# Program 4:

# a)Upload Image using Camera - Get Image from Camera | Android Studio Tutorial

# activity\_main.xml

# <?xml version="1.0" encoding="utf-8"?> <LinearLayout 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:gravity="center\_horizontal" android:orientation="vertical" tools:context=".MainActivity"> <ImageView android:id="@+id/imgCamera" android:layout\_width="400dp" android:layout\_height="240dp" android:scaleType="fitXY" /> <Button android:id="@+id/btnCamera" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginTop="21dp" android:text="Open Camera"/> </LinearLayout>

# MainActivity.java

# package com.example.camerasd; import androidx.annotation.Nullable; import androidx.appcompat.app.AppCompatActivity; import android.content.Intent; import android.graphics.Bitmap; import android.os.Bundle; import android.provider.MediaStore; import android.view.View; import android.widget.Button; import android.widget.ImageView; public class MainActivity extends AppCompatActivity { private final int CAMERA\_REQ\_CODE = 100; ImageView imgCamera; @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.*activity\_main*); imgCamera = findViewById(R.id.*imgCamera*); Button btnCamera = findViewById(R.id.*btnCamera*); btnCamera.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View view) { Intent iCamera = new Intent(MediaStore.*ACTION\_IMAGE\_CAPTURE*); startActivityForResult(iCamera, CAMERA\_REQ\_CODE); } }); } @Override protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) { super.onActivityResult(requestCode, resultCode, data); if (resultCode == *RESULT\_OK*) { if (requestCode == CAMERA\_REQ\_CODE) { //for camera Bitmap img = (Bitmap) (data.getExtras().get("data")); imgCamera.setImageBitmap(img); } } } }

# 4b) Android select an image from the gallery or storage

# AndriodMainfest.xml

# Note: Only Boldface code should add to be added to AndriodMainfest.xml

# <?xml version="1.0" encoding="utf-8"?> <manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.example.galleryapp"> <!--//Adding Read External storage Permission--> <uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"/> <application android:allowBackup="true" android:icon="@mipmap/ic\_launcher" android:label="Select Image from Gallery" android:roundIcon="@mipmap/ic\_launcher\_round" android:supportsRtl="true" android:theme="@style/Theme.GalleryApp"> <activity android:name=".MainActivity" android:exported="true"> <intent-filter> <action android:name="android.intent.action.MAIN" /> <category android:name="android.intent.category.LAUNCHER" /> </intent-filter> </activity> </application> </manifest>

# MainActivity

# package com.example.galleryapp; import androidx.annotation.NonNull; import androidx.annotation.Nullable; import androidx.appcompat.app.AppCompatActivity; import androidx.core.app.ActivityCompat; import androidx.core.content.ContextCompat; import android.Manifest; import android.app.Activity; import android.content.Intent; import android.content.pm.PackageManager; import android.graphics.Bitmap; import android.graphics.BitmapFactory; import android.net.Uri; import android.os.Build; import android.os.Bundle; import android.provider.MediaStore; import android.view.View; import android.widget.Button; import android.widget.ImageView; import android.widget.Toast; import java.io.InputStream; public class MainActivity extends AppCompatActivity { private static final int *REQUEST\_CODE\_STORAGE\_PERMISSION* = 1; private static final int *REQUEST\_CODE\_SELECT\_IMAGE* = 2; private ImageView imageSelected; @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.*activity\_main*); imageSelected = findViewById(R.id.*selectedImage*); findViewById(R.id.*buttonSelectedImage*).setOnClickListener(new View.OnClickListener() { @Override public void onClick(View view) { if (ContextCompat.*checkSelfPermission*( getApplicationContext(), Manifest.permission.*READ\_EXTERNAL\_STORAGE* ) != PackageManager.*PERMISSION\_GRANTED*) { ActivityCompat.*requestPermissions*( MainActivity.this, new String[]{Manifest.permission.*READ\_EXTERNAL\_STORAGE*}, *REQUEST\_CODE\_STORAGE\_PERMISSION*); } else { selectImage(); } } }); } private void selectImage() { Intent intent = new Intent(Intent.*ACTION\_PICK*,MediaStore.Images.Media.*EXTERNAL\_CONTENT\_URI*); if(intent.resolveActivity(getPackageManager()) != null){ startActivityForResult(intent,*REQUEST\_CODE\_SELECT\_IMAGE*); } } @Override public void onRequestPermissionsResult(int requestCode,@Nullable String[] permissions,@Nullable int[] grantResults) { super.onRequestPermissionsResult(requestCode,permissions,grantResults); if(requestCode == *REQUEST\_CODE\_STORAGE\_PERMISSION* && grantResults.length > 0) { if(grantResults[0] == PackageManager.*PERMISSION\_GRANTED*) { selectImage(); } else { Toast.*makeText*(this, "Permission Denied", Toast.*LENGTH\_SHORT*).show(); } } } @Override protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) { super.onActivityResult(requestCode, resultCode, data); if(requestCode == *REQUEST\_CODE\_SELECT\_IMAGE* && resultCode == *RESULT\_OK*){ if(data != null){ Uri selectedImageUri = data.getData(); if(selectedImageUri != null){ try{ InputStream inputStream = getContentResolver().openInputStream(selectedImageUri); Bitmap bitmap = BitmapFactory.*decodeStream*(inputStream); imageSelected.setImageBitmap(bitmap); }catch (Exception exception){ Toast.*makeText*(this,exception.getMessage(), Toast.*LENGTH\_SHORT*).show(); } } } } } }

# activitymain.xml

# <?xml version="1.0" encoding="utf-8"?> <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" xmlns:tools="http://schemas.android.com/tools" android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:orientation="vertical" tools:context=".MainActivity"> <Button android:id="@+id/buttonSelectedImage" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:text="@string/select\_image"/> <ImageView android:id="@+id/selectedImage" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:adjustViewBounds="true" android:contentDescription="select\_image" /> </LinearLayout>