# **ASSIGNMENT NO.6**

# **Gaurav\_bodkhe\_2124UCEM1041**

## **1. Introduction**

## In this program, we will develop a simple Android application that enables users to play audio files on their devices. The app will feature a user-friendly interface with **Play**, **Pause**, and **Stop** buttons to control audio playback. Additionally, users will have the ability to select their favorite songs from the device's storage. Upon selecting a song, the app will start playing it, and users can pause, resume, or stop the audio as desired. This project demonstrates how to build an audio player in Android, providing users with an intuitive way to listen to their favorite music directly from their mobile devices.

## **2. Tools & Technologies Used**

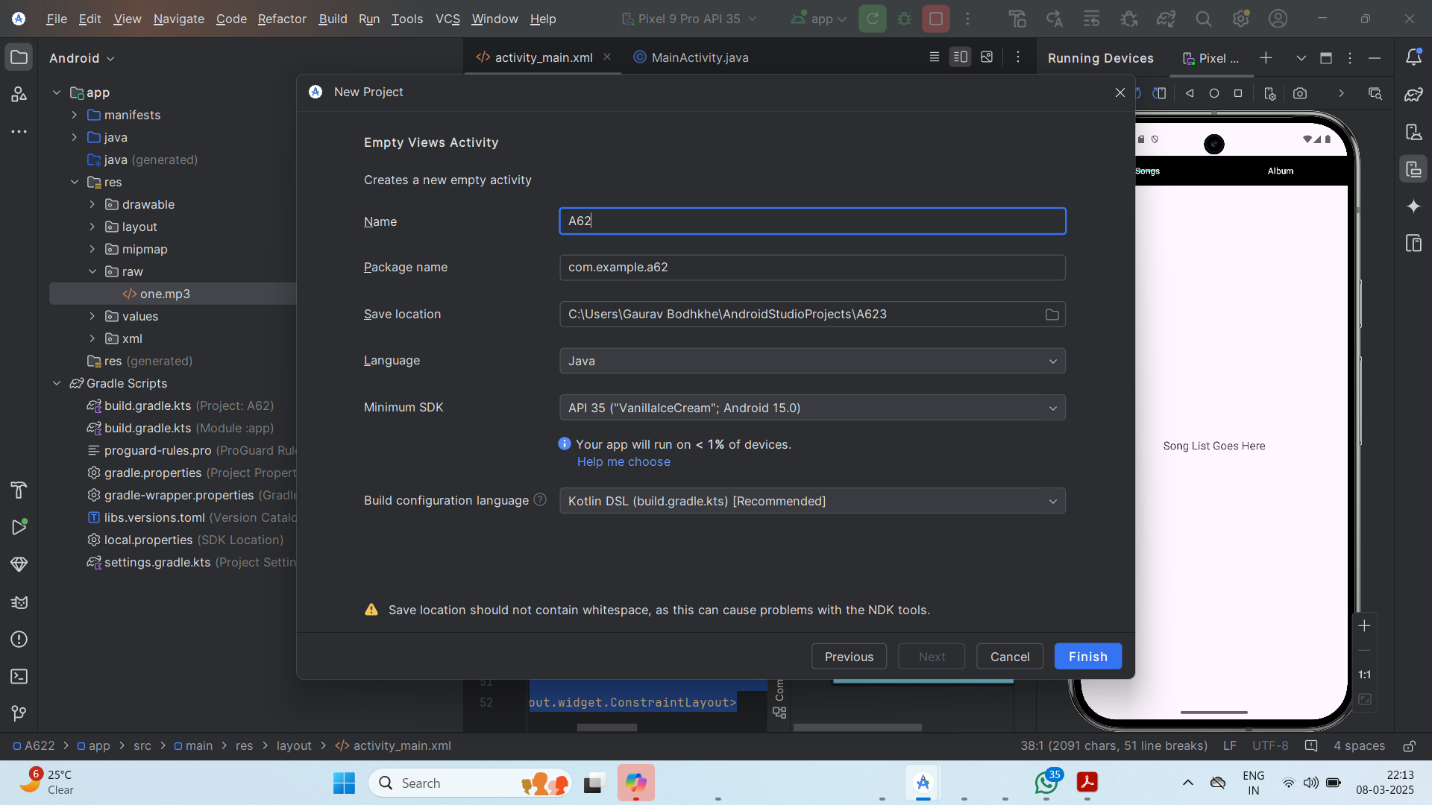
* Android Studio
* Java
* Emulator

## **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:**



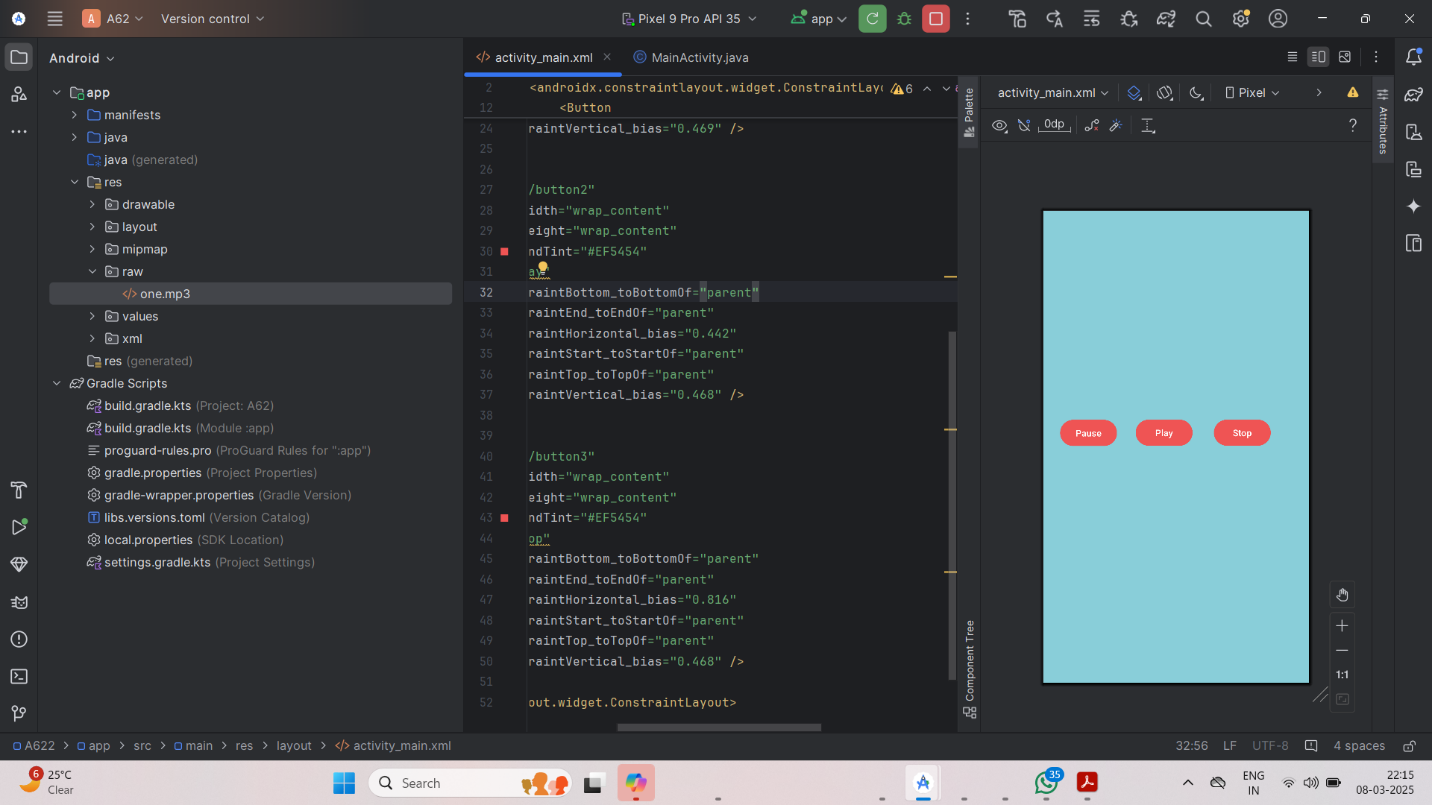
### 

### **Step 2: Designing the UI**

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

**Screenshot:**

For Play,Pause,Stop Buttons=

****

For Songs & Album Attribute=

**A screen shot of a computer

AI-generated content may be incorrect.**

### **Step 3: Writing the Code**

* Open Activity\_main.xml
* Implement functionality such as Pause, Play, Stop, Songs Albums.
* Use necessary Android components like Buttons,Table etc
* **Screenshot:**

For Play,Pause,Stop Buttons=

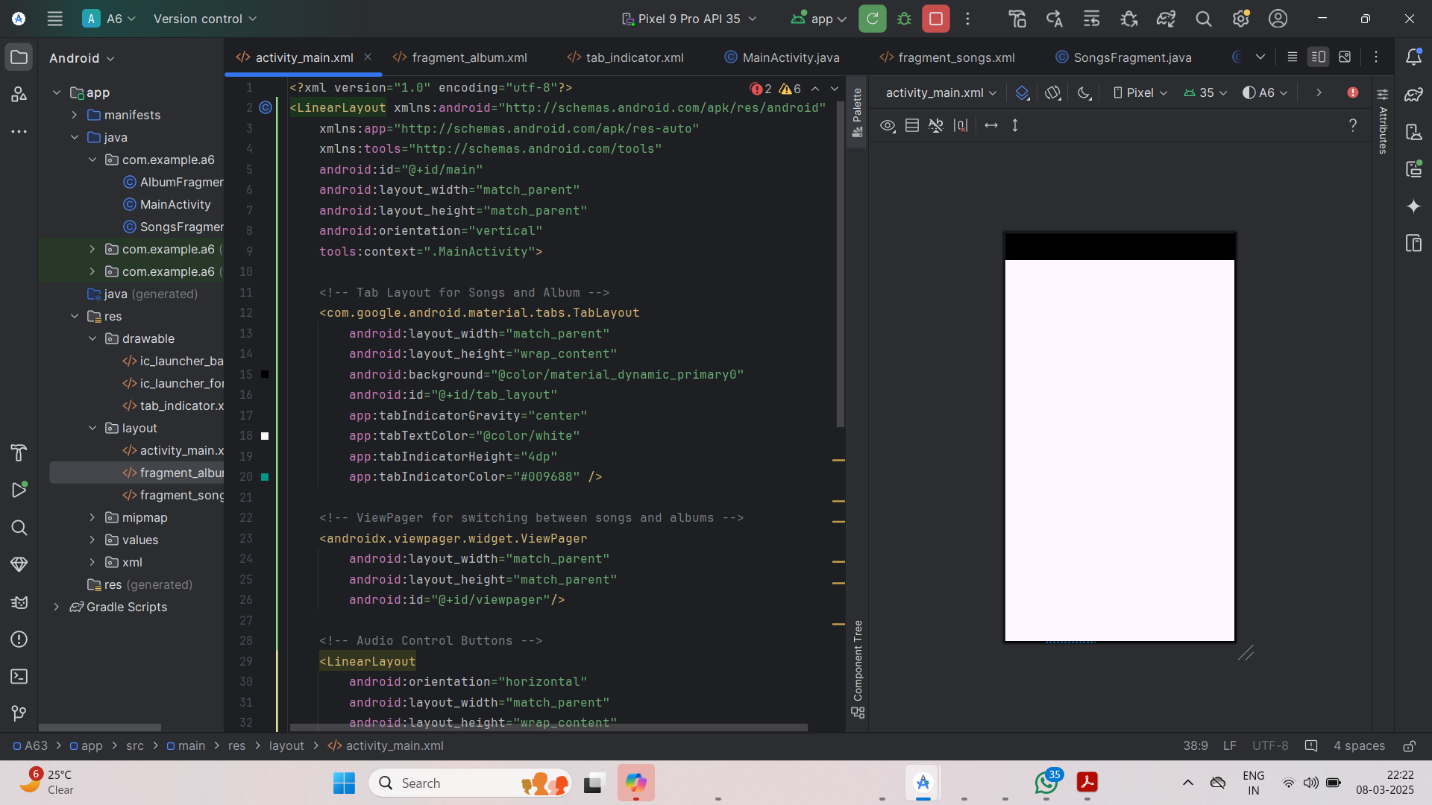
*A screenshot of a computer

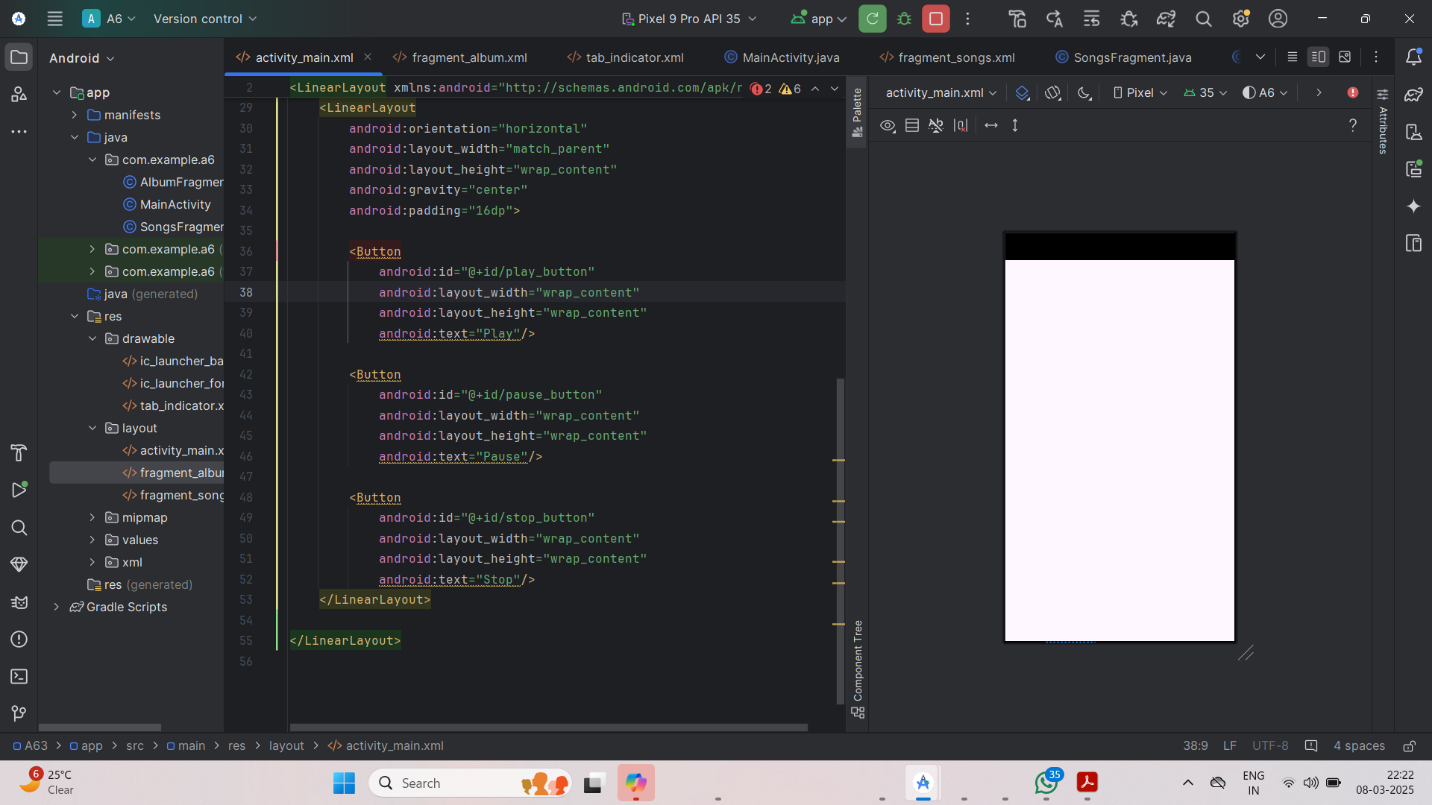
AI-generated content may be incorrect.*

*A screenshot of a computer

AI-generated content may be incorrect.*

For Songs & Album Attribute=





* **XML**

For Play,Pause,Stop Buttons=

*<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout 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:background="#89CED9"  
 tools:context=".MainActivity">  
  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:backgroundTint="#EF5454"  
 android:text="Pause"  
 android:textColorLink="#E92727"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.08"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.469" />  
  
 <Button  
 android:id="@+id/button2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:backgroundTint="#EF5454"  
 android:text="Play"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.442"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.468" />  
  
 <Button  
 android:id="@+id/button3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:backgroundTint="#EF5454"  
 android:text="Stop"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.816"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.468" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>*

For Songs & Album Attribute=

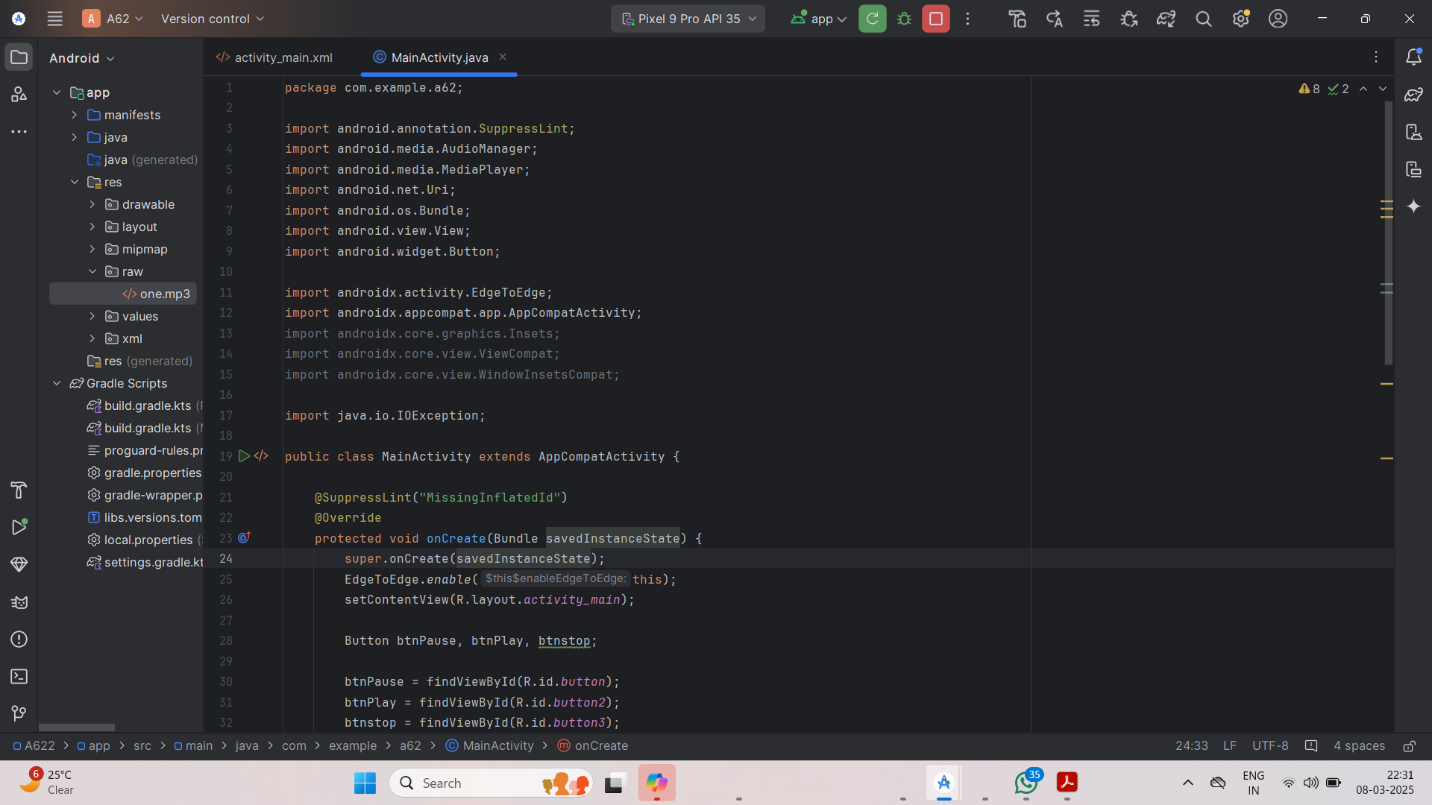
*<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".MainActivity">  
  
 <!-- Tab Layout for Songs and Album -->  
 <com.google.android.material.tabs.TabLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="@color/material\_dynamic\_primary0"  
 android:id="@+id/tab\_layout"  
 app:tabIndicatorGravity="center"  
 app:tabTextColor="@color/white"  
 app:tabIndicatorHeight="4dp"  
 app:tabIndicatorColor="#009688" />  
  
 <!-- ViewPager for switching between songs and albums -->  
 <androidx.viewpager.widget.ViewPager  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/viewpager"/>  
  
 <!-- Audio Control Buttons -->  
 <LinearLayout  
 android:orientation="horizontal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 android:padding="16dp">  
  
 <Button  
 android:id="@+id/play\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Play"/>  
  
 <Button  
 android:id="@+id/pause\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Pause"/>  
  
 <Button  
 android:id="@+id/stop\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Stop"/>  
 </LinearLayout>  
  
</LinearLayout>*

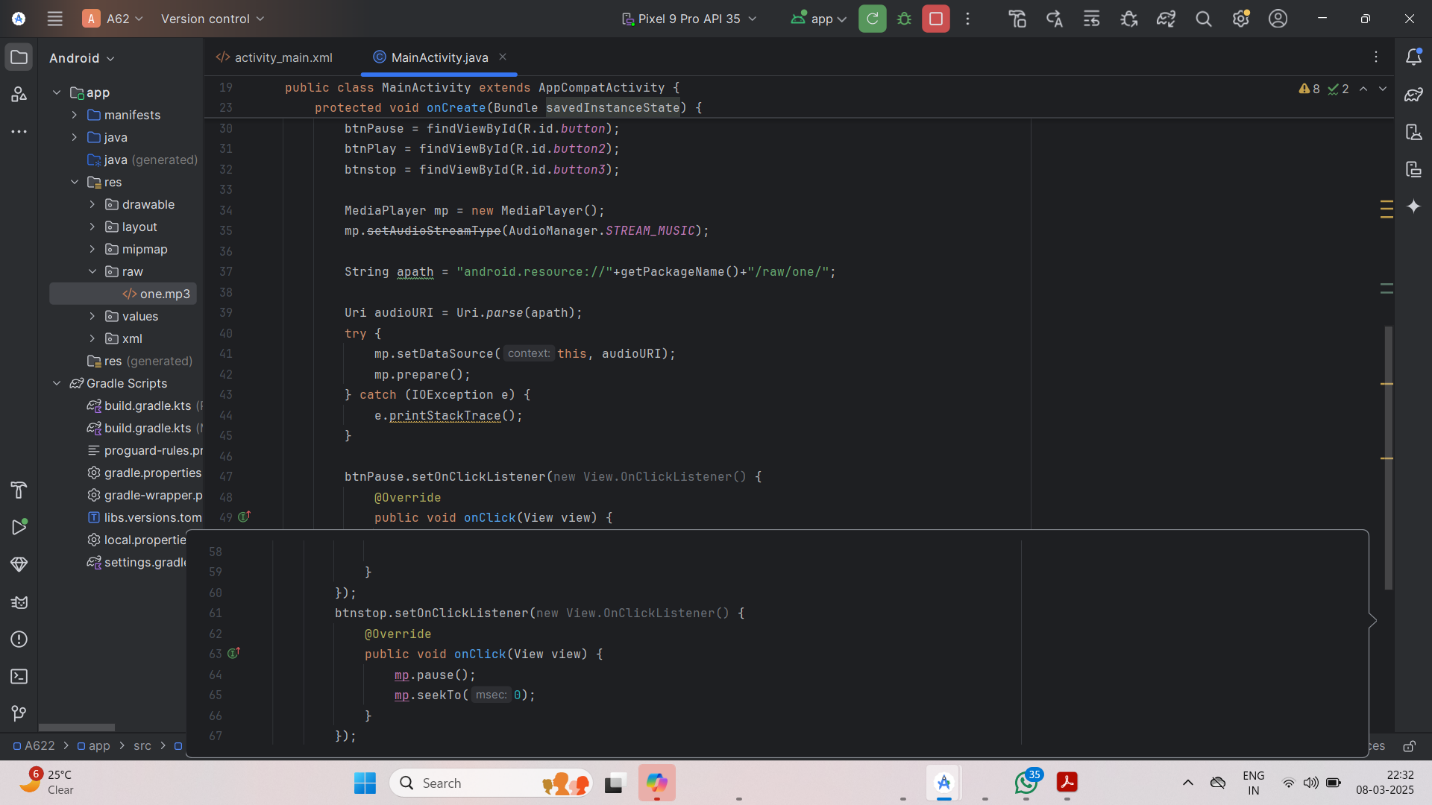
### **Step 4: Writing the Backend Code (java)**

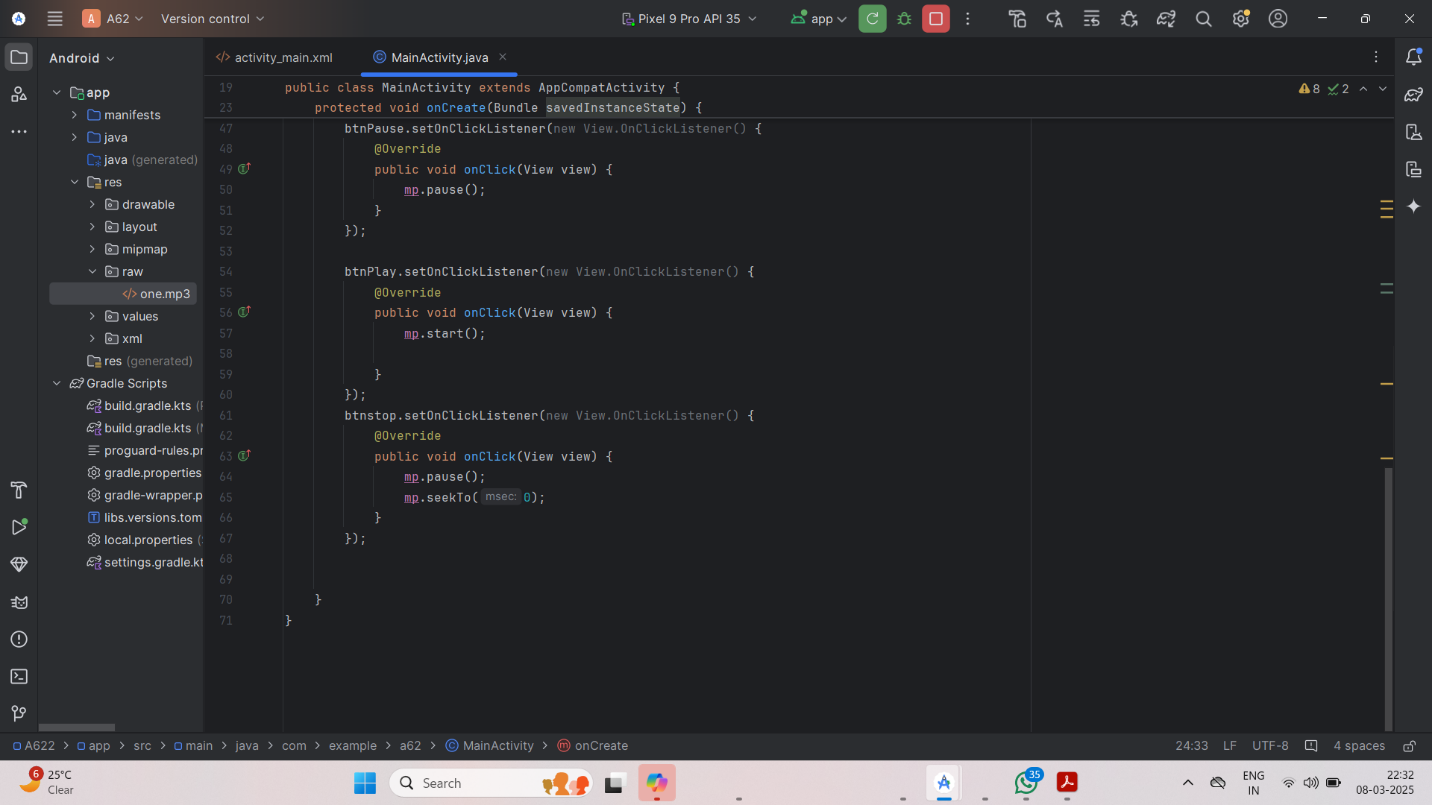
* Open MainActivity.java
* Implement functionality such as By Click on Play the Song is Get Start , when Click on The Pause the Song is get Pause Whereas On Click the Stop the song is get Stop And get start Again.
* For the other Page (Songs And Album) when click on the Songs All the Songs are Get Visible Whereas when Clicking on the Album the Songs are get open they we will be add in Folder e.g. Favorite Songs Folder

**Screenshot:**

For Play,Pause,Stop Buttons=

****

****

****

For Songs & Album Attribute=

A screen shot of a computer

AI-generated content may be incorrect.A computer screen with many colorful text

AI-generated content may be incorrect.A computer screen with text on it

AI-generated content may be incorrect.A computer screen with text on it

AI-generated content may be incorrect.

## **Code(java):**

For Play,Pause,Stop Buttons=

*package com.example.a62;  
  
import android.annotation.SuppressLint;  
import android.media.AudioManager;  
import android.media.MediaPlayer;  
import android.net.Uri;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
  
import androidx.activity.EdgeToEdge;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.graphics.Insets;  
import androidx.core.view.ViewCompat;  
import androidx.core.view.WindowInsetsCompat;  
  
import java.io.IOException;  
  
public class MainActivity extends AppCompatActivity {  
  
 @SuppressLint("MissingInflatedId")  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.enable(this);  
 setContentView(R.layout.activity\_main);  
  
 Button btnPause, btnPlay, btnstop;  
  
 btnPause = findViewById(R.id.button);  
 btnPlay = findViewById(R.id.button2);  
 btnstop = findViewById(R.id.button3);  
  
 MediaPlayer mp = new MediaPlayer();  
 mp.setAudioStreamType(AudioManager.STREAM\_MUSIC);  
  
 String apath = "android.resource://"+getPackageName()+"/raw/one/";  
  
 Uri audioURI = Uri.parse(apath);  
 try {  
 mp.setDataSource(this, audioURI);  
 mp.prepare();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
 btnPause.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 mp.pause();  
 }  
 });  
  
 btnPlay.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 mp.start();  
  
 }  
 });  
 btnstop.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 mp.pause();  
 mp.seekTo(0);  
 }  
 });  
  
  
 }  
}*

For Songs & Album Attribute=

*package com.example.a6;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import androidx.activity.EdgeToEdge;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.graphics.Insets;  
import androidx.core.view.ViewCompat;  
import androidx.core.view.WindowInsetsCompat;  
import androidx.fragment.app.Fragment;  
import androidx.fragment.app.FragmentManager;  
import androidx.fragment.app.FragmentPagerAdapter;  
import androidx.viewpager.widget.ViewPager;  
import com.google.android.material.tabs.TabLayout;  
import java.util.ArrayList;  
  
public class MainActivity extends AppCompatActivity {  
  
 private ViewPager viewPager;  
 private TabLayout tabLayout;  
 private Button playButton, pauseButton, stopButton;  
  
 @Override  
  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.enable(this);  
 setContentView(R.layout.activity\_main);  
  
 initViewPager();  
 initButtons();  
  
 // Handling window insets for edge-to-edge display  
 ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {  
 Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());  
 v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);  
 return insets;  
 });  
 }  
  
 private void initViewPager() {  
 viewPager = findViewById(R.id.viewpager);  
 tabLayout = findViewById(R.id.tab\_layout);  
 ViewPagerAdapter viewPagerAdapter = new ViewPagerAdapter(getSupportFragmentManager());  
 viewPagerAdapter.addFragment(new SongsFragment(), "Songs");  
 viewPagerAdapter.addFragment(new AlbumFragment(), "Album");  
 viewPager.setAdapter(viewPagerAdapter);  
 tabLayout.setupWithViewPager(viewPager);  
 }  
  
 private void initButtons() {  
 playButton = findViewById(R.id.play\_button);  
 pauseButton = findViewById(R.id.pause\_button);  
 stopButton = findViewById(R.id.stop\_button);  
  
 playButton.setOnClickListener(v -> {  
 // Handle play button logic  
 });  
  
 pauseButton.setOnClickListener(v -> {  
 // Handle pause button logic  
 });  
  
 stopButton.setOnClickListener(v -> {  
 // Handle stop button logic  
 });  
 }  
  
 public static class ViewPagerAdapter extends FragmentPagerAdapter {  
  
 private ArrayList<Fragment> fragments;  
 private ArrayList<String> titles;  
  
 public ViewPagerAdapter(FragmentManager fm) {  
 super(fm);  
 this.fragments = new ArrayList<>();  
 this.titles = new ArrayList<>();  
 }  
  
 void addFragment(Fragment fragment, String title) {  
 fragments.add(fragment);  
 titles.add(title);  
 }  
  
 @Override  
 public Fragment getItem(int position) {  
 return fragments.get(position);  
 }  
  
 @Override  
 public int getCount() {  
 return fragments.size();  
 }  
  
 @Override  
 public CharSequence getPageTitle(int position) {  
 return titles.get(position);  
 }  
 }  
}*

### **Step 5: Running the Application on Emulator**

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

**Screenshot:**

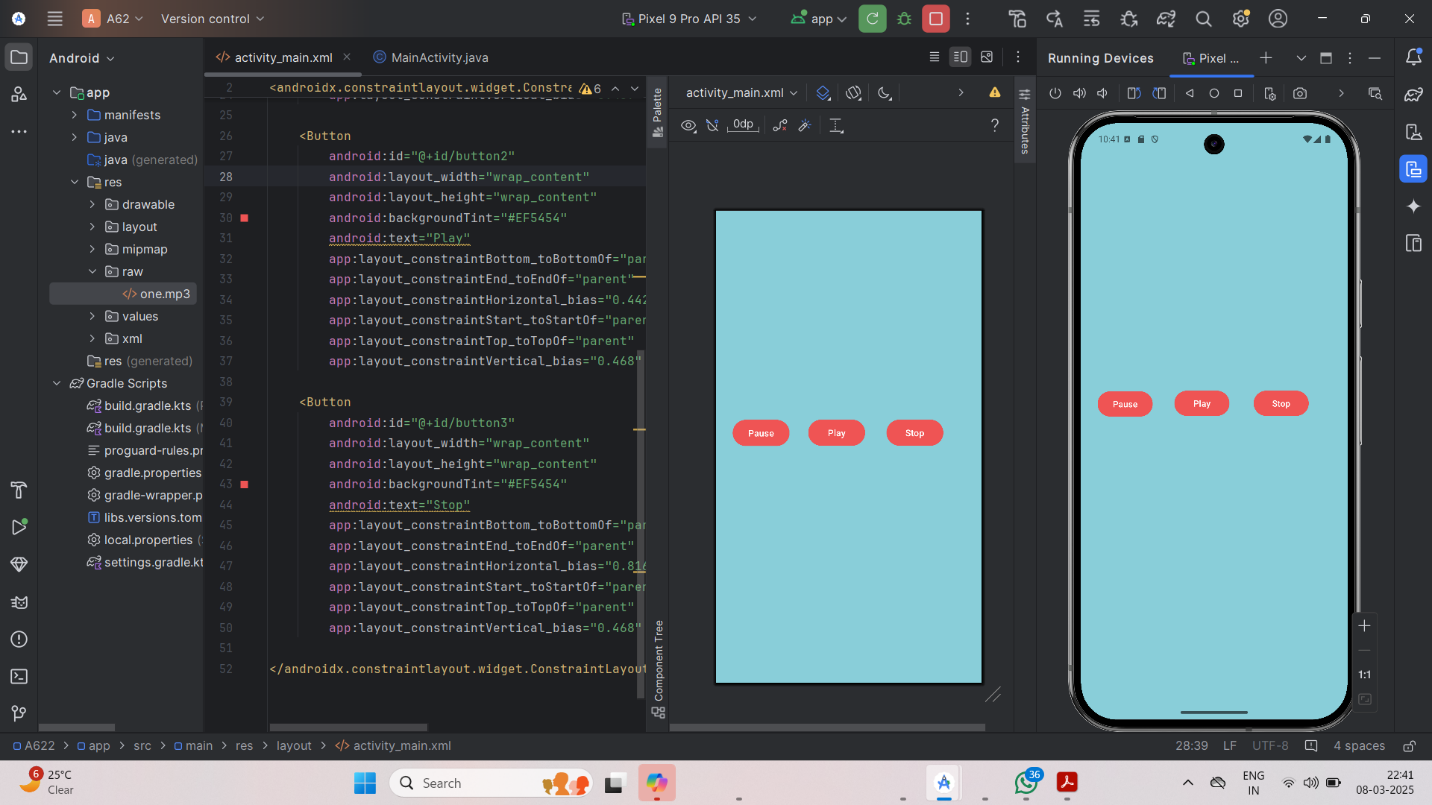
### 

### 

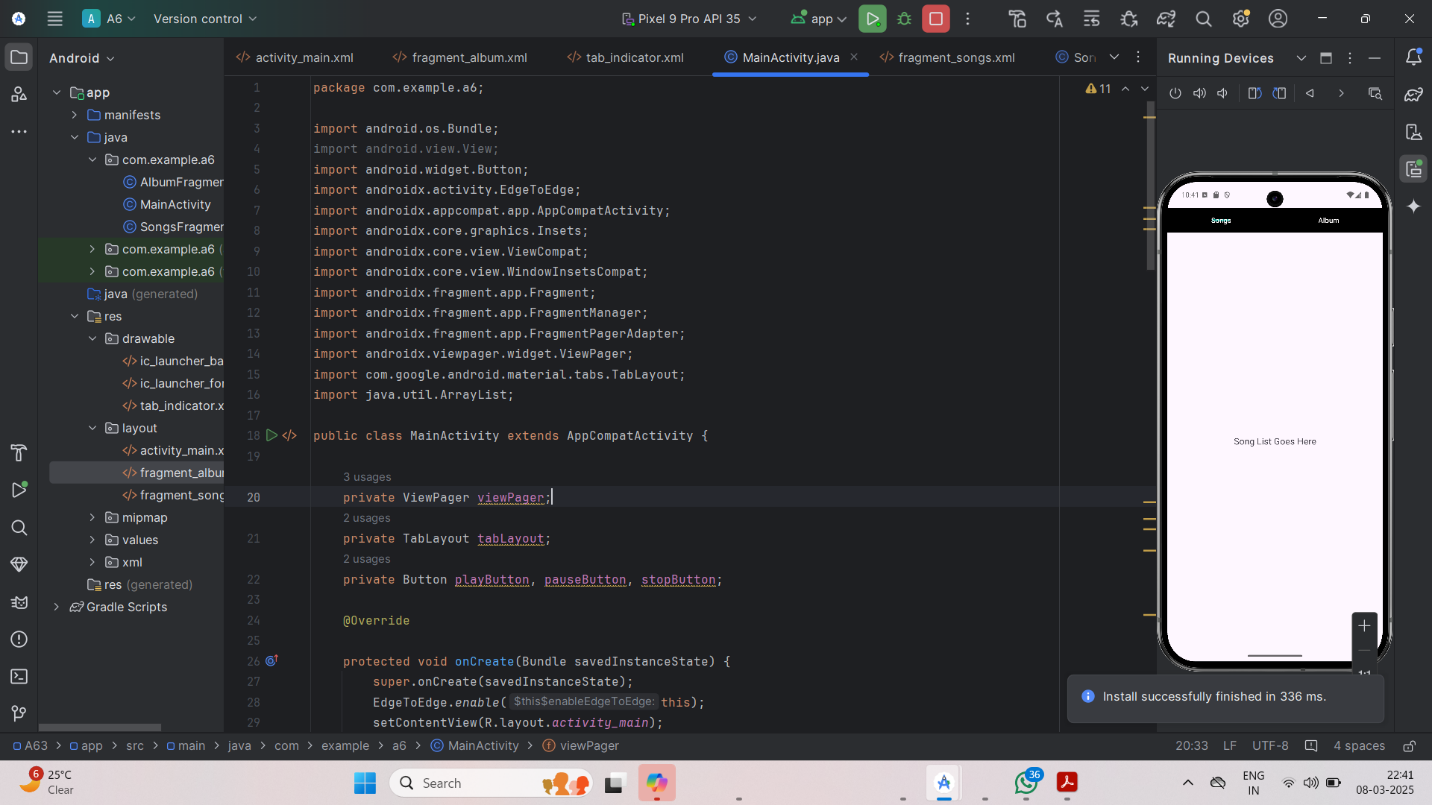
### **Step 6: Testing & Output**

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

For Play,Pause,Stop Buttons=



For Songs & Album Attribute=

****

## **4. Conclusion**

## In this assignment, I successfully developed an Android app that plays audio files with basic control functionalities such as Play, Pause, and Stop. I learned how to handle media playback in Android using the MediaPlayer class and how to manage different audio files stored in the res/raw directory.

## During development, I encountered challenges related to managing media resources and ensuring smooth audio playback, but I overcame them by researching relevant documentation and experimenting with various methods. This project improved my understanding of Android media handling and user interface design.

## Overall, this assignment was a valuable experience that enhanced my skills in Android development and helped me understand how to implement a simple media player in a mobile application.