**Sanika Kadam**

**2124UCSF1112**

**ASSIGNMENT NO.8**

1. **Introduction**

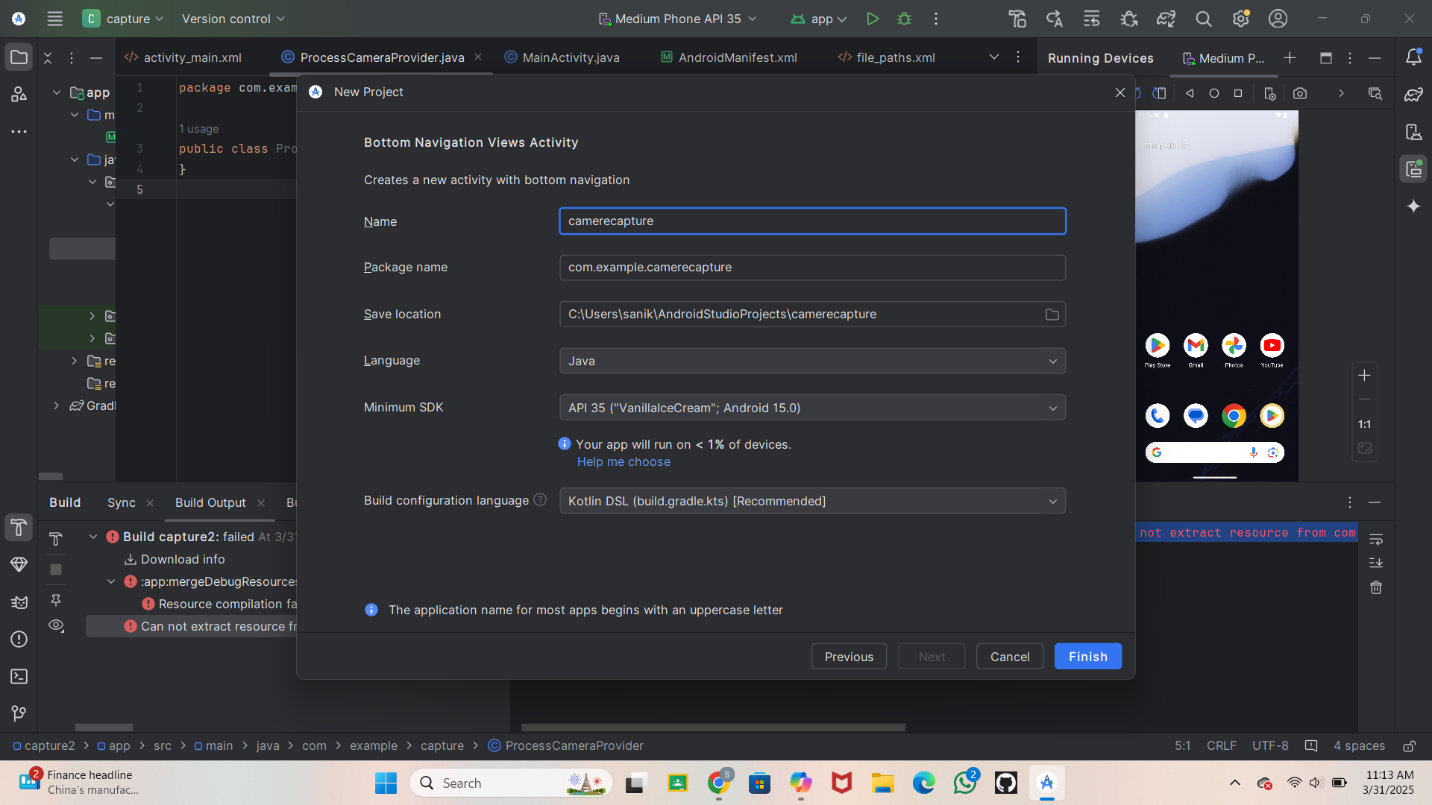
Capture Images Using Camera: Write a program to capture images using the built-in camera. For example, create an app with a "Take Photo" button that opens the camera, captures an image, and displays it on the screen

1. **2. Tools & Technologies Used**

* Android Studio
* Language-Java

**3. Procedure & Steps**

**Step 1: Create a New Project**

* Open Android Studio and create a new project.
* Select second page activity by app-activity-page you selected
* Set the project name(cameracapture) and package name.(com.example.camerecapture)
* Select the programming language (Java).
* **Screenshot**
* 

**Step 2: Designing the UI**

* Open activity\_main.xml and design the layout using XML.
* Add UI components such as TextView, EditText, Button, etc.
* Write Code

<?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"

android:padding="16dp">

<Button

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

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Capture Image" />

<ImageView

android:id="@+id/imageView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="25dp"

android:contentDescription="Captured Image"

android:scaleType="centerInside" />

</LinearLayout

* **Step 3: Write a code in main activity java**

package com.example.camerecapture;

import android.Manifest;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.graphics.Bitmap;

import android.os.Bundle;

import android.provider.MediaStore;

import android.util.Log;

import android.widget.ImageView;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import androidx.core.content.ContextCompat;

public class MainActivity extends AppCompatActivity {

private static final int CAMERA\_REQUEST\_CODE = 100;

private static final int CAMERA\_PERMISSION\_REQUEST\_CODE = 101;

private ImageView imageView;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

imageView = findViewById(R.id.imageView);

// Trigger camera intent when a button is clicked

findViewById(R.id.capture\_button).setOnClickListener(view -> openCamera());

}

// Method to open the camera app

private void openCamera() {

// Check if the app has camera permission

if (ContextCompat.checkSelfPermission(this, Manifest.permission.CAMERA) == PackageManager.PERMISSION\_GRANTED) {

// Check if there is a camera app available to handle the intent

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

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

startActivityForResult(cameraIntent, CAMERA\_REQUEST\_CODE);

} else {

Toast.makeText(this, "Camera not found", Toast.LENGTH\_SHORT).show();

}

} else {

// Request camera permission

ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.CAMERA}, CAMERA\_PERMISSION\_REQUEST\_CODE);

}

}

// Handle the result of the camera permission request

@Override

public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {

super.onRequestPermissionsResult(requestCode, permissions, grantResults);

if (requestCode == CAMERA\_PERMISSION\_REQUEST\_CODE) {

if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION\_GRANTED) {

openCamera(); // Retry opening the camera if permission is granted

} else {

Toast.makeText(this, "Camera permission denied", Toast.LENGTH\_SHORT).show();

}

}

}

// Handle the result from the camera intent

@Override

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

super.onActivityResult(requestCode, resultCode, data);

if (requestCode == CAMERA\_REQUEST\_CODE && resultCode == RESULT\_OK) {

Bitmap photo = (Bitmap) data.getExtras().get("data");

// Display the image in ImageView

imageView.setImageBitmap(photo);

} else {

Toast.makeText(this, "Failed to capture image", Toast.LENGTH\_SHORT).show();

}

}

}

**Step 4: Write a code in Android Manifest**

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

<!-- Permissions for devices targeting Android 10 and above -->

<uses-permission android:name="android.permission.MANAGE\_EXTERNAL\_STORAGE" />

<!-- For Android 11 and above, declare that you want to use the camera feature -->

<uses-feature android:name="android.hardware.camera" android:required="true" />

<uses-feature android:name="android.hardware.camera.autofocus" android:required="false" />

<application

android:requestLegacyExternalStorage="true"

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:theme="@style/Theme.AppCompat.DayNight.oActionBar">

<!-- Main Activity with the exported attribute -->

<activity

android:name=".MainActivity"

android:label="@string/app\_name"

android:theme="@style/Theme.AppCompat.DayNight.NoActionBar"

android:exported="true"> <!-- Explicitly set android:exported -->

<!-- Main launcher activity -->

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

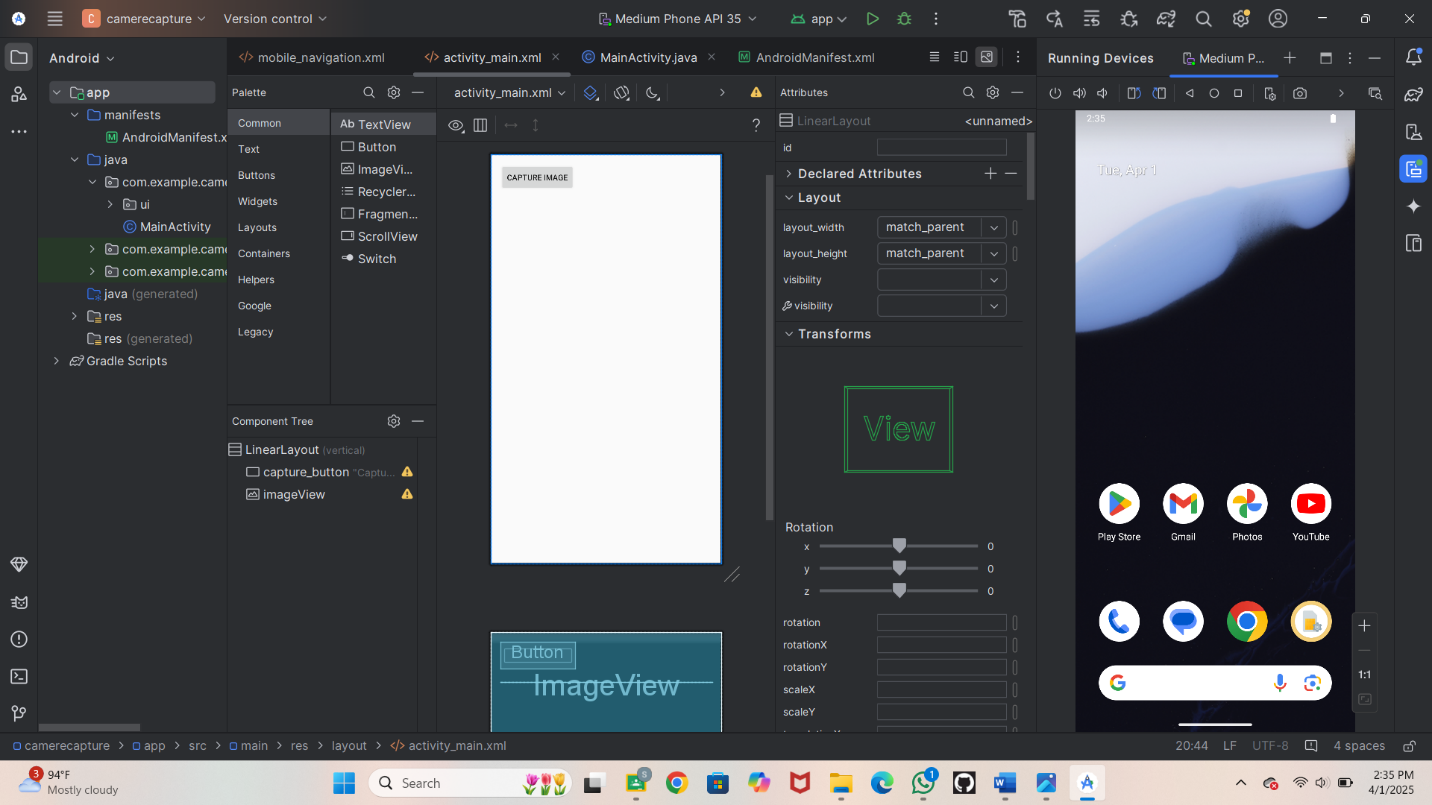
</intent-filter>

</activity>

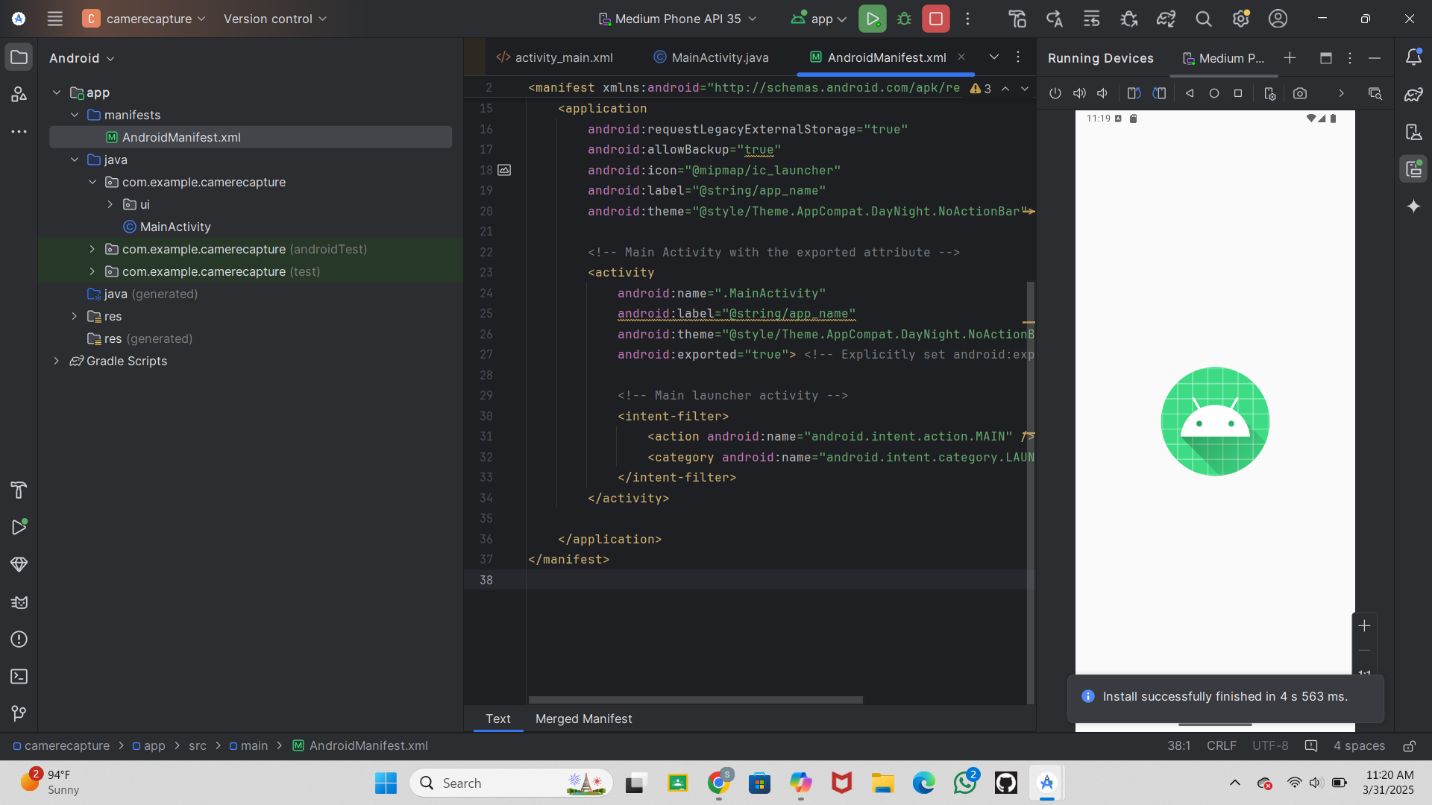
</application>

</manifest>

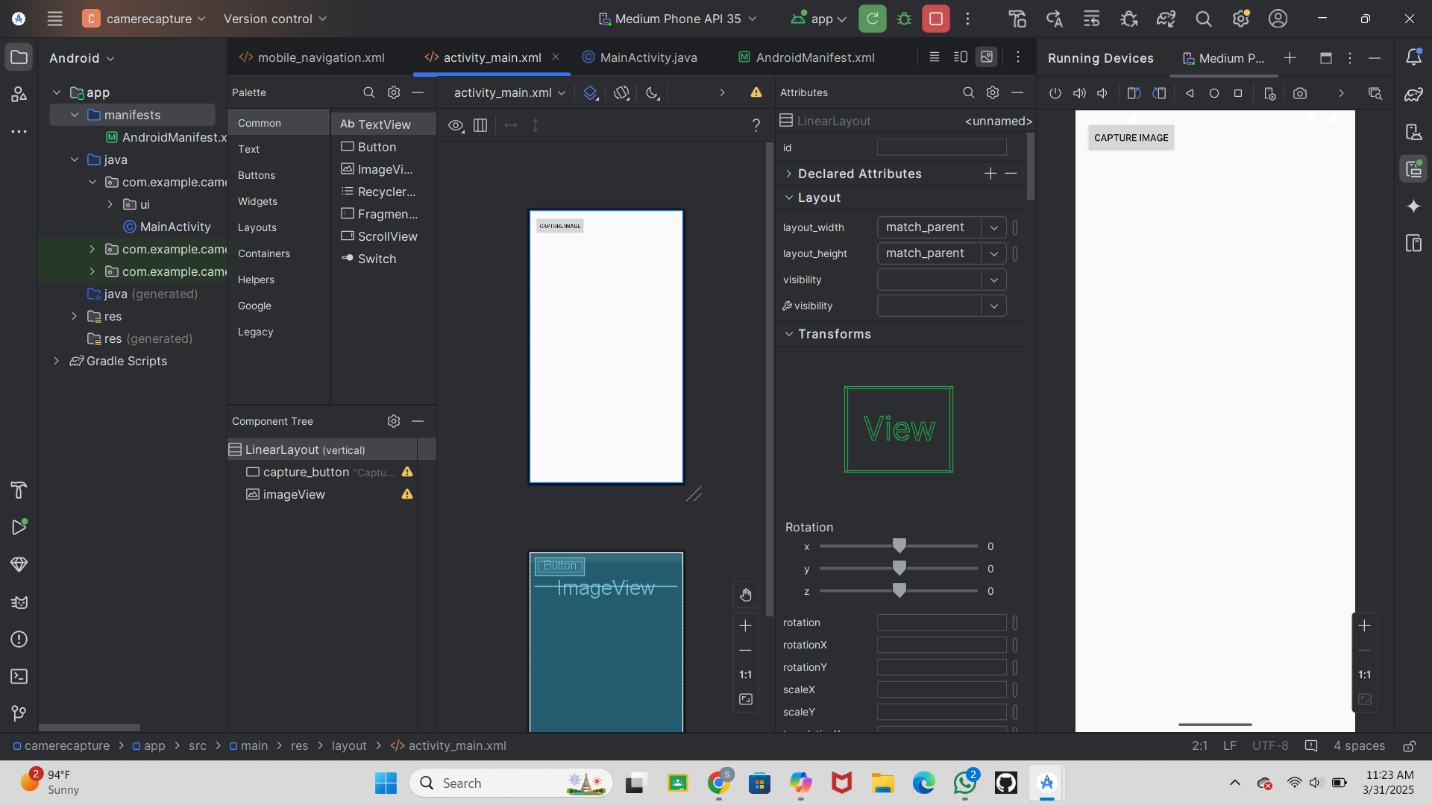
**Screenshot**

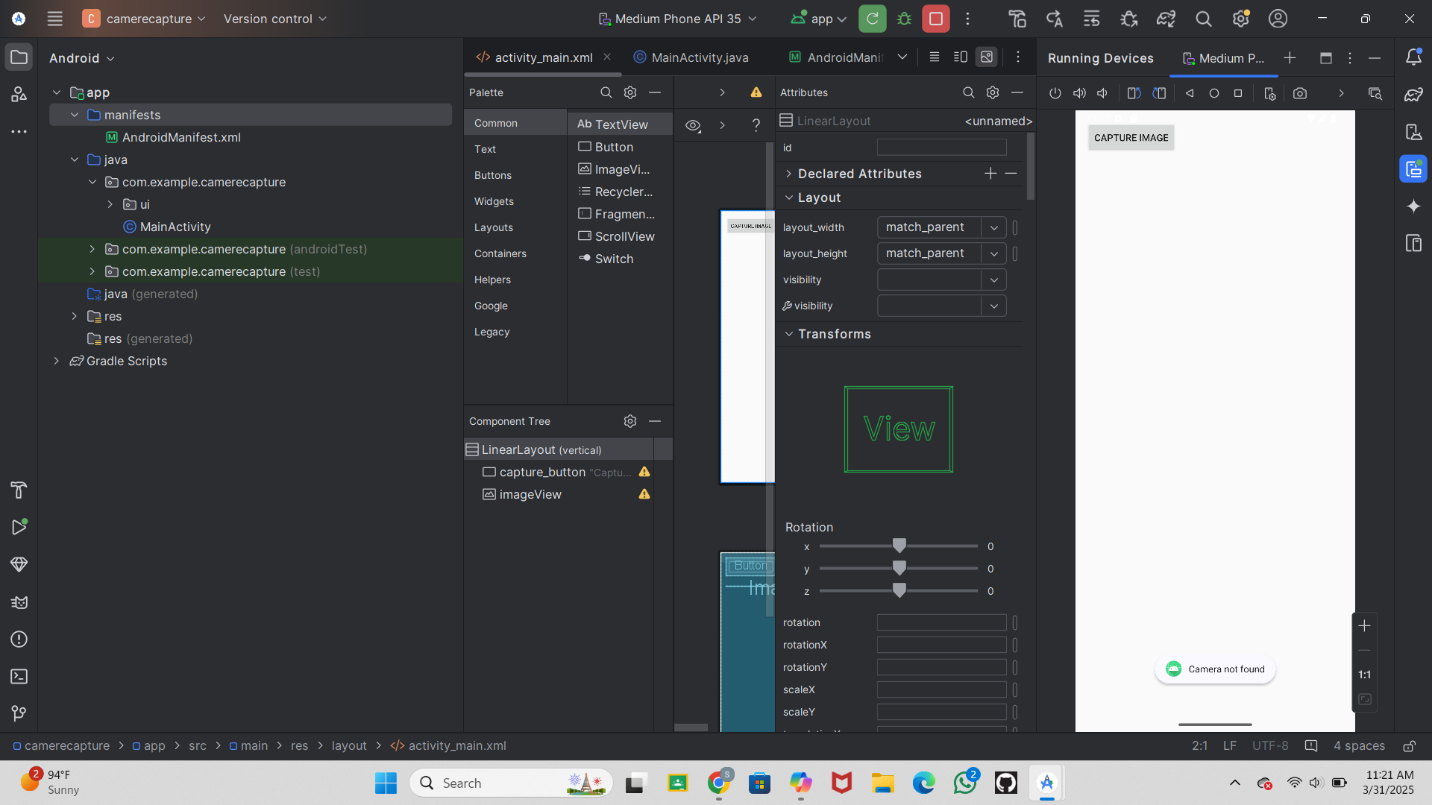


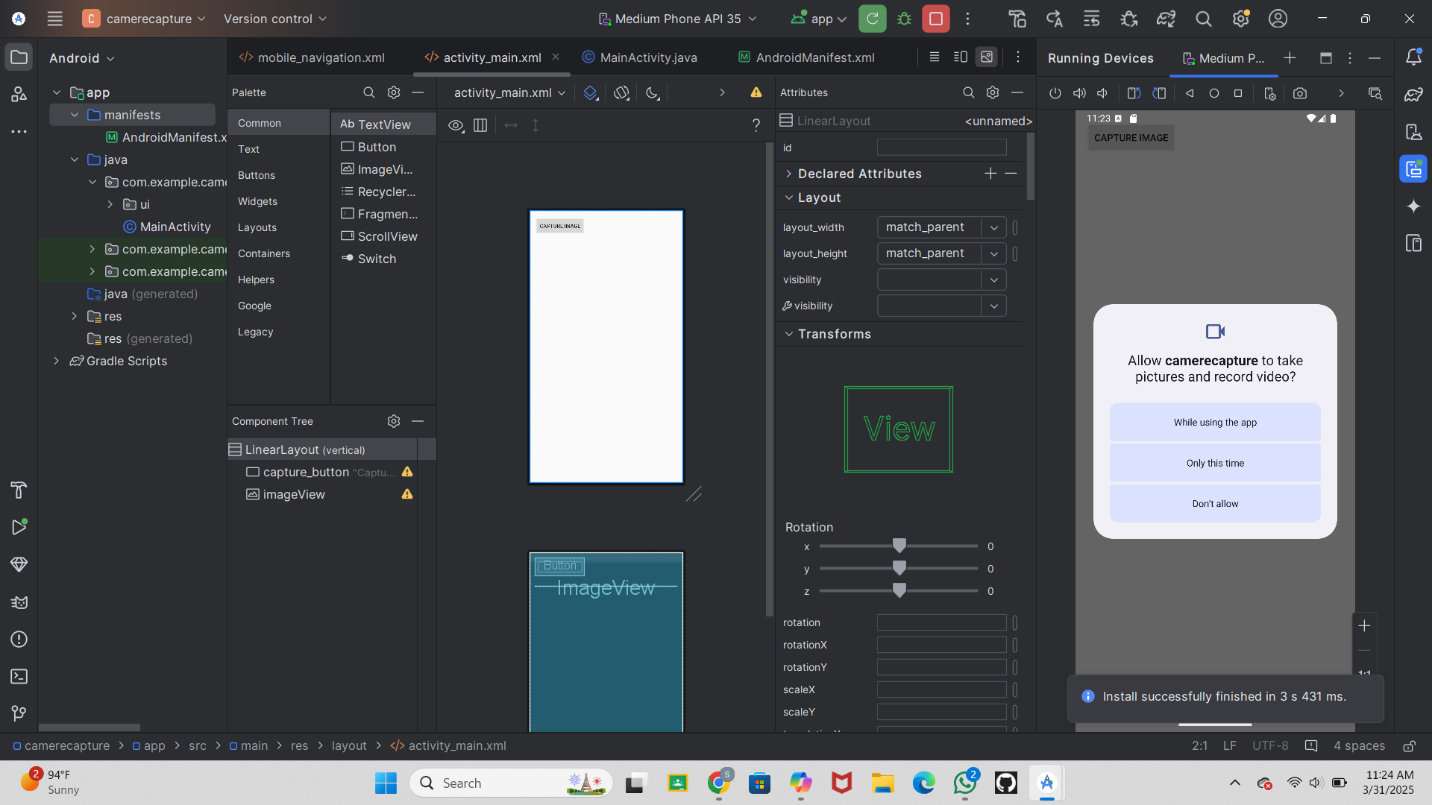
* **Step 5: Running the Application on Emulator**
* Click on the **Run** button in Android Studio.
* **ScreenShot**



Click on Capture image

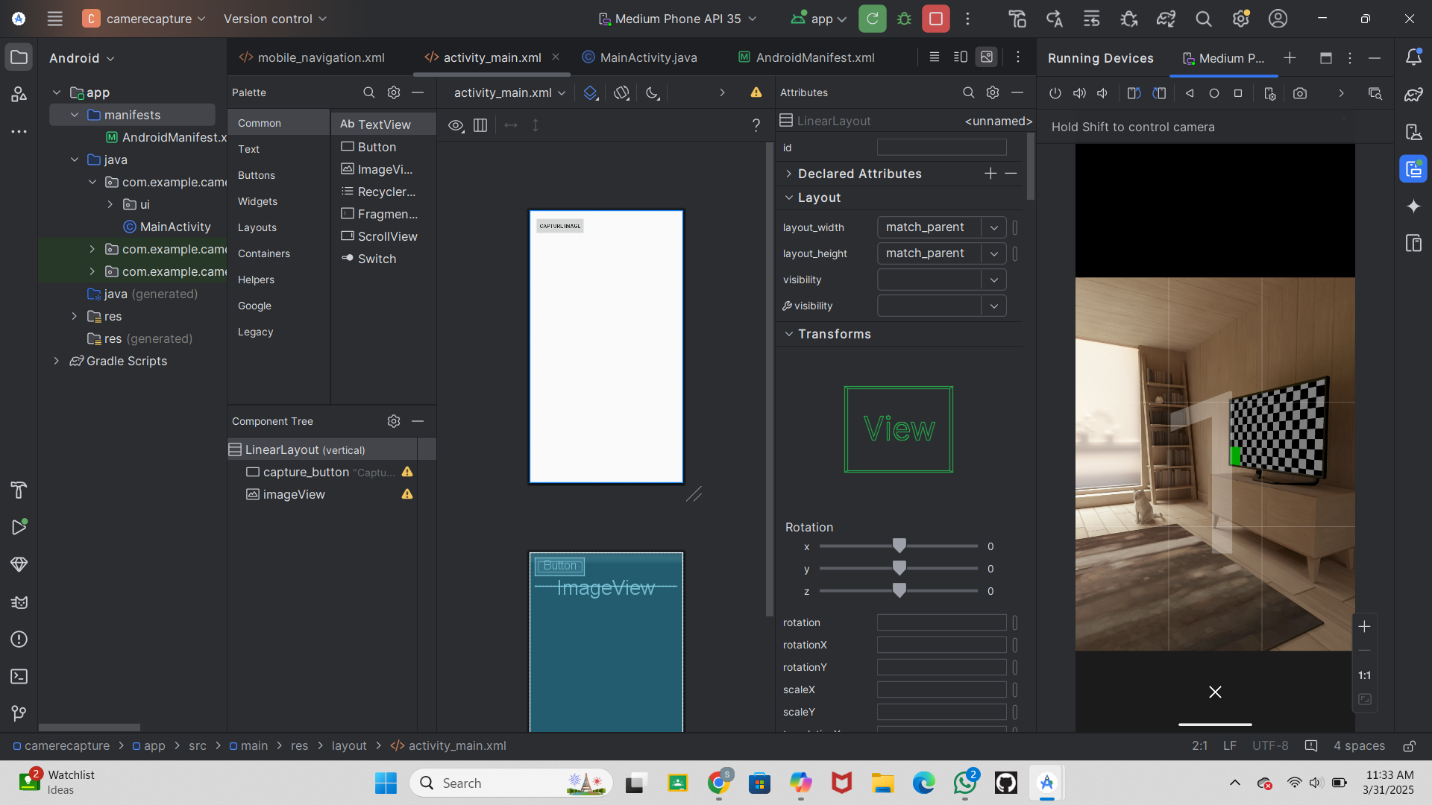






Allow to capture image

Take The photo



**Step 6: Conclusion:**

Capturing images with an Android app involves using the built-in camera to take photos and display them on the screen. This improves the app’s functionality while ensuring smooth performance across different devices. Using CameraX provides better stability and control, making the experience more seamless. Managing permissions and storage efficiently ensures that photos are correctly saved and retrieved. Enhancing the app with features like image filters or gallery integration can further refine its usefulness.