Create a class named **Utility.java** and paste below codes

This code is used to get permission to store in external storage.

**package** in.nic.kerala.training;  
**import** android.Manifest;  
**import** android.annotation.TargetApi;  
**import** android.app.Activity;  
**import** android.content.Context;  
**import** android.content.DialogInterface;  
**import** android.content.pm.PackageManager;  
**import** android.os.Build;  
**import** android.support.v4.app.ActivityCompat;  
**import** android.support.v4.content.ContextCompat;  
**import** android.support.v7.app.AlertDialog;  
  
@SuppressWarnings(**"SpellCheckingInspection"**)  
**public class** Utility {  
 **public static final int *MY\_PERMISSIONS\_REQUEST\_READ\_EXTERNAL\_STORAGE*** = 123;  
  
 @TargetApi(Build.VERSION\_CODES.***JELLY\_BEAN***)  
 **public static boolean** checkPermission(**final** Context context)  
 {  
 **int** currentAPIVersion = Build.VERSION.***SDK\_INT***;  
 **if**(currentAPIVersion>= Build.VERSION\_CODES.***M***)  
 {  
 **if** (ContextCompat.*checkSelfPermission*(context, Manifest.permission.***READ\_EXTERNAL\_STORAGE***) != PackageManager.***PERMISSION\_GRANTED***) {  
 **if** (ActivityCompat.*shouldShowRequestPermissionRationale*((Activity) context, Manifest.permission.***READ\_EXTERNAL\_STORAGE***)) {  
 AlertDialog.Builder alertBuilder = **new** AlertDialog.Builder(context);  
 alertBuilder.setCancelable(**true**);  
 alertBuilder.setTitle(**"Permission necessary"**);  
 alertBuilder.setMessage(**"External storage permission is necessary"**);  
 alertBuilder.setPositiveButton(android.R.string.***yes***, **new** DialogInterface.OnClickListener() {  
 @TargetApi(Build.VERSION\_CODES.***JELLY\_BEAN***)  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 ActivityCompat.*requestPermissions*((Activity) context, **new** String[]{Manifest.permission.***READ\_EXTERNAL\_STORAGE***}, ***MY\_PERMISSIONS\_REQUEST\_READ\_EXTERNAL\_STORAGE***);  
 }  
 });  
 AlertDialog alert = alertBuilder.create();  
 alert.show();  
  
 } **else** {  
 ActivityCompat.*requestPermissions*((Activity) context, **new** String[]{Manifest.permission.***READ\_EXTERNAL\_STORAGE***}, ***MY\_PERMISSIONS\_REQUEST\_READ\_EXTERNAL\_STORAGE***);  
 }  
 **return false**;  
 } **else** {  
 **return true**;  
 }  
 } **else** {  
 **return true**;  
 }  
 }  
}

Create a new Activity named **camera** and place below code in **camera.java**

**package** in.nic.kerala.training;  
  
**import** android.Manifest;  
**import** android.app.Activity;  
**import** android.app.AlertDialog;  
**import** android.content.ContentValues;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.content.pm.PackageManager;  
**import** android.graphics.Bitmap;  
**import** android.graphics.BitmapFactory;  
**import** android.graphics.drawable.BitmapDrawable;  
**import** android.graphics.drawable.Drawable;  
**import** android.net.Uri;  
**import** android.os.Build;  
**import** android.os.Environment;  
**import** android.provider.MediaStore;  
**import** android.support.v4.app.ActivityCompat;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.ImageView;  
**import** android.widget.Toast;  
  
**import** java.io.BufferedInputStream;  
**import** java.io.ByteArrayOutputStream;  
**import** java.io.File;  
**import** java.io.FileInputStream;  
**import** java.io.FileNotFoundException;  
**import** java.io.FileOutputStream;  
**import** java.io.IOException;  
**import** java.util.Random;  
  
**public class** camera **extends** AppCompatActivity {  
 **private** ImageView **ivImage**;  
 **public static final int *MY\_PERMISSIONS\_REQUEST\_READ\_EXTERNAL\_STORAGE*** = 123;  
 Drawable **drawable**;  
 **final int MY\_PERMISSIONS\_REQUEST\_WRITE\_EXTERNAL\_STORAGE** = 1;  
 **private** String **userChoosenTask**;  
 String **ImagePath**;  
 Uri **URI**;  
 **private int REQUEST\_CAMERA** = 0, **SELECT\_FILE** = 1;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_camera***);  
  
 **ivImage** = (ImageView) findViewById(R.id.***imageView2***);  
  
 **ivImage**.setImageResource(R.drawable.***image***);  
 }  
  
  
  
  
  
  
 **private void** cameraIntent()  
 {  
 Intent intent = **new** Intent(MediaStore.***ACTION\_IMAGE\_CAPTURE***);  
 startActivityForResult(intent, **REQUEST\_CAMERA**);  
 }  
 **private void** galleryIntent()  
 {  
 Intent intent = **new** Intent();  
 intent.setType(**"image/\*"**);  
 intent.setAction(Intent.***ACTION\_GET\_CONTENT***);*//* startActivityForResult(Intent.*createChooser*(intent, **"Select File"**),**SELECT\_FILE**);  
 }  
  
  
 **public void** selectImage(View view) {  
  
  
 **final** CharSequence[] items = { **"Take Photo"**, **"Choose from Library"**,  
 **"Cancel"** };  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(camera.**this**);  
 builder.setTitle(**"Add Photo!"**);  
 builder.setItems(items, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** item) {  
 **boolean** result=Utility.*checkPermission*(camera.**this**);  
 **if** (items[item].equals(**"Take Photo"**)) {  
 **userChoosenTask**=**"Take Photo"**;  
 **if**(result)  
 cameraIntent();  
 } **else if** (items[item].equals(**"Choose from Library"**)) {  
 **userChoosenTask**=**"Choose from Library"**;  
 **if**(result)  
 galleryIntent();  
 } **else if** (items[item].equals(**"Cancel"**)) {  
 dialog.dismiss();  
 }  
 }  
 });  
 builder.show();  
  
 }  
  
 @Override  
 **public void** onRequestPermissionsResult(**int** requestCode, String[] permissions, **int**[] grantResults) {  
 **switch** (requestCode) {  
 **case** Utility.***MY\_PERMISSIONS\_REQUEST\_READ\_EXTERNAL\_STORAGE***:  
 **if** (grantResults.**length** > 0 && grantResults[0] == PackageManager.***PERMISSION\_GRANTED***) {  
 **if**(**userChoosenTask**.equals(**"Take Photo"**))  
 cameraIntent();  
 **else if**(**userChoosenTask**.equals(**"Choose from Library"**))  
 galleryIntent();  
 } **else** {  
 *//code for deny* }  
 **break**;  
 }  
 }  
  
 @Override  
 **public void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  
 **super**.onActivityResult(requestCode, resultCode, data);  
 **if** (resultCode == Activity.***RESULT\_OK***) {  
 **if** (requestCode == **SELECT\_FILE**)  
 onSelectFromGalleryResult(data);  
 **else if** (requestCode == **REQUEST\_CAMERA**)  
 onCaptureImageResult(data);  
 }  
 }  
  
 **private void** onSelectFromGalleryResult(Intent data) {  
 Bitmap bm=**null**;  
 Bitmap thumbnail =**null**;  
 **if** (data != **null**) {  
 **try** {  
 bm = MediaStore.Images.Media.*getBitmap*(getApplicationContext().getContentResolver(), data.getData());  
 thumbnail = **this**.CompressResizeImage(bm);  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 **ivImage**.setImageBitmap(thumbnail);  
 **ivImage**.setTag(thumbnail);  
  
  
 }  
  
 **private void** onCaptureImageResult(Intent data) {  
 Bitmap thumbnail = (Bitmap) data.getExtras().get(**"data"**);  
 ByteArrayOutputStream bytes = **new** ByteArrayOutputStream();  
 thumbnail.compress(Bitmap.CompressFormat.***JPEG***, 90, bytes);  
 File destination = **new** File(Environment.*getExternalStorageDirectory*(),  
 System.*currentTimeMillis*() + **".jpg"**);  
 FileOutputStream fo;  
 **try** {  
 destination.createNewFile();  
 fo = **new** FileOutputStream(destination);  
 fo.write(bytes.toByteArray());  
 fo.close();  
 } **catch** (FileNotFoundException e) {  
 e.printStackTrace();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 **ivImage**.setImageBitmap(thumbnail);  
  
 **ivImage**.setTag(thumbnail);  
  
  
 }  
  
 **public static byte**[] ConvertDrawableToByteArray(Drawable drawableResource) {  
  
 Bitmap imageBitmap = ((BitmapDrawable) drawableResource).getBitmap();  
 ByteArrayOutputStream imageByteStream = **new** ByteArrayOutputStream();  
 imageBitmap.compress(Bitmap.CompressFormat.***PNG***, 100, imageByteStream);  
 **byte**[] imageByteData = imageByteStream.toByteArray();  
 **return** imageByteData;  
 }  
  
 **public** Bitmap CompressResizeImage(Bitmap bm)  
 {  
 **int** bmWidth = bm.getWidth();  
 **int** bmHeight = bm.getHeight();  
 **int** ivWidth = **ivImage**.getWidth();  
 **int** ivHeight = **ivImage**.getHeight();  
  
  
 **int** new\_height = (**int**) Math.*floor*((**double**) bmHeight \*( (**double**) ivWidth / (**double**) bmWidth));  
 Bitmap newbitMap = Bitmap.*createScaledBitmap*(bm, ivWidth, new\_height, **true**);  
  
 ByteArrayOutputStream baos = **new** ByteArrayOutputStream();  
 newbitMap.compress(Bitmap.CompressFormat.***JPEG***, 100, baos);  
 **byte**[] b = baos.toByteArray();  
  
 Bitmap bm1 = BitmapFactory.*decodeByteArray*(b, 0, b.**length**);  
  
 **return** bm1;  
 }  
  
  
 **public static** String random() {  
 Random generator = **new** Random();  
 StringBuilder randomStringBuilder = **new** StringBuilder();  
 **int** randomLength = generator.nextInt(8);  
 **char** tempChar;  
 **for** (**int** i = 0; i < randomLength; i++){  
 tempChar = (**char**) (generator.nextInt(96) + 32);  
 randomStringBuilder.append(tempChar);  
 }  
 **return** randomStringBuilder.toString();  
 }  
  
  
 *//BUTTON CLICK* **public void** saveImage(View view) {  
  
 **if** (ActivityCompat.*checkSelfPermission*(camera.**this**, Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***) != PackageManager.***PERMISSION\_GRANTED***) {  
  
 ActivityCompat.*requestPermissions*(camera.**this**,  
 **new** String[]{Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***},  
 **MY\_PERMISSIONS\_REQUEST\_WRITE\_EXTERNAL\_STORAGE**);  
  
 }  
  
 Bitmap bmImg = ((BitmapDrawable)**ivImage**.getDrawable()).getBitmap();  
  
  
 File filename;  
 **try** {  
 String path1 = android.os.Environment.*getExternalStorageDirectory*()  
 .toString();  
 Log.*i*(**"in save()"**, **"after mkdir"**);  
 String dirctioary = *random*();  
 File file = **new** File(path1 + **"/"** + dirctioary);  
  
  
 **if** (!file.exists())  
 file.mkdirs();  
 filename = **new** File(file.getAbsolutePath() + **"/"** + **"tests"** + **".jpg"**);  
 Log.*i*(**"in save()"**, **"after file"**);  
 FileOutputStream out = **new** FileOutputStream(filename);  
 Log.*i*(**"in save()"**, **"after outputstream"**);  
 bmImg.compress(Bitmap.CompressFormat.***JPEG***, 90, out);  
 out.flush();  
 out.close();  
 Log.*i*(**"in save()"**, **"after outputstream closed"**);  
 *//File parent = filename.getParentFile();* ContentValues image = getImageContent(filename);  
 Uri result = getContentResolver().insert(  
 MediaStore.Images.Media.***EXTERNAL\_CONTENT\_URI***, image);  
 Toast.*makeText*(getApplicationContext(),  
 **"File is Saved in "** + filename, Toast.***LENGTH\_SHORT***).show();  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 Toast.*makeText*(camera.**this**, **"Image Saved Successfully"**, Toast.***LENGTH\_LONG***).show();  
  
 }  
  
 **public** ContentValues getImageContent(File parent) {  
 ContentValues image = **new** ContentValues();  
 image.put(MediaStore.Images.Media.***TITLE***, **"TRAINING"**);  
 image.put(MediaStore.Images.Media.***DISPLAY\_NAME***, **"tests"**);  
 image.put(MediaStore.Images.Media.***DESCRIPTION***, **"App Image"**);  
 image.put(MediaStore.Images.Media.***DATE\_ADDED***, System.*currentTimeMillis*());  
 image.put(MediaStore.Images.Media.***MIME\_TYPE***, **"image/jpg"**);  
 image.put(MediaStore.Images.Media.***ORIENTATION***, 0);  
 image.put(MediaStore.Images.ImageColumns.***BUCKET\_ID***, parent.toString()  
 .toLowerCase().hashCode());  
 image.put(MediaStore.Images.ImageColumns.***BUCKET\_DISPLAY\_NAME***, parent.getName()  
 .toLowerCase());  
 image.put(MediaStore.Images.Media.***SIZE***, parent.length());  
 image.put(MediaStore.Images.Media.***DATA***, parent.getAbsolutePath());  
 **return** image;  
 }  
 **public void** RequestRunTimePermission(){  
  
 **if** (ActivityCompat.*shouldShowRequestPermissionRationale*(camera.**this**, Manifest.permission.***READ\_EXTERNAL\_STORAGE***))  
 {  
  
 Toast.*makeText*(camera.**this**,**"READ\_EXTERNAL\_STORAGE permission Access Dialog"**, Toast.***LENGTH\_LONG***).show();  
  
 } **else** {  
  
 ActivityCompat.*requestPermissions*(camera.**this**,**new** String[]{ Manifest.permission.***READ\_EXTERNAL\_STORAGE***}, 1);  
  
 }  
 }  
  
  
 **public boolean** isStoragePermissionGranted() {  
 **if** (Build.VERSION.***SDK\_INT*** >= 23) {  
 **if** (checkSelfPermission(android.Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***)  
 == PackageManager.***PERMISSION\_GRANTED***) {  
  
  
 **return true**;  
 } **else** {  
  
  
 ActivityCompat.*requestPermissions*(**this**, **new** String[]{Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***}, 1);  
 **return false**;  
 }  
 }  
 **else** { *//permission is automatically granted on sdk<23 upon installation* **return true**;  
 }  
 }  
  
  
}

**Add below Layout to activity\_camera**

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.v7.widget.LinearLayoutCompat 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:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".camera"**>  
  
  
  
 <**LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_marginTop="100dp"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginRight="10dp"** >  
  
 <**ImageView  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:src="@drawable/image"  
 android:id="@+id/imageView2"** />  
  
 <**LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:layout\_gravity="center\_vertical|center\_horizontal"  
 android:layout\_marginLeft="20dp"** >  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Please Capture Your Favourite Place"  
 android:textSize="16sp"  
 android:textColor="@android:color/black"** />  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Choose Image"  
 android:textSize="16sp"  
 android:background="@drawable/cellbg"  
 android:textColor="@android:color/white"  
 android:padding="4dp"  
 android:layout\_marginTop="5dp"  
 android:textAlignment="center"  
 android:onClick="selectImage"** />  
  
  
 </**LinearLayout**>  
  
  
  
 </**LinearLayout**>  
  
 <**Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="10dp"  
 android:onClick="saveImage"  
 android:text="Save Your Image"  
 android:textColor="@android:color/black"** />  
  
  
</**android.support.v7.widget.LinearLayoutCompat**>