1.Xml file:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<Button

android:id="@+id/capture\_button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Capture Image"

android:layout\_marginLeft="130dp"

android:layout\_marginTop="10dp"/>

<ImageView

android:id="@+id/image\_view"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20dp"

android:scaleType="centerCrop"/>

</LinearLayout>

2.Java file:

package com.example.camera;

import android.app.Activity;

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 Activity {

static final int REQUEST\_IMAGE\_CAPTURE = 1;

ImageView imageView;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Button captureButton = findViewById(R.id.capture\_button);

imageView = findViewById(R.id.image\_view);

captureButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

dispatchTakePictureIntent();

}

});

}

private void dispatchTakePictureIntent() {

Intent takePictureIntent = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);

if (takePictureIntent.resolveActivity(getPackageManager()) != null) {

startActivityForResult(takePictureIntent, REQUEST\_IMAGE\_CAPTURE);

}

}

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (requestCode == REQUEST\_IMAGE\_CAPTURE && resultCode == RESULT\_OK) {

Bundle extras = data.getExtras();

Bitmap imageBitmap = (Bitmap) extras.get("data");

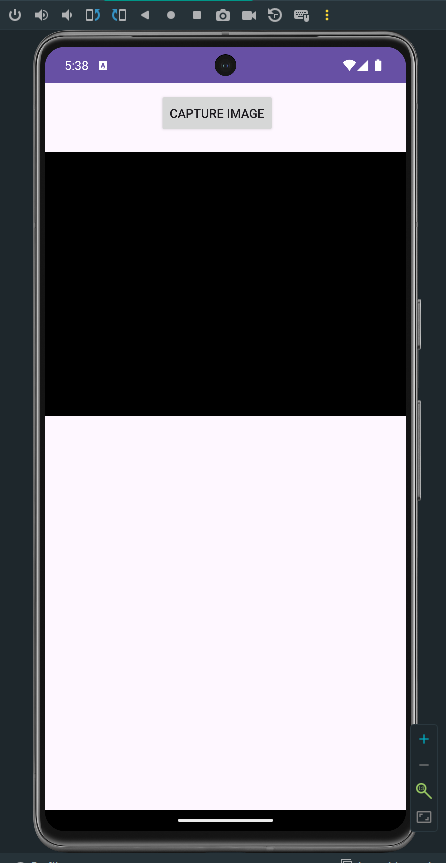
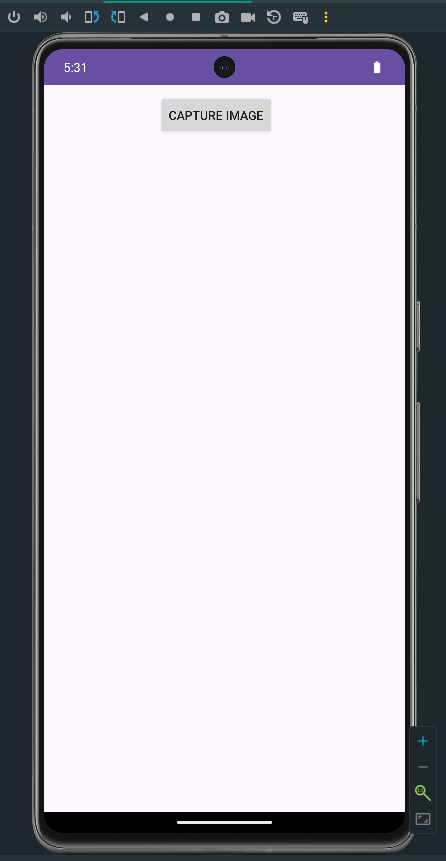
imageView.setImageBitmap(imageBitmap);

}

}

}

Output:



1.Xml file:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<SurfaceView

android:id="@+id/surface\_view"

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_weight="1" />

<Button

android:id="@+id/capture\_button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Capture Video"

android:layout\_marginLeft="130dp"/>

<Button

android:id="@+id/record\_button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Record Video"

android:layout\_marginLeft="135dp"/>

</LinearLayout>

2.Java file:

package com.example.camera2;

import android.app.Activity;

import android.content.Context;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.media.CamcorderProfile;

import android.media.MediaRecorder;

import android.net.Uri;

import android.os.Bundle;

import android.provider.MediaStore;

import android.util.Log;

import android.view.SurfaceHolder;

import android.view.SurfaceView;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

import java.io.IOException;

public class MainActivity extends Activity {

private static final int REQUEST\_VIDEO\_CAPTURE = 1;

private static final String TAG = "VideoRecording";

private SurfaceView mSurfaceView;

private SurfaceHolder mSurfaceHolder;

private MediaRecorder mMediaRecorder;

private boolean isRecording = false;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

if (!getPackageManager().hasSystemFeature(PackageManager.FEATURE\_CAMERA\_ANY)) {

Toast.makeText(this, "No camera on this device", Toast.LENGTH\_LONG).show();

return;

}

mSurfaceView = findViewById(R.id.surface\_view);

mSurfaceHolder = mSurfaceView.getHolder();

Button captureButton = findViewById(R.id.capture\_button);

Button recordButton = findViewById(R.id.record\_button);

captureButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

dispatchTakeVideoIntent();

}

});

recordButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if (isRecording) {

stopRecording();

} else {

startRecording();

}

}

});

mMediaRecorder = new MediaRecorder();

}

private void dispatchTakeVideoIntent() {

Intent takeVideoIntent = new Intent(MediaStore.ACTION\_VIDEO\_CAPTURE);

if (takeVideoIntent.resolveActivity(getPackageManager()) != null) {

startActivityForResult(takeVideoIntent, REQUEST\_VIDEO\_CAPTURE);

}

}

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (requestCode == REQUEST\_VIDEO\_CAPTURE && resultCode == RESULT\_OK) {

Uri videoUri = data.getData();

}

}

private void startRecording() {

try {

mMediaRecorder.setAudioSource(MediaRecorder.AudioSource.DEFAULT);

mMediaRecorder.setVideoSource(MediaRecorder.VideoSource.DEFAULT);

mMediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.MPEG\_4);

mMediaRecorder.setVideoEncoder(MediaRecorder.VideoEncoder.DEFAULT);

mMediaRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.DEFAULT);

mMediaRecorder.setProfile(CamcorderProfile.get(CamcorderProfile.QUALITY\_HIGH));

mMediaRecorder.setOutputFile(getOutputMediaFile().toString());

mMediaRecorder.setPreviewDisplay(mSurfaceHolder.getSurface());

mMediaRecorder.prepare();

mMediaRecorder.start();

isRecording = true;

Log.d(TAG, "Recording started");

} catch (IOException e) {

Log.e(TAG, "Error starting recording", e);

}

}

private void stopRecording() {

mMediaRecorder.stop();

mMediaRecorder.reset();

isRecording = false;

Log.d(TAG, "Recording stopped");

}

private Uri getOutputMediaFile() {

return null;

}

}

Outout:

