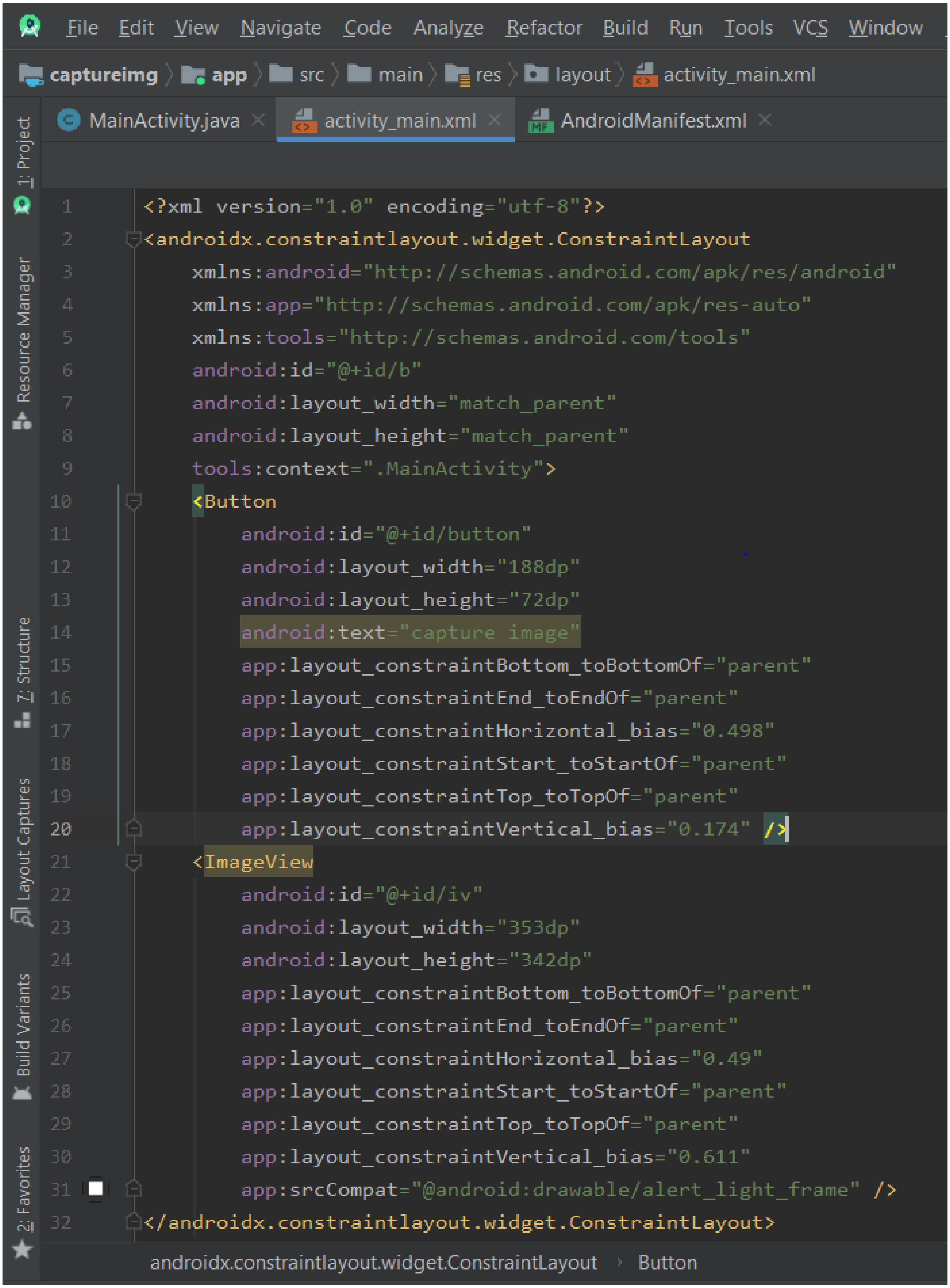
5.Now we know how to capture and pick image from camera and gallery now its time to set image in an **imageView**.
startActivityForResult() contains intent and related request code returns an intent to **onActivityResult**() .Inside onActivityResult().

Code :

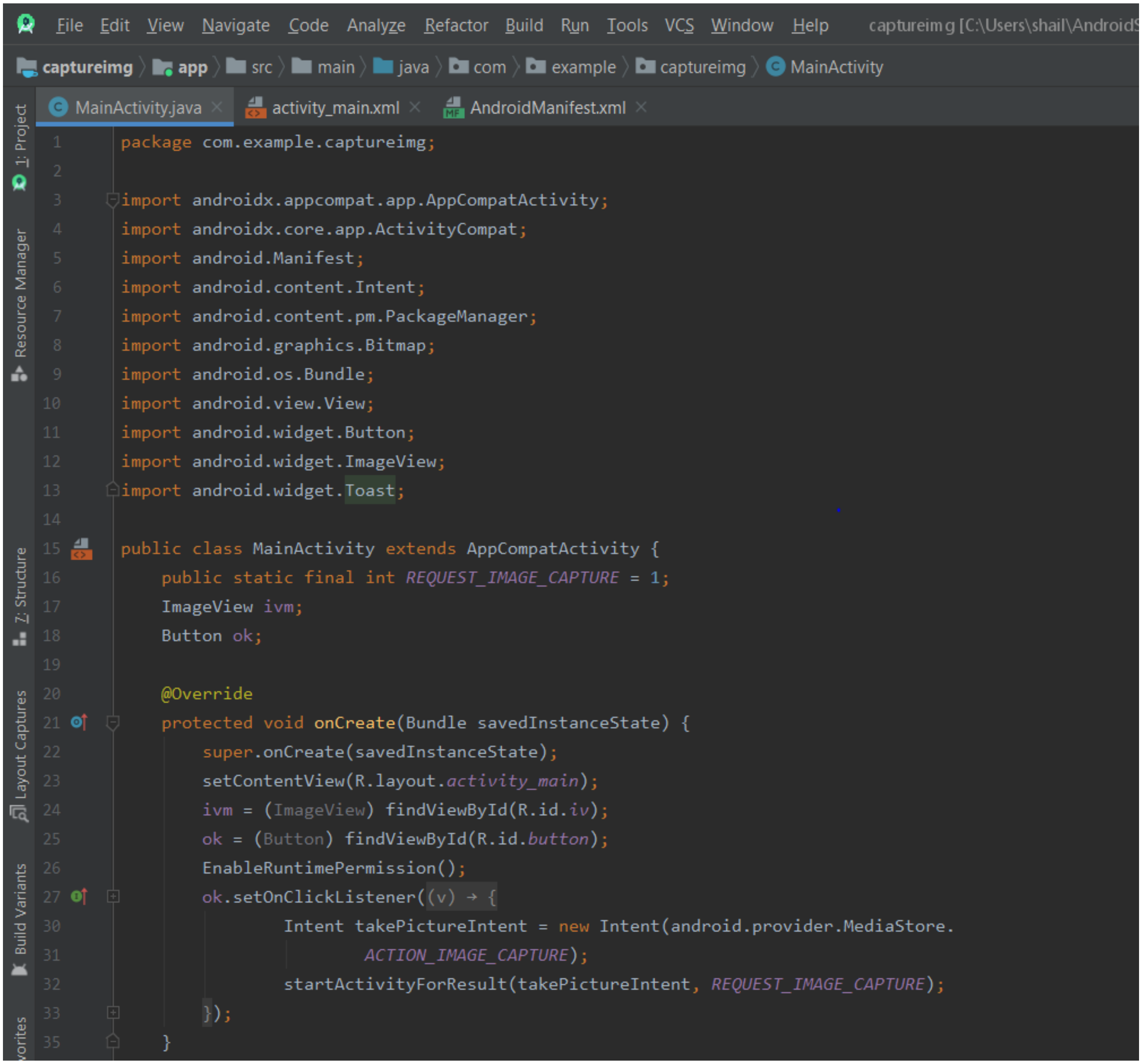
Output :



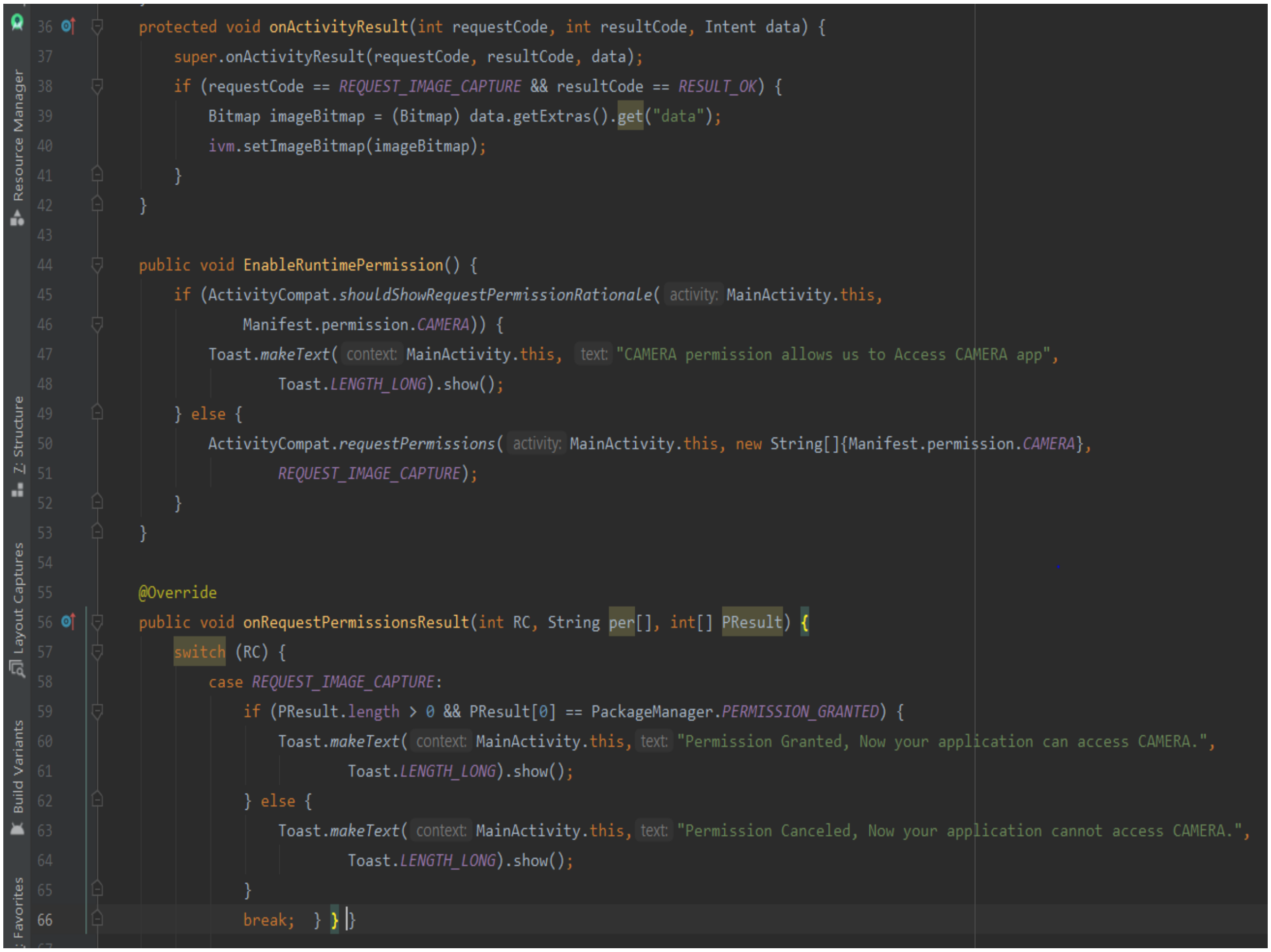
```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.captureimg">

    <uses-permission android:name="android.permission.CAMERA" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```



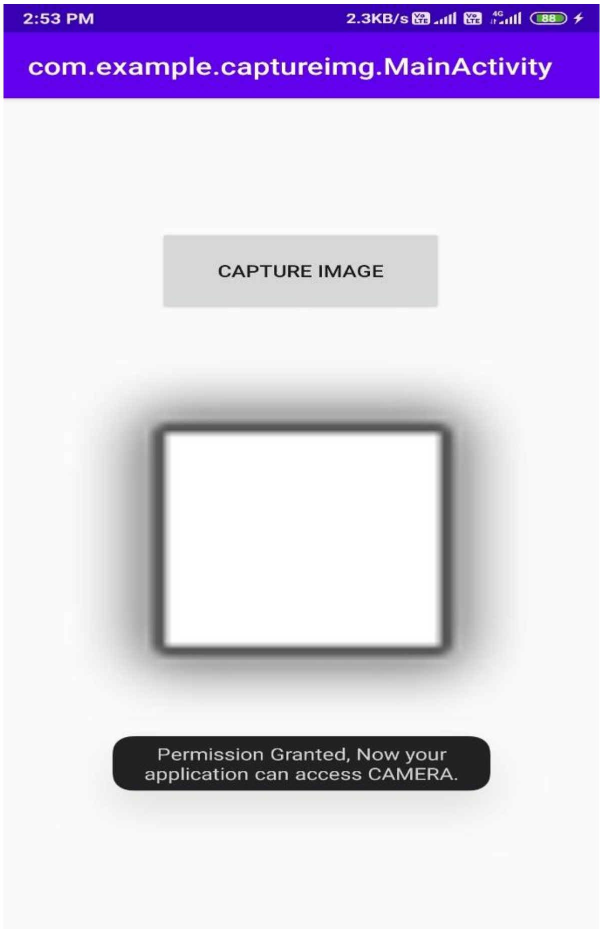
```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/b"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/button"
        android:layout_width="188dp"
        android:layout_height="72dp"
        android:text="capture image"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.174" />
    <ImageView
        android:id="@+id/iv"
        android:layout_width="353dp"
        android:layout_height="342dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.49"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.611"
        app:srcCompat="@android:drawable/alert_light_frame" />
</androidx.constraintlayout.widget.ConstraintLayout>
```


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
captureimg [C:\Users\shail\AndroidS
captureimg 〉 app 〉 src 〉 main 〉 java 〉 com 〉 example 〉 captureimg 〉 MainActivity
MainActivity.java ×
activity_main.xml ×
AndroidManifest.xml ×
1: Project
Resource Manager
7: Structure
Layout Captures
Build Variants
vorites

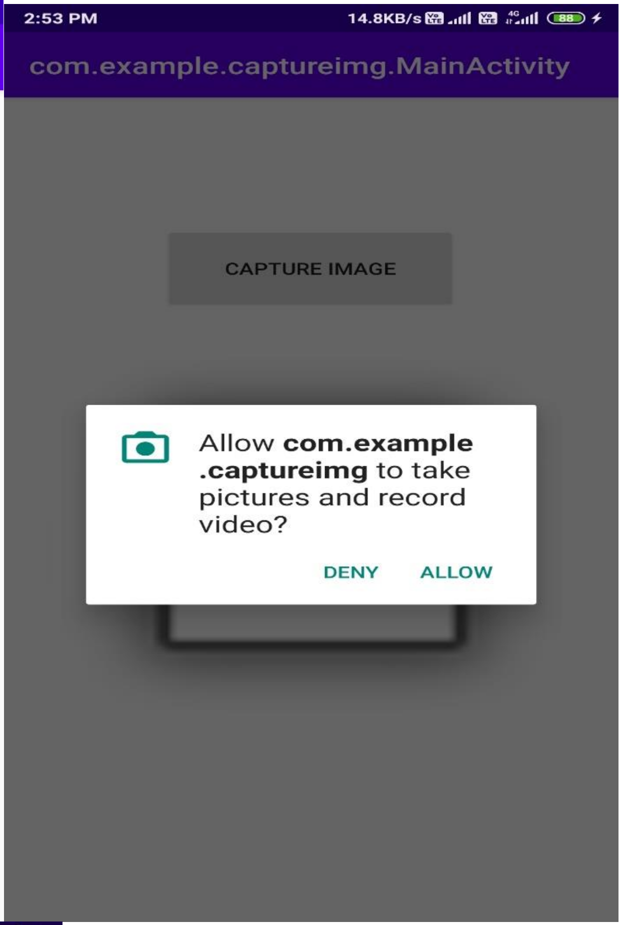
```java
package com.example.captureimg;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    public static final int REQUEST_IMAGE_CAPTURE = 1;
    ImageView ivm;
    Button ok;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ivm = (ImageView) findViewById(R.id.iv);
        ok = (Button) findViewById(R.id.button);
        EnableRuntimePermission();
        ok.setOnClickListener((v) → {
                Intent takePictureIntent = new Intent(android.provider.MediaStore.
                        ACTION_IMAGE_CAPTURE);
                startActivityForResult(takePictureIntent, REQUEST_IMAGE_CAPTURE);
        });
    }
```



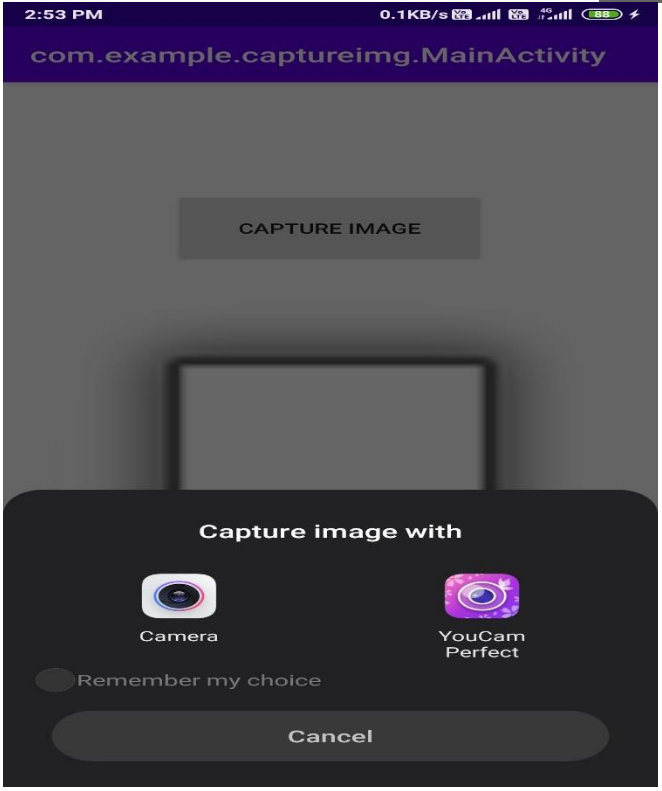
```java
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
        Bitmap imageBitmap = (Bitmap) data.getExtras().get("data");
        ivm.setImageBitmap(imageBitmap);
    }
}

public void EnableRuntimePermission() {
    if (ActivityCompat.shouldShowRequestPermissionRationale( activity: MainActivity.this,
            Manifest.permission.CAMERA)) {
        Toast.makeText( context: MainActivity.this,  text: "CAMERA permission allows us to Access CAMERA app",
                Toast.LENGTH_LONG).show();
    } else {
        ActivityCompat.requestPermissions( activity: MainActivity.this, new String[]{Manifest.permission.CAMERA},
                REQUEST_IMAGE_CAPTURE);
    }
}

@Override
public void onRequestPermissionsResult(int RC, String per[], int[] PResult) {
    switch (RC) {
        case REQUEST_IMAGE_CAPTURE:
            if (PResult.length > 0 && PResult[0] == PackageManager.PERMISSION_GRANTED) {
                Toast.makeText( context: MainActivity.this, text: "Permission Granted, Now your application can access CAMERA.",
                        Toast.LENGTH_LONG).show();
            } else {
                Toast.makeText( context: MainActivity.this, text: "Permission Canceled, Now your application cannot access CAMERA.",
                        Toast.LENGTH_LONG).show();
            }
            break;  } } }
```

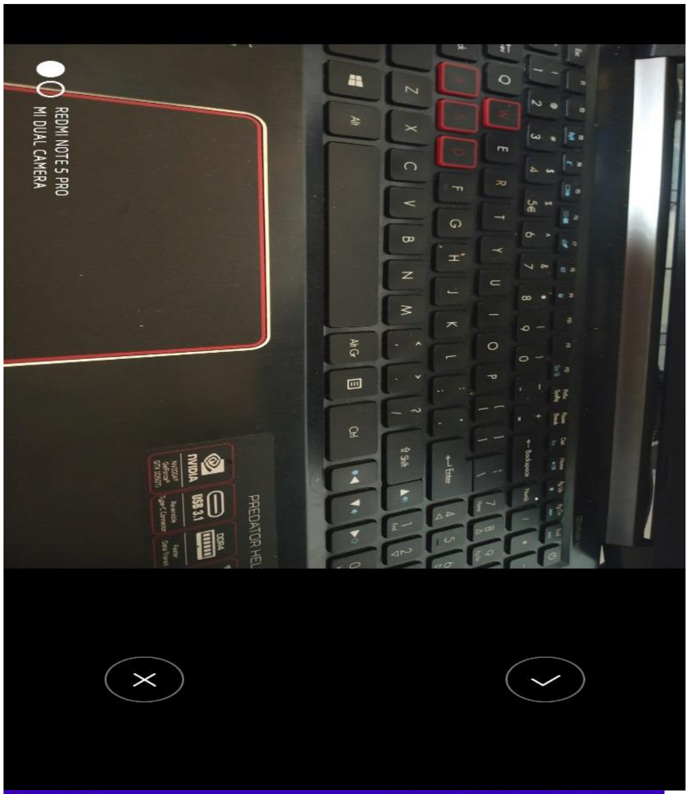


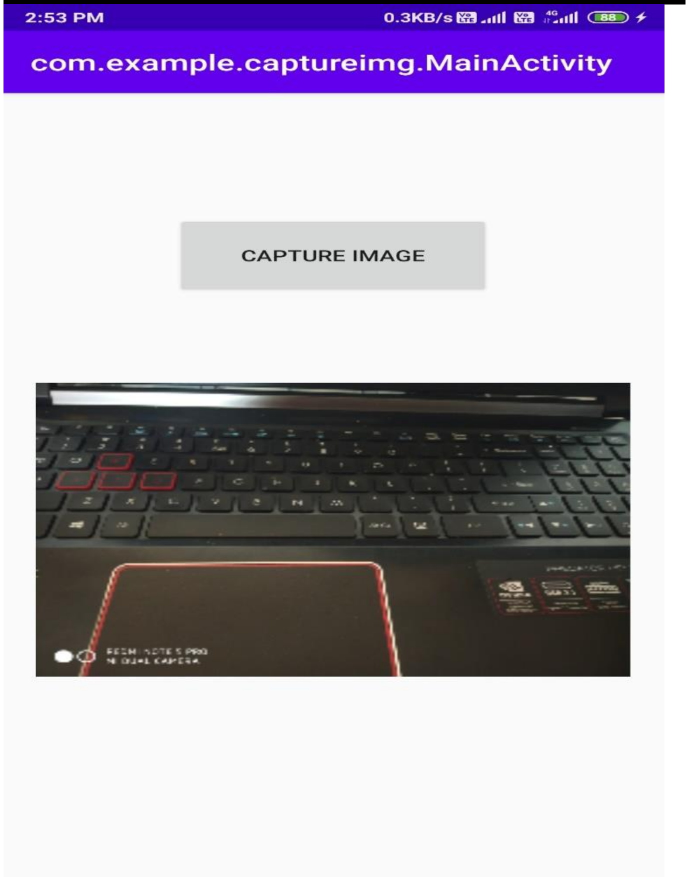
Permission Granted, Now your application can access CAMERA.



Allow **com.example .captureimg** to take pictures and record video?




REDMI NOTE 5 PRO
MI DUAL CAMERA


2:53 PM
0.3KB/s
88
com.example.captureimg.MainActivity
CAPTURE IMAGE
REDMI NOTE 5 PRO
MI DUAL CAMERA