ASSIGNMENT NO.8

GAURAV_BODKHE_2124UCEM1041

1. Introduction

In this program, we will develop a simple Android application that enables users to capture images using their device's built-in camera. The app will feature a user-friendly interface with a "Take Photo" button that opens the camera, captures an image, and displays it on the screen. This project demonstrates how to access and utilize the camera functionality in Android, providing users with a straightforward way to take and view photos directly from their mobile devices.

2. Tools & Technologies Used

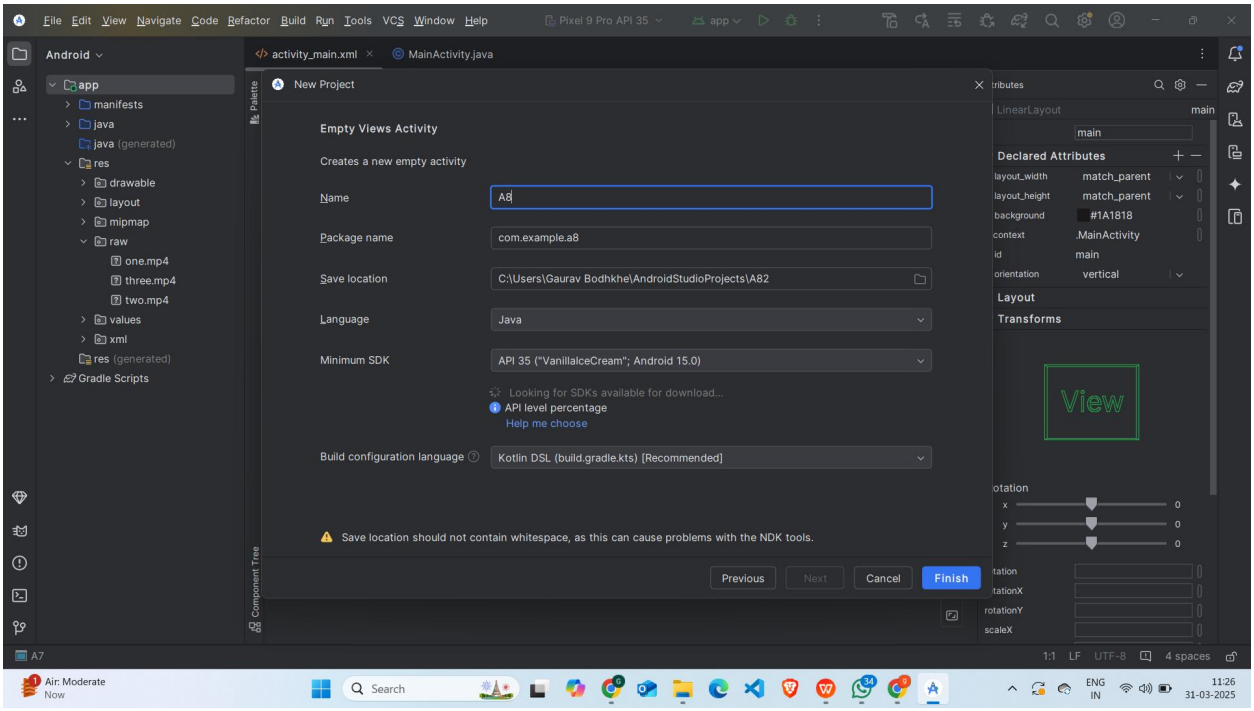
- Android Studio
- Java
- Emulator

3. Procedure & Steps

Step 1: Create a New Project

- Open Android Studio and create a new project.
- Choose an Empty Views Activity template.
- Set the project name and package name of your Application
- Select the programming language (Java).

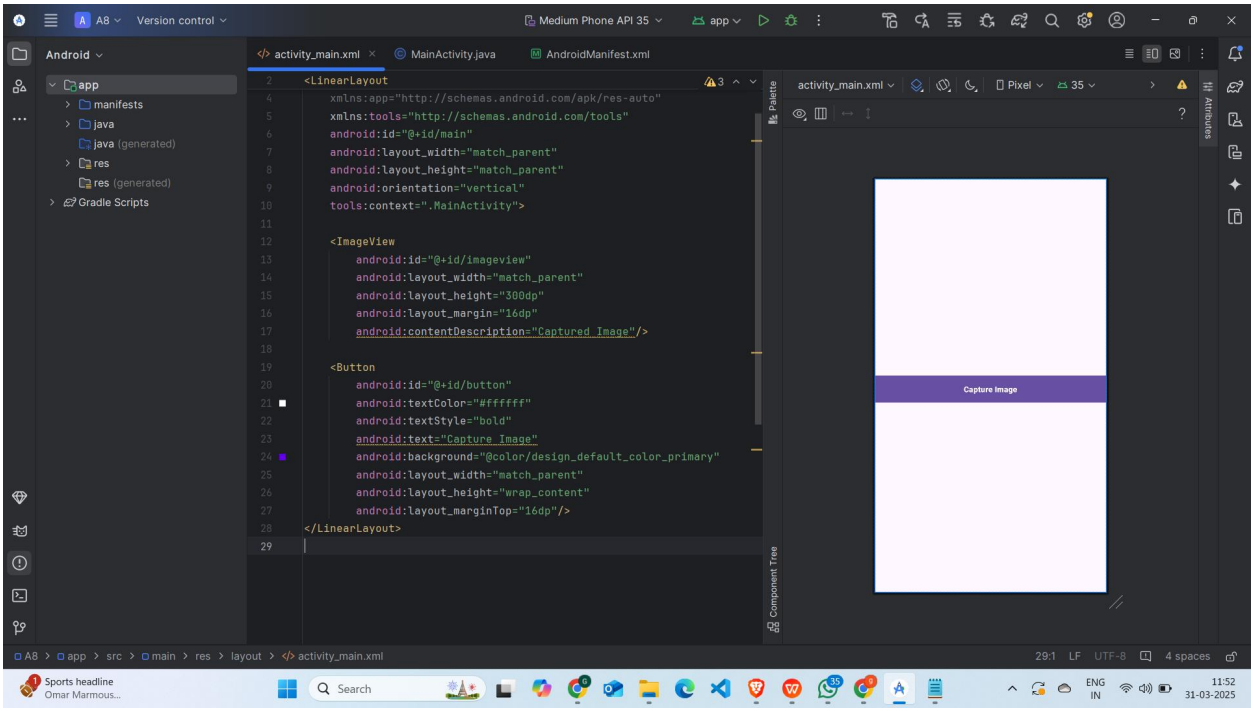
Screenshot:



Step 2: Designing the UI

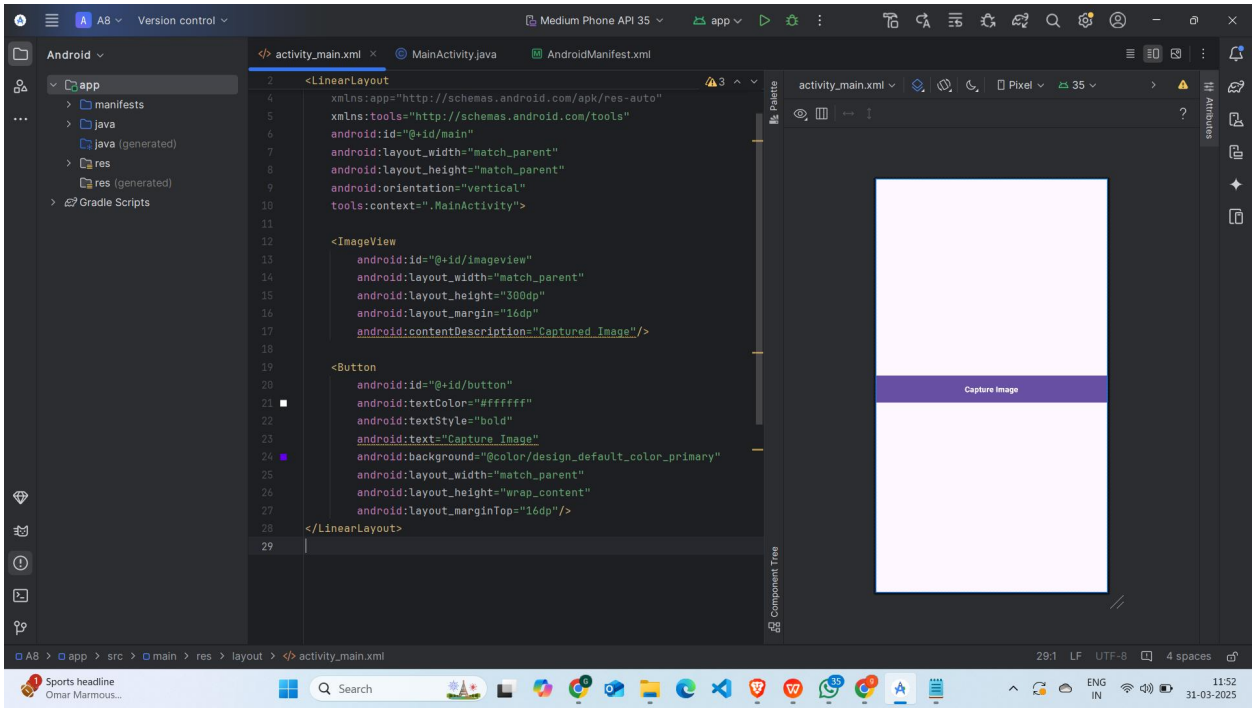
- Open *activity_main.xml* and design the layout using XML.
- Add UI components such as *Button,ImageView* etc.

Screenshot:



Step 3: Writing the Code

- Open *Activity_main.xml*
- Implement functionality such as *Button,Textview.*
- Use necessary Android components like *Buttons,textview* etc
- **Screenshot:**



- **XML**

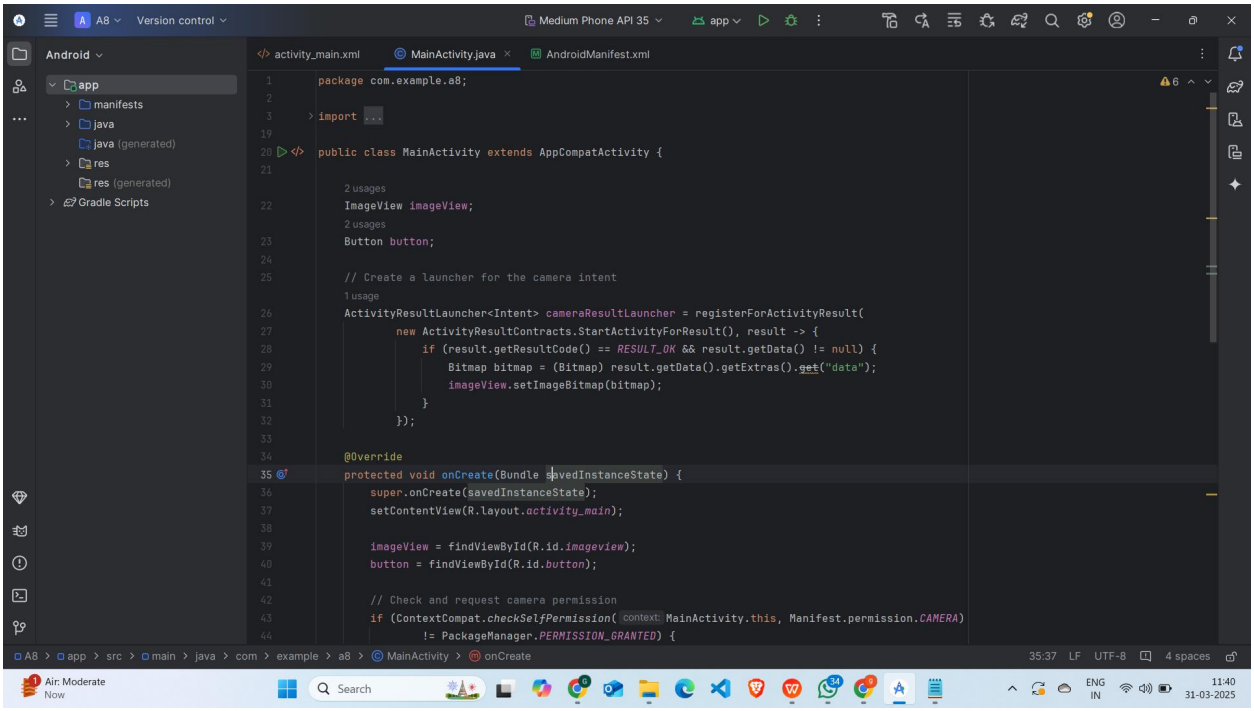
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/imageview"
```

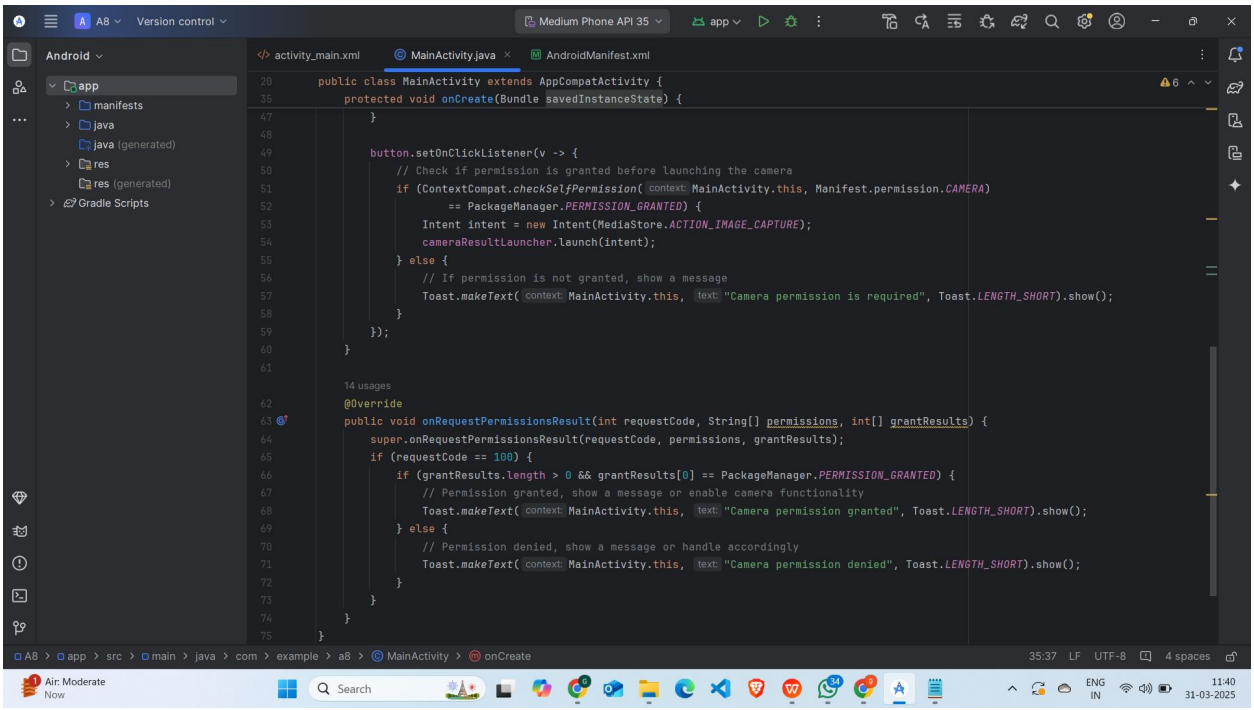
```
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
  <Button
    android:id="@+id/button"
    android:textColor="#ffffff"
    android:textStyle="bold"
    android:text="Capture Image"
    android:background="@color/design_default_color_primary"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>

</LinearLayout>
```

Step 4: Writing the Backend Code (java)

- Open MainActivity.java
- Implement functionality such as By Click on Capture Image the Image is get Clicked .
- When we have to seen to image that we have been Captured then when we click on the image that images is get Open

Screenshot:





Code(java):

```
package com.example.a8;
```

```
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.Manifest;
import android.widget.Toast;
import androidx.activity.result.ActivityResultLauncher;
import androidx.activity.result.contract.ActivityResultContracts;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

public class MainActivity extends AppCompatActivity {

    ImageView imageView;
    Button button;

    // Create a launcher for the camera intent
```

```
ActivityResultLauncher<Intent> cameraResultLauncher = registerForActivityResult(
        new ActivityResultContracts.StartActivityForResult(), result -> {
            if (result.getResultCode() == RESULT_OK && result.getData() != null) {
                Bitmap bitmap = (Bitmap) result.getData().getExtras().get("data");
                imageView.setImageBitmap(bitmap);
            }
        });

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    imageView = findViewById(R.id.imageview);
    button = findViewById(R.id.button);

    // Check and request camera permission
    if (ContextCompat.checkSelfPermission(MainActivity.this, Manifest.permission.CAMERA)
            != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(MainActivity.this,
                new String[]{Manifest.permission.CAMERA}, 100);
    }

    button.setOnClickListener(v -> {
```

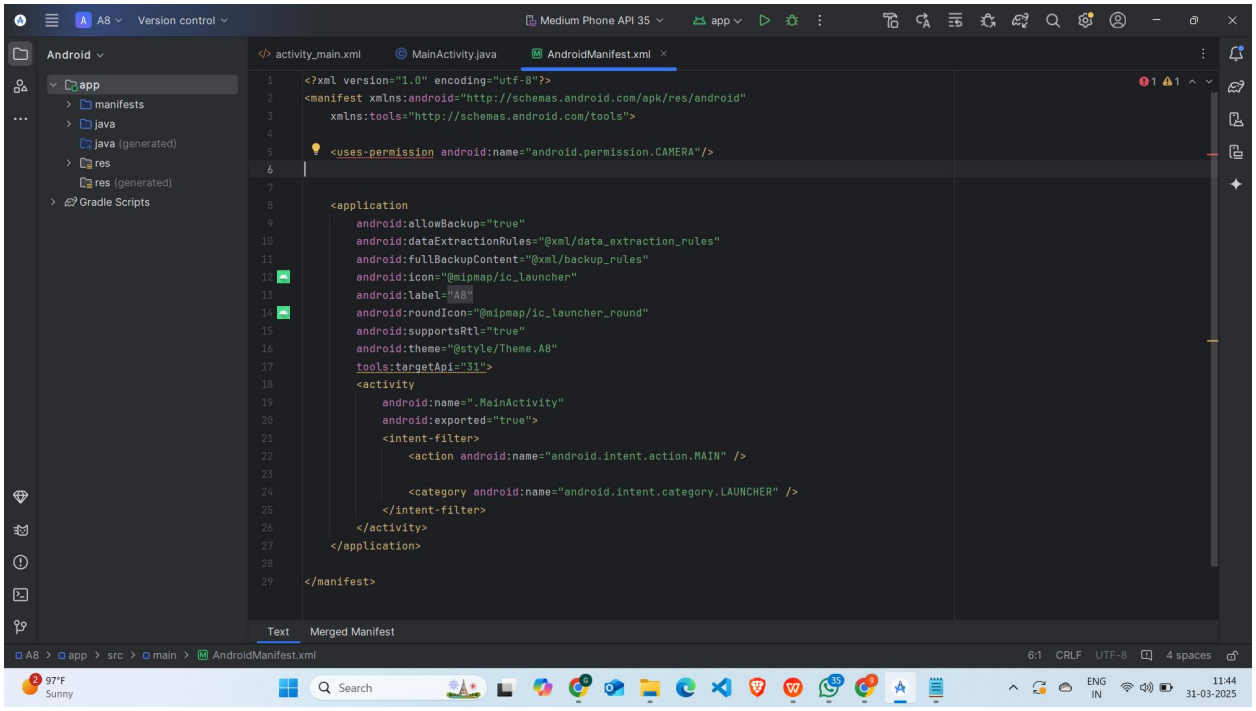
```
        // Check if permission is granted before launching the camera
        if (ContextCompat.checkSelfPermission(MainActivity.this, Manifest.permission.CAMERA)
                == PackageManager.PERMISSION_GRANTED) {
            Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
            cameraResultLauncher.launch(intent);
        } else {
            // If permission is not granted, show a message
            Toast.makeText(MainActivity.this, "Camera permission is required",
Toast.LENGTH_SHORT).show();
        }
    });
  }


  @Override
  public void onRequestPermissionsResult(int requestCode, String[] permissions, int[]
grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == 100) {
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            // Permission granted, show a message or enable camera functionality
            Toast.makeText(MainActivity.this, "Camera permission granted",
Toast.LENGTH_SHORT).show();
        } else {
            // Permission denied, show a message or handle accordingly
            Toast.makeText(MainActivity.this, "Camera permission denied",
Toast.LENGTH_SHORT).show();
```

```
        }
    }
}
}
```

Step 6: Taking Permission To open the Camera

- Open AndroidManifest.xml
- Write the code for the Giving Permission to Opening the Camera

Screenshot:



Code:

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

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

xmlns:tools="http://schemas.android.com/tools">

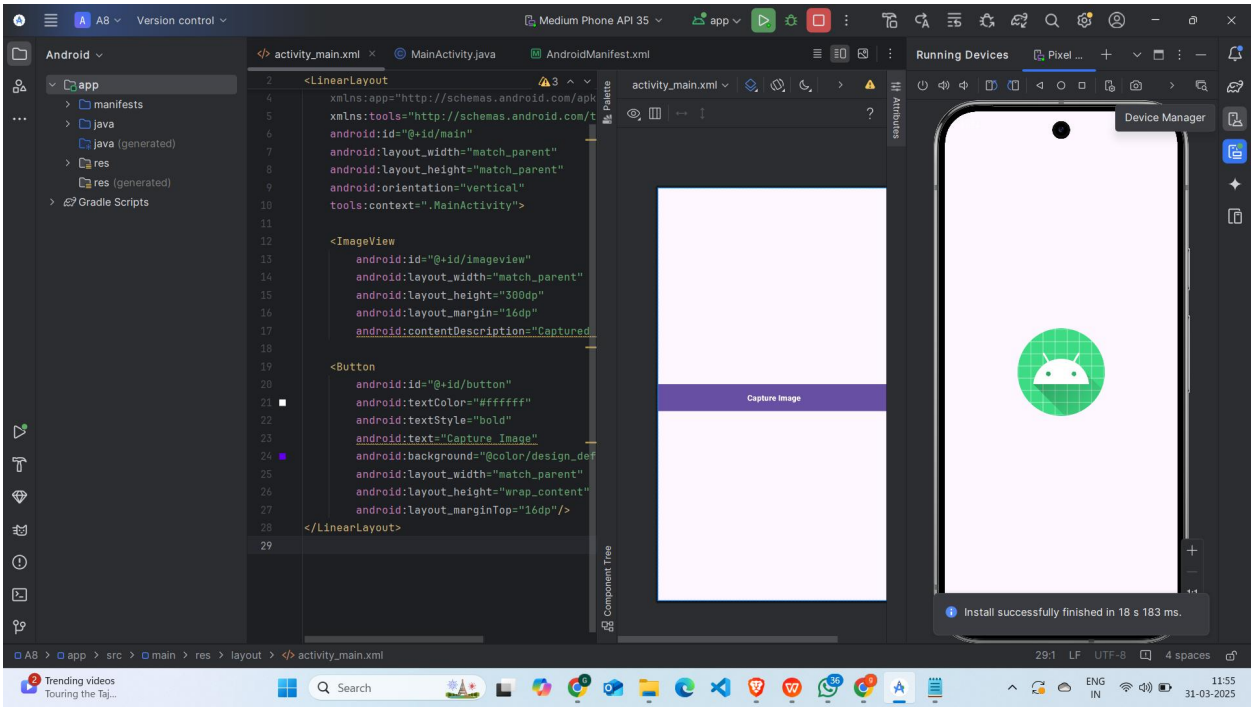
```
<uses-permission android:name="android.permission.CAMERA"/>



<application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.A8"
    tools:targetApi="31">
    <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>

Step 6: Running the Application on Emulator

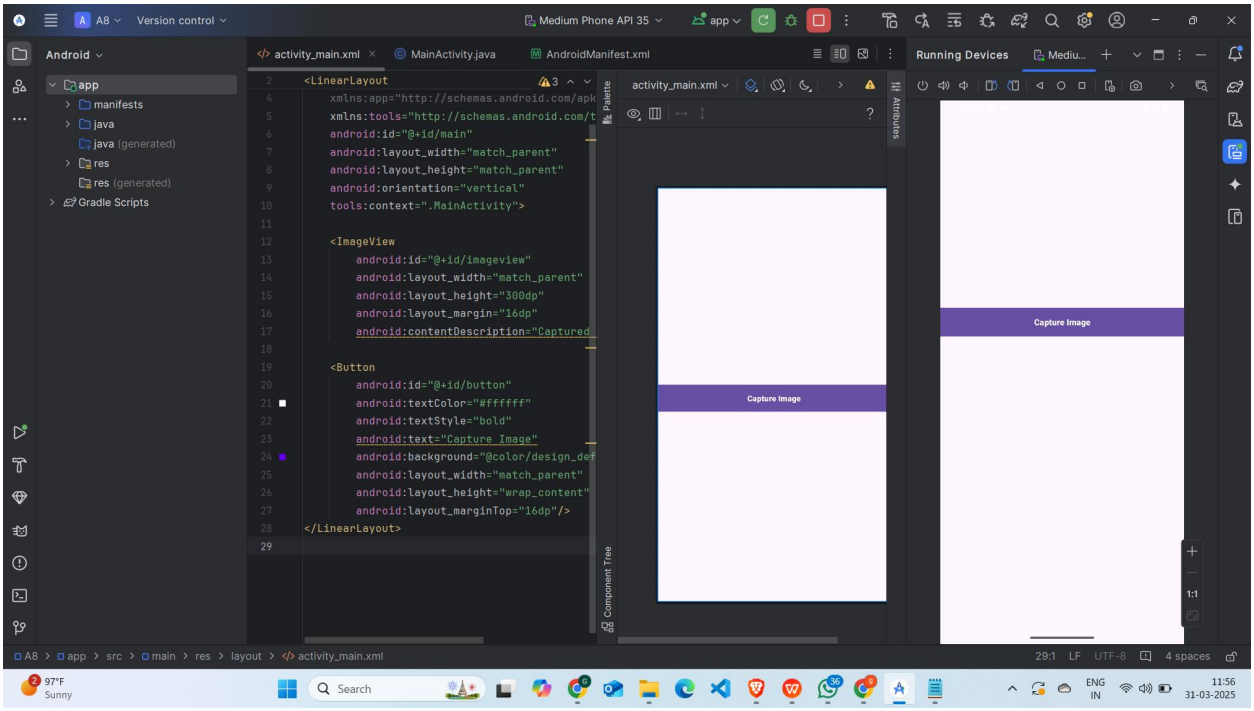
- Click on the **Run** button in Android Studio.
- Select the emulator and launch the app.

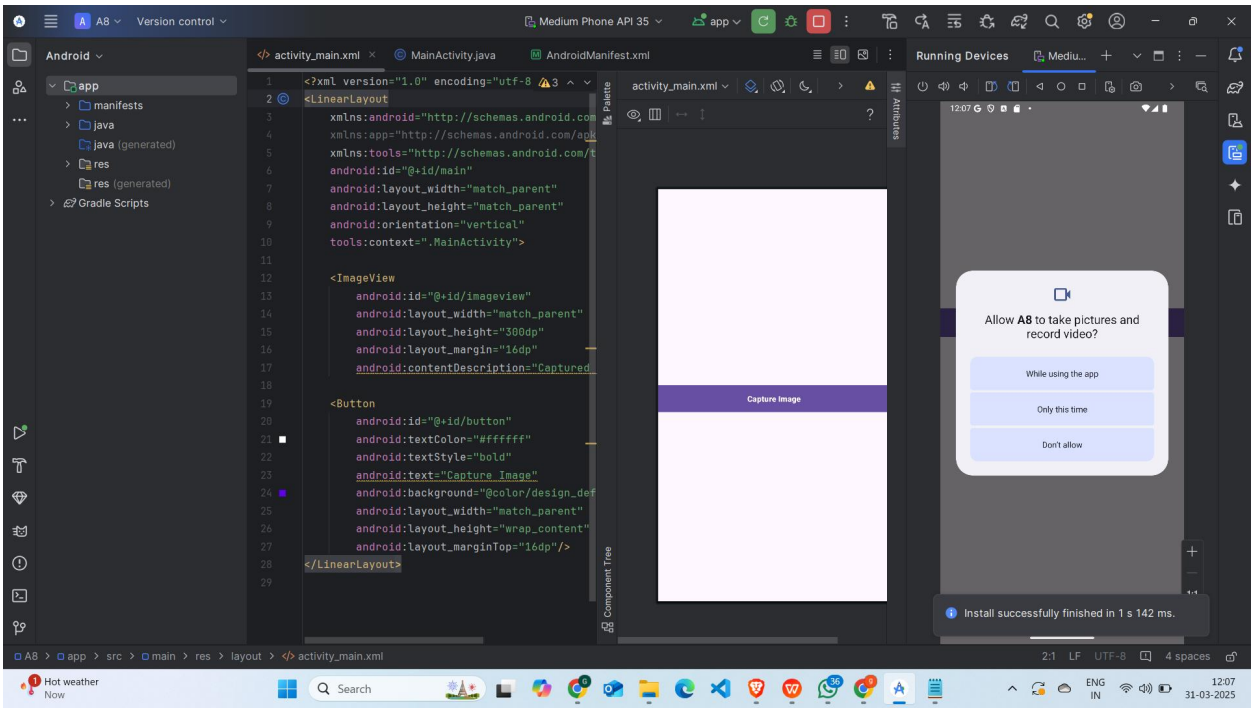
Screenshot:

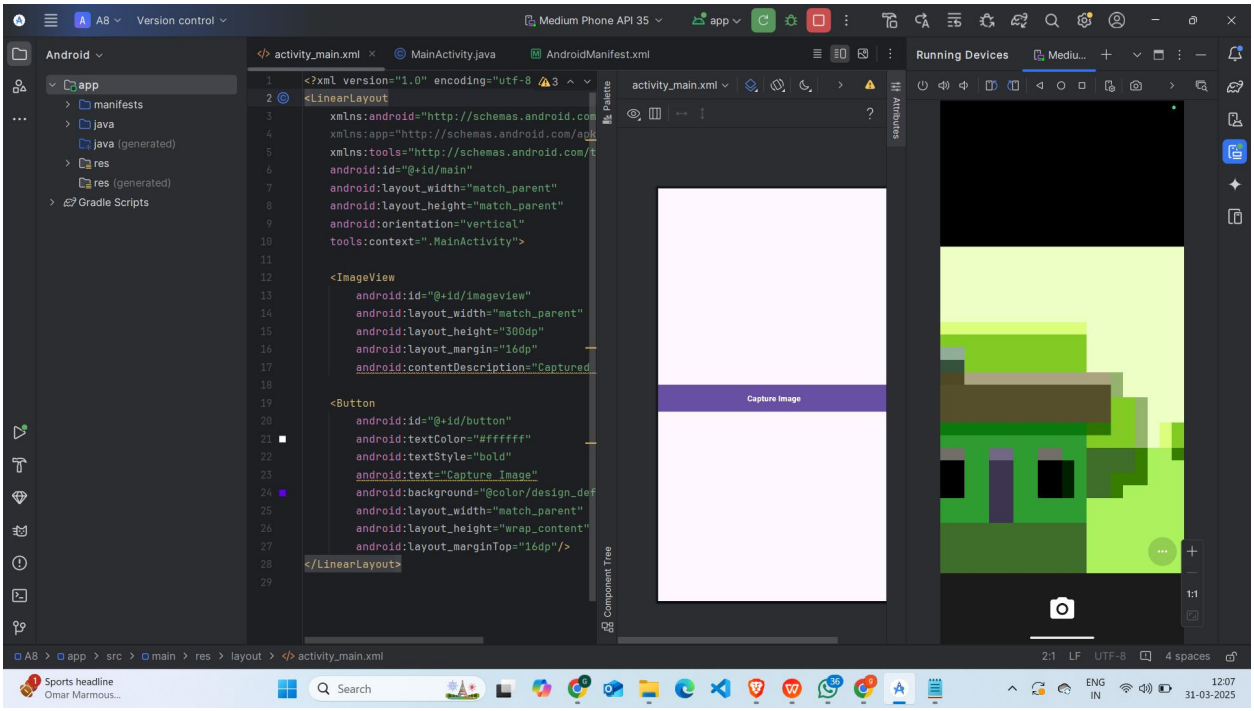


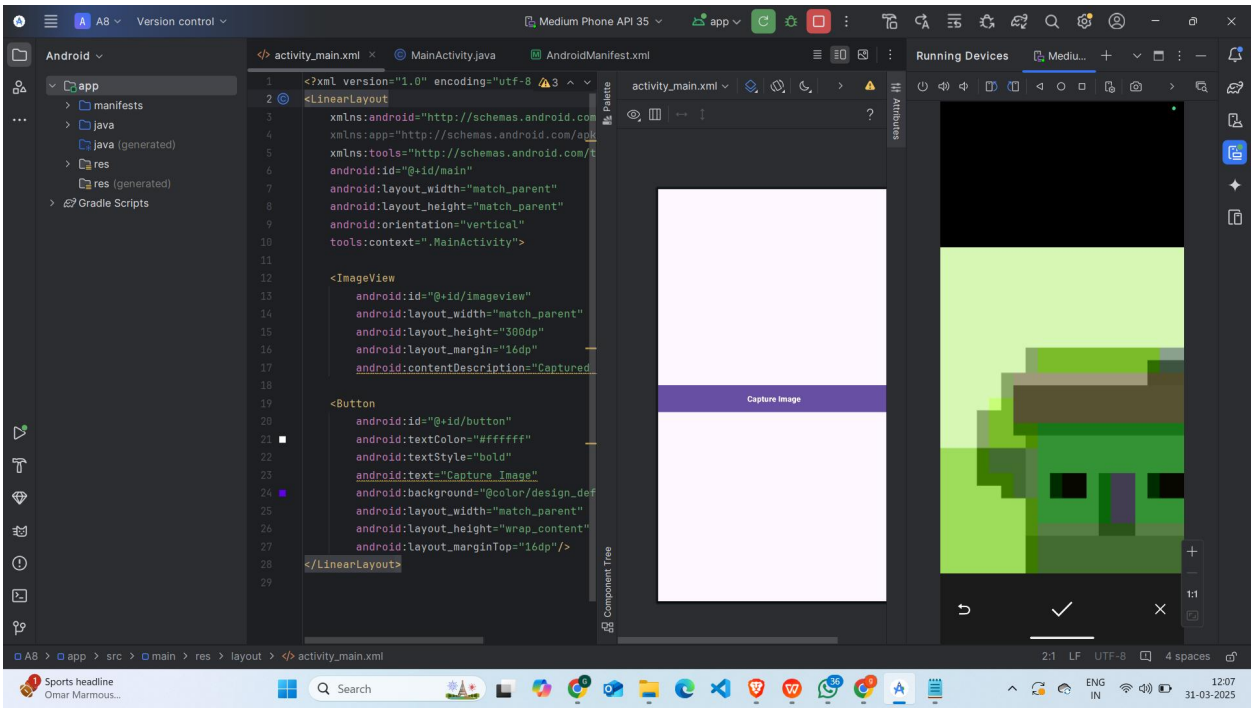
Step 6: Testing & Output

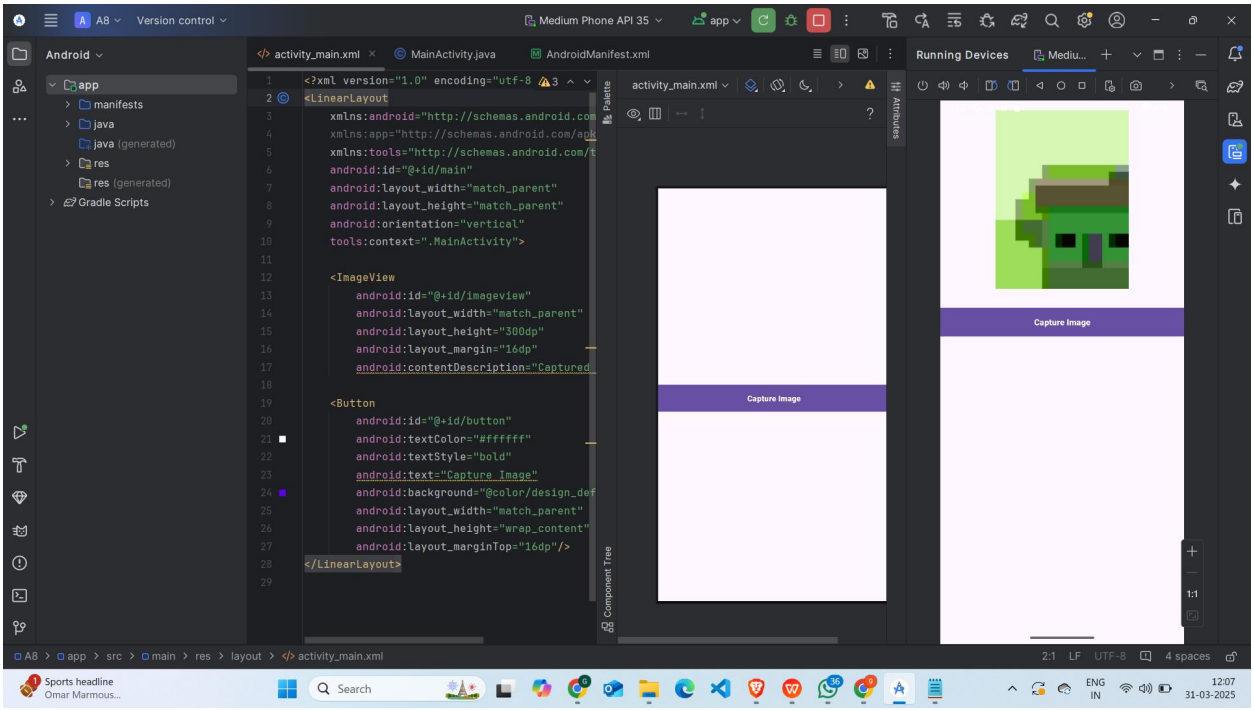
- Test different functionalities of the app.
- Capture the output results.
- **Screenshot:**


Android
app
manifests
java
java (generated)
res
res (generated)
Gradle Scripts
activity_main.xml
MainActivity.java
AndroidManifest.xml
<LinearLayout
xmlns:app="http://schemas.android.com/apk
xmlns:tools="http://schemas.android.com/t
android:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".MainActivity">
<ImageView
android:id="@+id/imageview"
android:layout_width="match_parent"
android:layout_height="300dp"
android:layout_margin="16dp"
android:contentDescription="Captured
<Button
android:id="@+id/button"
android:textColor="#ffffff"
android:textStyle="bold"
android:text="Capture Image"
android:background="@color/design_def
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="16dp"/>
</LinearLayout>
Capture Image
Running Devices
Capture Image
Component Tree
Attributes
Palette


Android
app
manifests
java
java (generated)
res
res (generated)
Gradle Scripts
activity_main.xml
MainActivity.java
AndroidManifest.xml
<?xml version="1.0" encoding="utf-8
<LinearLayout
xmlns:android="http://schemas.android.com
xmlns:app="http://schemas.android.com/apk
xmlns:tools="http://schemas.android.com/t
android:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".MainActivity">
<ImageView
android:id="@+id/imageview"
android:layout_width="match_parent"
android:layout_height="300dp"
android:layout_margin="16dp"
android:contentDescription="Captured
<Button
android:id="@+id/button"
android:textColor="#ffffff"
android:textStyle="bold"
android:text="Capture Image"
android:background="@color/design_def
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="16dp"/>
</LinearLayout>
Capture Image
Running Devices
Allow A8 to take pictures and record video?
While using the app
Only this time
Don't allow
Install successfully finished in 1 s 142 ms.
Component Tree
Attributes
Palette


A8
Version control
Medium Phone API 35
app
Android
activity_main.xml
MainActivity.java
AndroidManifest.xml
Running Devices
app
manifests
java
java (generated)
res
res (generated)
Gradle Scripts
<?xml version="1.0" encoding="utf-8
<LinearLayout
xmlns:android="http://schemas.android.com
xmlns:app="http://schemas.android.com/apk
xmlns:tools="http://schemas.android.com/t
android:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".MainActivity">
<ImageView
android:id="@+id/imageview"
android:layout_width="match_parent"
android:layout_height="300dp"
android:layout_margin="16dp"
android:contentDescription="Captured
<Button
android:id="@+id/button"
android:textColor="#ffffff"
android:textStyle="bold"
android:text="Capture Image"
android:background="@color/design_def
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="16dp"/>
</LinearLayout>
Capture Image
Palette
Attributes
Component Tree
1:1
A8 > app > src > main > res > layout > activity_main.xml
2:1 LF UTF-8 4 spaces
Sports headline
Omar Marmous...
Search
ENG
IN
12:07
31-03-2025


A8
Version control
Medium Phone API 35
app
Android
activity_main.xml
MainActivity.java
AndroidManifest.xml
Running Devices
app
manifests
java
java (generated)
res
res (generated)
Gradle Scripts
<?xml version="1.0" encoding="utf-8
<LinearLayout
xmlns:android="http://schemas.android.com
xmlns:app="http://schemas.android.com/apk
xmlns:tools="http://schemas.android.com/t
android:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".MainActivity">
<ImageView
android:id="@+id/imageview"
android:layout_width="match_parent"
android:layout_height="300dp"
android:layout_margin="16dp"
android:contentDescription="Captured
<Button
android:id="@+id/button"
android:textColor="#ffffff"
android:textStyle="bold"
android:text="Capture Image"
android:background="@color/design_def
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="16dp"/>
</LinearLayout>
Capture Image
Palette
Attributes
Component Tree
1:1
A8 > app > src > main > res > layout > activity_main.xml
2:1 LF UTF-8 4 spaces
Sports headline
Omar Marmous...
Search
ENG
IN
12:07
31-03-2025



4. Conclusion

In this assignment, I successfully developed an Android app that captures images using the built-in camera. I learned how to access and utilize the camera functionality in Android, handle image processing, and display captured images on the screen. During development, I encountered challenges related to managing camera resources, handling image orientations, and ensuring smooth camera operations, but I overcame them by researching relevant documentation and experimenting with various methods. This project improved my understanding of Android camera handling, user interface design, and image processing. Overall, this assignment was a valuable experience that enhanced my skills in Android development and helped me understand how to implement a simple camera application in a mobile device.