# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

## “Jnana Sangama”, Belagavi, Karnataka, India.



A Mini Project Report On

**Media Player Application**

*Submitted in partial fulfillment of the requirement for VI Semester*

***Mobile Application Development Laboratory[18CSMP68]***

## in

**Computer Science and Engineering**

*Submitted By*

**Arpitha N G [1GA18CS031]**

**Amrutha K [1GA18CS194]**

*Under the guidance of*

**Dr. KAVITHA K S, Professor**

**Prof. SOWMYA M, Assistant Professor**



**Department of Computer Science and Engineering**

**(Accredited by NBA 2019-2022)**

**GLOBAL ACADEMY OF TECHNOLOGY**

Rajarajeshwari Nagar, Bengaluru - 560 098

## 2020 – 2021

# ABSTRACT

The project is an Android Studio-based media player application. In addition, the media player will add some features based on the user's usage scenario. The media Player, as usual, has a limited amount of tracks kept in it, and we may use it to play, pause, and advance an audio or song. To shift the audio forward or backward, use the seek bar's indicator. The project is comprised of a unique dashboard, and we can also refresh the media player; when we refresh, a notice stating "refreshing" appears. The song's name is likewise displayed in a similar manner. When we click on the "About" option on the dashboard, a pop-up window appears with information about the section.

**IMPLEMENTATION**

**Program code:**

* **activity\_main.xml**

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

<android.support.v4.widget.DrawerLayout

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/drawer\_layout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".Activity.MainActivity">

<RelativeLayout

android:id="@+id/content\_frame"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<android.support.v7.widget.Toolbar

android:id="@+id/toolbar"

android:layout\_width="match\_parent"

android:layout\_height="?attr/actionBarSize"

android:background="?attr/colorPrimary"

android:theme="@style/ThemeOverlay.AppCompat.ActionBar" />

<LinearLayout

android:id="@+id/ll\_include\_controls"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@id/toolbar">

<include layout="@layout/player\_layout" />

</LinearLayout>

<LinearLayout

android:id="@+id/ll\_tab\_layout"

android:layout\_width="match\_parent"

android:layout\_height="?attr/actionBarSize"

android:layout\_below="@id/ll\_include\_controls"

android:background="@color/colorPrimary">

<android.support.design.widget.TabLayout

android:id="@+id/tabs"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:elevation="10dp"

app:tabBackground="@color/colorPrimary"

app:tabGravity="fill"

app:tabIndicatorColor="@color/colorPrimary"

app:tabMode="fixed"

app:tabSelectedTextColor="@color/text\_color"

app:tabTextColor="@color/off\_color" />

</LinearLayout>

<android.support.v4.view.ViewPager

android:id="@+id/songs\_viewpager"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/ll\_tab\_layout"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"/>

<android.support.design.widget.FloatingActionButton

android:id="@+id/btn\_refresh"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:layout\_alignParentEnd="true"

android:layout\_marginBottom="15dp"

android:layout\_marginEnd="15dp"

android:focusable="true"

android:src="@drawable/refresh\_icon"

app:backgroundTint="@color/colorPrimary"

app:layout\_anchor="@id/songs\_viewpager" />

</RelativeLayout>

<android.support.design.widget.NavigationView

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

android:layout\_width="wrap\_content"

android:layout\_height="match\_parent"

android:layout\_gravity="start"

android:background="@color/drawer\_color"

android:fitsSystemWindows="true"

app:headerLayout="@layout/drawer\_header\_layout"

app:itemIconTint="@color/text\_color"

app:itemTextColor="@color/text\_color"

app:menu="@menu/navigation\_menu" />

</android.support.v4.widget.DrawerLayout>

* **drawer\_header\_layout.xml**

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

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

android:layout\_width="match\_parent"

android:layout\_height="200dp"

android:layout\_centerHorizontal="true"

android:background="@drawable/header\_background"

android:fitsSystemWindows="true"

android:gravity="center">

<ImageView

android:layout\_marginStart="10dp"

android:id="@+id/img\_icon"

android:layout\_width="120dp"

android:layout\_height="120dp"

android:focusable="true"

android:src="@drawable/headset\_icon" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/img\_icon"

android:text="@string/title"

android:textColor="@color/text\_color"

android:textSize="30sp"

android:textStyle="bold" />

</RelativeLayout>

* **fragment\_tab.xml**

<?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:paddingLeft="10dp"

android:paddingRight="10dp">

<ListView

android:id="@+id/list\_playlist"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:divider="@color/gray\_color"

android:dividerHeight="1dp"

android:longClickable="true" />

</LinearLayout>

* **player\_layout.xml**

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

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

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="100dp"

android:background="@color/colorPrimary">

<LinearLayout

android:id="@+id/ll\_player\_controls"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="10dp"

android:gravity="center\_horizontal"

android:orientation="horizontal">

<ImageButton

android:id="@+id/img\_btn\_replay"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="10dp"

android:backgroundTint="@color/colorPrimary"

android:src="@drawable/undo\_icon" />

<ImageButton

android:id="@+id/img\_btn\_previous"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="30dp"

android:backgroundTint="@color/colorPrimary"

android:src="@drawable/previous\_icon" />

<ImageButton

android:id="@+id/img\_btn\_play"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="10dp"

android:backgroundTint="@color/colorPrimary"

android:src="@drawable/play\_icon" />

<ImageButton

android:id="@+id/img\_btn\_next"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="10dp"

android:backgroundTint="@color/colorPrimary"

android:src="@drawable/next\_icon" />

<ImageButton

android:id="@+id/img\_btn\_setting"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="30dp"

android:backgroundTint="@color/colorPrimary"

android:src="@drawable/ic\_settings" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/ll\_player\_controls"

android:layout\_marginTop="10dp"

android:gravity="center\_horizontal"

android:orientation="horizontal">

<TextView

android:id="@+id/tv\_current\_time"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/start\_time"

android:textColor="@color/text\_color"

android:textSize="15sp"

android:textStyle="bold" />

<SeekBar

android:id="@+id/seekbar\_controller"

android:layout\_width="250dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:backgroundTint="@color/text\_color"

android:fitsSystemWindows="true"

android:progressTint="@color/text\_color"

android:thumbTint="@color/text\_color" />

<TextView

android:id="@+id/tv\_total\_time"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/final\_time"

android:textColor="@color/text\_color"

android:textSize="15sp"

android:textStyle="bold" />

</LinearLayout>

</RelativeLayout>

* **playlist\_items.xml**

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/text\_color"

android:padding="10dp">

<ImageView

android:id="@+id/iv\_music\_list"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:padding="5dp"

android:src="@drawable/ic\_music\_player" />

<TextView

android:id="@+id/tv\_music\_name"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginStart="10dp"

android:layout\_toEndOf="@id/iv\_music\_list"

android:singleLine="true"

android:text="Title"

android:textColor="@color/color\_black"

android:textSize="16sp"

android:textStyle="bold" />

<TextView

android:id="@+id/tv\_music\_subtitle"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@id/tv\_music\_name"

android:layout\_marginStart="10dp"

android:layout\_toEndOf="@id/iv\_music\_list"

android:singleLine="true"

android:text="SubTitle"

android:textColor="@color/gray\_color"

android:textSize="12sp"

android:textStyle="normal" />

</RelativeLayout>

* **colors.xml**

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

<resources>

<color name="colorPrimary">#45b5ec</color>

<color name="colorPrimaryDark">#1c8fc7</color>

<color name="colorAccent">#000000</color>

<color name="text\_color">#ffffff</color>

<color name="color\_black">#000000</color>

<color name="drawer\_color">#335763</color>

<color name="light\_color">#fafafa</color>

<color name="off\_color">#dfdada</color>

<color name="gray\_color">#9c9b9b</color>

</resources>

* **string.xml**

<resources>

<string name="app\_name">Nav Music</string>

<string name="title">Nav Music</string>

<string name="theme\_color">Theme Color</string>

<string name="sleep\_timer">Sleep Timer</string>

<string name="rate\_app">Rate This App</string>

<string name="about">About</string>

<string name="search">Search</string>

<string name="favorites">Favorites</string>

<string name="about\_text">Nav Music is an application to play mp3 music on your android device. With simple interface make you enjoy the music easily. Custom your color as you feel.</string>

<string name="ok">OK</string>

<string name="start\_time">0.00</string>

<string name="final\_time">0.00</string>

<string name="search\_hint">Song Name</string>

<string name="no">No</string>

<string name="yes">Yes</string>

<string name="delete">Delete</string>

<string name="delete\_text"> Are you sure you want to delete</string>

<string name="play\_next">Play Next</string>

</resources>

* **styles.xml**

<resources>

<!-- Base application theme. -->

<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">

<!-- Customize your theme here. -->

<item name="colorPrimary">@color/colorPrimary</item>

<item name="colorPrimaryDark">@color/colorPrimaryDark</item>

<item name="colorAccent">@color/colorAccent</item>

</style>

<style name="ActionBar"

parent="@style/Widget.AppCompat.Light.ActionBar">

<item name="android:background">@color/colorPrimary</item>

<item name="background">@color/colorAccent</item>

</style>

</resources>

* **AndroidMainfest.xml**

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

<manifest xmlns:android=http://schemas.android.com/apk/res/android package="com.example.soc\_macmini\_15.musicplayer">

<uses-permission android:name="android.permission.WAKE\_LOCK" />

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

<application

android:allowBackup="true"

android:icon="@drawable/ic\_music"

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

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/AppTheme">

<activity

android:name=".Activity.MainActivity"

android:launchMode="singleTop"

android:screenOrientation="portrait">

<intent-filter>

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

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

</intent-filter>

<meta-data

android:name="android.app.searchable"

android:resource="@xml/searchable" />

</activity>

</application>

</manifest>

* **MainActivity.java**

package com.example.soc\_macmini\_15.musicplayer.Activity;

import android.Manifest;

import android.app.SearchManager;

import android.content.Context;

import android.content.DialogInterface;

import android.content.pm.PackageManager;

import android.media.MediaPlayer;

import android.os.Handler;

import android.support.annotation.NonNull;

import android.support.design.widget.FloatingActionButton;

import android.support.design.widget.NavigationView;

import android.support.design.widget.Snackbar;

import android.support.design.widget.TabLayout;

import android.support.v4.app.ActivityCompat;

import android.support.v4.content.ContextCompat;

import android.support.v4.view.ViewPager;

import android.support.v4.widget.DrawerLayout;

import android.support.v7.app.ActionBar;

import android.support.v7.app.AlertDialog;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.Gravity;

import android.view.Menu;

import android.view.MenuItem;

import android.support.v7.widget.Toolbar;

import android.view.View;

import android.widget.ImageButton;

import android.widget.SearchView;

import android.widget.SeekBar;

import android.widget.TextView;

import android.widget.Toast;

import com.example.soc\_macmini\_15.musicplayer.Adapter.ViewPagerAdapter;

import com.example.soc\_macmini\_15.musicplayer.DB.FavoritesOperations;

import com.example.soc\_macmini\_15.musicplayer.Fragments.AllSongFragment;

import com.example.soc\_macmini\_15.musicplayer.Fragments.CurrentSongFragment;

import com.example.soc\_macmini\_15.musicplayer.Fragments.FavSongFragment;

import com.example.soc\_macmini\_15.musicplayer.Model.SongsList;

import com.example.soc\_macmini\_15.musicplayer.R;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity implements View.OnClickListener, AllSongFragment.createDataParse, FavSongFragment.createDataParsed, CurrentSongFragment.createDataParsed {

private Menu menu;

private ImageButton imgBtnPlayPause, imgbtnReplay, imgBtnPrev, imgBtnNext, imgBtnSetting;

private TabLayout tabLayout;

private ViewPager viewPager;

private SeekBar seekbarController;

private DrawerLayout mDrawerLayout;

private TextView tvCurrentTime, tvTotalTime;

private ArrayList<SongsList> songList;

private int currentPosition;

private String searchText = "";

private SongsList currSong;

private boolean checkFlag = false, repeatFlag = false, playContinueFlag = false, favFlag = true, playlistFlag = false;

private final int MY\_PERMISSION\_REQUEST = 100;

private int allSongLength;

MediaPlayer mediaPlayer;

Handler handler;

Runnable runnable;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

init();

grantedPermission();

}

/\* Initialising the views\*/

private void init() {

imgBtnPrev = findViewById(R.id.img\_btn\_previous);

imgBtnNext = findViewById(R.id.img\_btn\_next);

imgbtnReplay = findViewById(R.id.img\_btn\_replay);

imgBtnSetting = findViewById(R.id.img\_btn\_setting);

tvCurrentTime = findViewById(R.id.tv\_current\_time);

tvTotalTime = findViewById(R.id.tv\_total\_time);

FloatingActionButton refreshSongs = findViewById(R.id.btn\_refresh);

seekbarController = findViewById(R.id.seekbar\_controller);

viewPager = findViewById(R.id.songs\_viewpager);

NavigationView navigationView = findViewById(R.id.nav\_view);

mDrawerLayout = findViewById(R.id.drawer\_layout);

imgBtnPlayPause = findViewById(R.id.img\_btn\_play);

Toolbar toolbar = findViewById(R.id.toolbar);

handler = new Handler();

mediaPlayer = new MediaPlayer();

toolbar.setTitleTextColor(getResources().getColor(R.color.text\_color));

setSupportActionBar(toolbar);

ActionBar actionBar = getSupportActionBar();

assert actionBar != null;

actionBar.setDisplayHomeAsUpEnabled(true);

actionBar.setHomeAsUpIndicator(R.drawable.menu\_icon);

imgBtnNext.setOnClickListener(this);

imgBtnPrev.setOnClickListener(this);

imgbtnReplay.setOnClickListener(this);

refreshSongs.setOnClickListener(this);

imgBtnPlayPause.setOnClickListener(this);

imgBtnSetting.setOnClickListener(this);

navigationView.setNavigationItemSelectedListener(new NavigationView.OnNavigationItemSelectedListener() {

@Override

public boolean onNavigationItemSelected(@NonNull MenuItem item) {

item.setChecked(true);

mDrawerLayout.closeDrawers();

switch (item.getItemId()) {

case R.id.nav\_about:

about();

break;

}

return true;

}

});

}

/\* Function to ask user to grant the permission.\*/

private void grantedPermission() {

if(ContextCompat.checkSelfPermission(MainActivity.this,

Manifest.permission.READ\_EXTERNAL\_STORAGE)!= PackageManager.PERMISSION\_GRANTED) {

ActivityCompat.requestPermissions(MainActivity.this,new String[]{Manifest.permission.READ\_EXTERNAL\_STORAGE},MY\_PERMISSION\_REQUEST;

if(ActivityCompat.shouldShowRequestPermissionRationale(MainActivity.this,

Manifest.permission.READ\_EXTERNAL\_STORAGE)) {

ActivityCompat.requestPermissions(MainActivity.this,new String[]{Manifest.permission.READ\_EXTERNAL\_STORAGE},MY\_PERMISSION\_REQUEST;

} else{

if (ContextCompat.checkSelfPermission(MainActivity.this,

Manifest.permission.READ\_EXTERNAL\_STORAGE)!=PackageManager.PERMISSION\_GRATED){

Snackbar snackbar = Snackbar.make(mDrawerLayout, "Provide the Storage Permission", Snackbar.LENGTH\_LONG);

snackbar.show();

}

}

} else{

setPagerLayout();

}

}

/\*Checking if the permission is granted or not, @param requestCode ,@param permissions, @param grantResults\*/

@Override

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

switch (requestCode) {

case MY\_PERMISSION\_REQUEST:

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

if(ContextCompat.checkSelfPermission(MainActivity.this,

Manifest.permission.READ\_EXTERNAL\_STORAGE)==PackageManager.PERMISSION\_GRANTE){Toast.makeText(this, "Permission Granted!", Toast.LENGTH\_SHORT).show();

setPagerLayout();}

else{Snackbar snackbar = Snackbar.make(mDrawerLayout, "Provide the Storage Permission", Snackbar.LENGTH\_LONG);

snackbar.show();

finish();

}

}

}

}

/\*Setting up the tab layout with the viewpager in it.\*/

private void setPagerLayout(){

ViewPagerAdapter adapter = new ViewPagerAdapter(getSupportFragmentManager(), getContentResolver());

viewPager.setAdapter(adapter);

viewPager.addOnPageChangeListener(new ViewPager.OnPageChangeListener(){

@Override

public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels) {}

@Override

public void onPageSelected(int position) {}

@Override

public void onPageScrollStateChanged(int state) {}

});

tabLayout = findViewById(R.id.tabs);

tabLayout.setTabGravity(TabLayout.GRAVITY\_FILL);

tabLayout.setupWithViewPager(viewPager);

tabLayout.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener(){

@Override

public void onTabSelected(TabLayout.Tab tab) {

viewPager.setCurrentItem(tab.getPosition());}

@Override

public void onTabUnselected(TabLayout.Tab tab) {}

@Override

public void onTabReselected(TabLayout.Tab tab) {}

});

}

/\* Function to show the dialog for about us.\*/

private void about() {

AlertDialog.Builder builder = new AlertDialog.Builder(this);

builder.setTitle(getString(R.string.about))

.setMessage(getString(R.string.about\_text))

.setPositiveButton(R.string.ok, new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {}

});

AlertDialog alertDialog = builder.create();

alertDialog.show();

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

this.menu = menu;

getMenuInflater().inflate(R.menu.action\_bar\_menu, menu);

SearchManager manager = (SearchManager) getSystemService(Context.SEARCH\_SERVICE);

SearchView searchView = (SearchView) menu.findItem(R.id.menu\_search).getActionView();

searchView.setSearchableInfo(manager.getSearchableInfo(getComponentName());

searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {

@Override

public boolean onQueryTextSubmit(String query) {

return false;

}

@Override

public boolean onQueryTextChange(String newText) {

searchText = newText;

queryText();

setPagerLayout();

return true;

}

});

return super.onCreateOptionsMenu(menu);

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {

case android.R.id.home : mDrawerLayout.openDrawer(Gravity.START);

return true;

case R.id.menu\_search:

Toast.makeText(this, "Search", Toast.LENGTH\_SHORT).show();

return true;

case R.id.menu\_favorites:

if(checkFlag)

if(mediaPlayer!=null){

if(favFlag){

Toast.makeText(this, "Added to Favorites", Toast.LENGTH\_SHORT).show();

item.setIcon(R.drawable.ic\_favorite\_filled);

SongsList favList = new SongsList(songList.get(currentPosition).getTitle(),songList.get(currentPosition).getSubTitle(), songList.get(currentPosition).getPath());

FavoritesOperations favoritesOperations = new FavoritesOperations(this);

favoritesOperations.addSongFav(favList);

setPagerLayout();

favFlag = false;

} else{

item.setIcon(R.drawable.favorite\_icon);

favFlag = true;

}

}

return true;

}

return super.onOptionsItemSelected(item);

}

/\* Function to handle the click events,@param \*/

@Override

public void onClick(View v){

switch(v.getId()){

case R.id.img\_btn\_play:

if(checkFlag) {

if(mediaPlayer.isPlaying()) {

mediaPlayer.pause();

imgBtnPlayPause.setImageResource(R.drawable.play\_icon);

} else if(!mediaPlayer.isPlaying()){

mediaPlayer.start();

imgBtnPlayPause.setImageResource(R.drawable.pause\_icon);

playCycle();

}

} else{

Toast.makeText(this,"Select the Song ..", Toast.LENGTH\_SHORT).show();

}

break;

case R.id.btn\_refresh:

Toast.makeText(this,"Refreshing",Toast.LENGTH\_SHORT).show();

setPagerLayout();

break;

case R.id.img\_btn\_replay:

if(repeatFlag){

Toast.makeText(this,"Replaying Removed..", Toast.LENGTH\_SHORT).show();

mediaPlayer.setLooping(false);

repeatFlag = false;

} else{

Toast.makeText(this, "Replaying Added..", Toast.LENGTH\_SHORT).show();

mediaPlayer.setLooping(true);

repeatFlag = true;

}

break;

case R.id.img\_btn\_previous:

if (checkFlag){

if (mediaPlayer.getCurrentPosition() > 10) {

if (currentPosition - 1 > -1) {

attachMusic(songList.get(currentPosition - 1).getTitle(), songList.get(currentPosition - 1).getPath());

currentPosition = currentPosition - 1;

} else{

attachMusic(songList.get(currentPosition).getTitle(), songList.get(currentPosition).getPath());

}

} else{

attachMusic(songList.get(currentPosition).getTitle(), songList.get(currentPosition).getPath());

}

} else{

Toast.makeText(this, "Select a Song . .", Toast.LENGTH\_SHORT).show();

}

break;

case R.id.img\_btn\_next:

if(checkFlag){

if(currentPosition+1<songList.size()){attachMusic(songList.get(currentPosition + 1).getTitle(), songList.get(currentPosition + 1).getPath());

currentPosition += 1;

} else{

Toast.makeText(this, "Playlist Ended", Toast.LENGTH\_SHORT).show();

}

} else{

Toast.makeText(this, "Select the Song ..", Toast.LENGTH\_SHORT).show();

}

break;

case R.id.img\_btn\_setting:

if (!playContinuseFlag) {

playContinueFlag = true;

Toast.makeText(this, "Loop Added", Toast.LENGTH\_SHORT).show();

} else{

playContinueFlag = false;

Toast.makeText(this, "Loop Removed", Toast.LENGTH\_SHORT).show();

}

break;

}

}

/\* Function to attach the song to the music player, @param name, @param path\*/

private void attachMusic(String name, String path) {

imgBtnPlayPause.setImageResource(R.drawable.play\_icon);

setTitle(name);

menu.getItem(1).setIcon(R.drawable.favorite\_icon);

favFlag = true;

try {

mediaPlayer.reset();

mediaPlayer.setDataSource(path);

mediaPlayer.prepare();

setControls();

} catch (Exception e) {

e.printStackTrace();

}

mediaPlayer.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {

@Override

public void onCompletion(MediaPlayer mp) {

imgBtnPlayPause.setImageResource(R.drawable.play\_icon);

if (playContinueFlag) {

if (currentPosition + 1 < songList.size()) {

attachMusic(songList.get(currentPosition + 1).getTitle(), songList.get(currentPosition + 1).getPath());

currentPosition += 1;

} else {

Toast.makeText(MainActivity.this, "PlayList Ended", Toast.LENGTH\_SHORT).show();

}

}

}

});

}

/\* Function to set the controls according to the song\*/

private void setControls() {

seekbarController.setMax(mediaPlayer.getDuration());

mediaPlayer.start();

playCycle();

checkFlag = true;

if (mediaPlayer.isPlaying()) {

imgBtnPlayPause.setImageResource(R.drawable.pause\_icon);

tvTotalTime.setText(getTimeFormatted(mediaPlayer.getDuration()));

}

seekbarController.setOnSeekBarChangeListener(newSeekBar.OnSeekBarChangeListener() {

@Override

public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {

if (fromUser) {

mediaPlayer.seekTo(progress);

tvCurrentTime.setText(getTimeFormatted(progress));

}

}

@Override

public void onStartTrackingTouch(SeekBar seekBar) {}

@Override

public void onStopTrackingTouch(SeekBar seekBar) {}

});

}

/\* Function to play the song using a thread\*/

private void playCycle() {

try{ seekbarController.setProgress(mediaPlayer.getCurrentPosition());

tvCurrentTime.setText(getTimeFormatted(mediaPlayer.getCurrentPosition()));

if (mediaPlayer.isPlaying()) {

runnable = new Runnable() {

@Override

public void run() {

playCycle();

}

};

handler.postDelayed(runnable, 100);

}

} catch (Exception e) {

e.printStackTrace();

}

}

private String getTimeFormatted(long milliSeconds) {

String finalTimerString = "";

String secondsString;

//Converting total duration into time

int hours = (int) (milliSeconds / 3600000);

int minutes = (int) (milliSeconds % 3600000) / 60000;

int seconds = (int) ((milliSeconds % 3600000) % 60000 / 1000);

// Adding hours if any

if (hours > 0)

finalTimerString = hours + ":";

// Prepending 0 to seconds if it is one digit

if (seconds < 10)

secondsString = "0" + seconds;

else

secondsString = "" + seconds;

finalTimerString = finalTimerString + minutes + ":" + secondsString;

// Return timer String;

return finalTimerString;

}

/\*Function Overrided to receive the data from the fragment, @param name, @param path\*/

@Override

public void onDataPass(String name, String path) {

Toast.makeText(this, name, Toast.LENGTH\_LONG).show();

attachMusic(name, path);

}

@Override

public void getLength(int length) {

this.allSongLength = length;

}

@Override

public void fullSongList(ArrayList<SongsList> songList, int position)

{

this.songList = songList;

this.currentPosition = position;

this.playlistFlag = songList.size() == allSongLength;

this.playContinueFlag = !playlistFlag;

}

@Override

public String queryText() {

return searchText.toLowerCase();

}

@Override

public SongsList getSong() {

currentPosition = -1;

return currSong;

}

@Override

public boolean getPlaylistFlag() {

return playlistFlag;

}

@Override

public void currentSong(SongsList songsList) {

this.currSong = songsList;

}

@Override

public int getPosition() {

return currentPosition;

}

@Override

protected void onResume() {

super.onResume();

}

@Override

protected void onPause() {

super.onPause();

}

@Override

protected void onDestroy() {

super.onDestroy();

mediaPlayer.release();

handler.removeCallbacks(runnable);

}

}

* **SongAdapter.java**

package com.example.soc\_macmini\_15.musicplayer.Adapter;

import android.content.Context;

import android.graphics.Movie;

import android.support.annotation.NonNull;

import android.support.annotation.Nullable;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.ArrayAdapter;

import android.widget.Filterable;

import android.widget.TextView;

import com.example.soc\_macmini\_15.musicplayer.Model.SongsList;

import com.example.soc\_macmini\_15.musicplayer.R;

import java.util.ArrayList;

import java.util.List;

public class SongAdapter extends ArrayAdapter<SongsList> implements

Filterable{

private Context mContext;

private ArrayList<SongsList> songList = new ArrayList<>();

public SongAdapter(Context mContext, ArrayList<SongsList> songList) {

super(mContext, 0, songList);

this.mContext = mContext;

this.songList = songList;

}

@NonNull

@Override

public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {

View listItem = convertView;

if (listItem == null) {

listItem =

LayoutInflater.from(mContext).inflate(R.layout.playlist\_items,

parent, false);

}

SongsList currentSong = songList.get(position);

TextView tvTitle = listItem.findViewById(R.id.tv\_music\_name);

TextView tvSubtitle = listItem.findViewById(R.id.tv\_music\_subtitle);

tvTitle.setText(currentSong.getTitle());

tvSubtitle.setText(currentSong.getSubTitle());

return listItem;

}

}

* **ViewPagerAdapter.java**

package com.example.soc\_macmini\_15.musicplayer.Adapter;

import android.content.ContentResolver;

import android.support.annotation.Nullable;

import android.support.v4.app.Fragment;

import android.support.v4.app.FragmentManager;

import android.support.v4.app.FragmentPagerAdapter;

import com.example.soc\_macmini\_15.musicplayer.Fragments.AllSongFragment;

import com.example.soc\_macmini\_15.musicplayer.Fragments.CurrentSongFragm

nt;

import com.example.soc\_macmini\_15.musicplayer.Fragments.FavSongFragment;

public class ViewPagerAdapter extends FragmentPagerAdapter {

private ContentResolver contentResolver;

private String title[] = {"All SONGS","CURRENT PLAYLIST","FAVORITES"};

public ViewPagerAdapter(FragmentManager fm, ContentResolver

contentResolver){

super(fm);

this.contentResolver = contentResolver;

}

@Override

public Fragment getItem(int position) {

switch (position) {

case 0:

return AllSongFragment.getInstance(position,contentResolver);

case 1:

return CurrentSongFragment.getInstance(position);

case 2:

return FavSongFragment.getInstance(position);

default:

return null;

}

}

@Override

public int getCount() {

return title.length;

}

@Nullable

@Override

public CharSequence getPageTitle(int position) {

return title[position];

}

}

* **AllSongFragment**

package com.example.soc\_macmini\_15.musicplayer.Fragments;

import android.content.ContentResolver;

import android.content.Context;

import android.content.DialogInterface;

import android.database.Cursor;

import android.graphics.Color;

import android.net.Uri;

import android.os.Bundle;

import android.provider.MediaStore;

import android.support.annotation.NonNull;

import android.support.annotation.Nullable;

import android.support.v4.app.Fragment;

import android.support.v4.app.ListFragment;

import android.support.v7.app.AlertDialog;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.ListView;

import android.widget.Toast;

import com.example.soc\_macmini\_15.musicplayer.Adapter.SongAdapter;

import com.example.soc\_macmini\_15.musicplayer.Model.SongsList;

import com.example.soc\_macmini\_15.musicplayer.R;

import java.util.ArrayList;

public class AllSongFragment extends ListFragment {

private static ContentResolver contentResolver1;

public ArrayList<SongsList> songsList;

public ArrayList<SongsList> newList;

private ListView listView;

private createDataParse createDataParse;

private ContentResolver contentResolver;

public static Fragment getInstance(int position, ContentResolver

mcontentResolver) {

Bundle bundle = new Bundle();

bundle.putInt("pos", position);

AllSongFragment tabFragment = new AllSongFragment();

tabFragment.setArguments(bundle);

contentResolver1 = mcontentResolver;

return tabFragment;

}

@Override

public void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

}

@Override

public void onAttach(Context context) {

super.onAttach(context);

createDataParse = (createDataParse) context;

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container,

Bundle savedInstanceState) {

return inflater.inflate(R.layout.fragment\_tab, container, false);

}

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle

savedInstanceState) {

listView = view.findViewById(R.id.list\_playlist);

contentResolver = contentResolver1;

setContent();

}

/\*Setting the content in the listView and sending the data to the Activity\*/

public void setContent() {

boolean searchedList = false;

songsList = new ArrayList<>();

newList = new ArrayList<>();

getMusic();

SongAdapter adapter = new SongAdapter(getContext(), songsList);

if (!createDataParse.queryText().equals("")) {

adapter = onQueryTextChange();

adapter.notifyDataSetChanged();

searchedList = true;

} else {

searchedList = false;

}

createDataParse.getLength(songsList.size());

listView.setAdapter(adapter);

final boolean finalSearchedList = searchedList;

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> parent, View view, int position, long id){

// Toast.makeText(getContext(), "You clicked :\n" + songsList.get(position), Toast.LENGTH\_SHORT).show();

if (!finalSearchedList) {

createDataParse.onDataPass(songsList.get(position).getTitle(), songsList.get(position).getPath());

createDataParse.fullSongList(songsList, position);

} else {

createDataParse.onDataPass(newList.get(position).getTitle(),

newList.get(position).getPath());

createDataParse.fullSongList(songsList, position);

}

}

});

listView.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener() {

@Override

public boolean onItemLongClick(AdapterView<?> parent, View view, int position, long id) {

showDialog(position);

return true;

}

});

}

public void getMusic() {

Uri songUri = MediaStore.Audio.Media.EXTERNAL\_CONTENT\_URI;

Cursor songCursor = contentResolver.query(songUri, null, null, null, null);

if(songCursor != null && songCursor.moveToFirst()) {

int songTitle =

songCursor.getColumnIndex(MediaStore.Audio.Media.TITLE);

int songArtist =

songCursor.getColumnIndex(MediaStore.Audio.Media.ARTIST);

int songPath =

songCursor.getColumnIndex(MediaStore.Audio.Media.DATA);

do {

songsList.add(new SongsList(songCursor.getString(songTitle),

songCursor.getString(songArtist),

songCursor.getString(songPath)));

} while (songCursor.moveToNext());

songCursor.close();

}

}

public SongAdapter onQueryTextChange() {

String text = createDataParse.queryText();

for (SongsList songs : songsList) {

String title = songs.getTitle().toLowerCase();

if (title.contains(text)) {

newList.add(songs);

}

}

return new SongAdapter(getContext(), newList);

}

private void showDialog(final int position) {

AlertDialog.Builder builder = new AlertDialog.Builder(getContext());

builder.setMessage(getString(R.string.play\_next))

.setCancelable(true)

.setNegativeButton(R.string.no, new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {}

})

.setPositiveButton(R.string.yes, new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

createDataParse.currentSong(songsList.get(position));

setContent();

}

});

AlertDialog alertDialog = builder.create();

alertDialog.show();

}

public interface createDataParse {

public void onDataPass(String name, String path);

public void fullSongList(ArrayList<SongsList> songList, int position);

public String queryText();

public void currentSong(SongsList songsList);

public void getLength(int length);

}

}

* **CurrentSongFragment**

package com.example.soc\_macmini\_15.musicplayer.Fragments;

import android.content.ContentResolver;

import android.content.Context;

import android.content.DialogInterface;

import android.database.Cursor;

import android.net.Uri;

import android.os.Bundle;

import android.provider.MediaStore;

import android.support.annotation.NonNull;

import android.support.annotation.Nullable;

import android.support.v4.app.Fragment;

import android.support.v4.app.ListFragment;

import android.support.v7.app.AlertDialog;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.AdapterView;

import android.widget.ListView;

import android.widget.Toast;

import com.example.soc\_macmini\_15.musicplayer.Adapter.SongAdapter;

import com.example.soc\_macmini\_15.musicplayer.DB.FavoritesOperations;

import com.example.soc\_macmini\_15.musicplayer.Model.SongsList;

import com.example.soc\_macmini\_15.musicplayer.R;

import java.util.ArrayList;

public class CurrentSongFragment extends ListFragment {

public ArrayList<SongsList> songsList = new ArrayList<>();

private ListView listView;

private createDataParsed createDataParsed;

public static Fragment getInstance(int position) {

Bundle bundle = new Bundle();

bundle.putInt("pos", position);

CurrentSongFragment tabFragment = new CurrentSongFragment();

tabFragment.setArguments(bundle);

return tabFragment;

}

@Override

public void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

}

@Override

public void onAttach(Context context) {

super.onAttach(context);

createDataParsed = (createDataParsed) context;

}

@Override

public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {

return inflater.inflate(R.layout.fragment\_tab, container, false);

}

@Override

public void onViewCreated(@NonNull View view, @Nullable Bundle

savedInstanceState) {

listView = view.findViewById(R.id.list\_playlist);

//songsList = new ArrayList<>();

setContent();

}

/\* Setting the content in the listView and sending the data to the Activity\*/

public void setContent() {

if (createDataParsed.getSong() != null)

songsList.add(createDataParsed.getSong());

SongAdapter adapter = new SongAdapter(getContext(), songsList);

if (songsList.size() > 1)

if (createDataParsed.getPlaylistFlag()) {

songsList.clear();

}

listView.setAdapter(adapter);

adapter.notifyDataSetChanged();

listView.setOnItemClickListener(new AdapterView.OnItemClickListener(){

@Override

public void onItemClick(AdapterView<?>parent,View view,int position,long id){

// Toast.makeText(getContext(), "You clicked :\n" +

songsList.get(position), Toast.LENGTH\_SHORT).show();

createDataParsed.onDataPass(songsList.get(position).getTitle(),

songsList.get(position).getPath());

createDataParsed.fullSongList(songsList, position);

}

});

listView.setOnItemLongClickListener(new

AdapterView.OnItemLongClickListener() {

@Override

public boolean onItemLongClick(AdapterView<?> parent,View view,int position,long id) {

return true;

}

});

}

public interface createDataParsed {

public void onDataPass(String name, String path);

public void fullSongList(ArrayList<SongsList> songList, int position);

public SongsList getSong();

public boolean getPlaylistFlag();

}

}

* **SongsList**

package com.example.soc\_macmini\_15.musicplayer.Model;

public class SongsList {

private String title;

private String subTitle;

private String path;

public String getPath() {

return path;

}

public void setPath(String path) {

this.path = path;

}

public SongsList(String title, String subTitle, String path) {

this.title = title;

this.subTitle = subTitle;

this.path = path;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

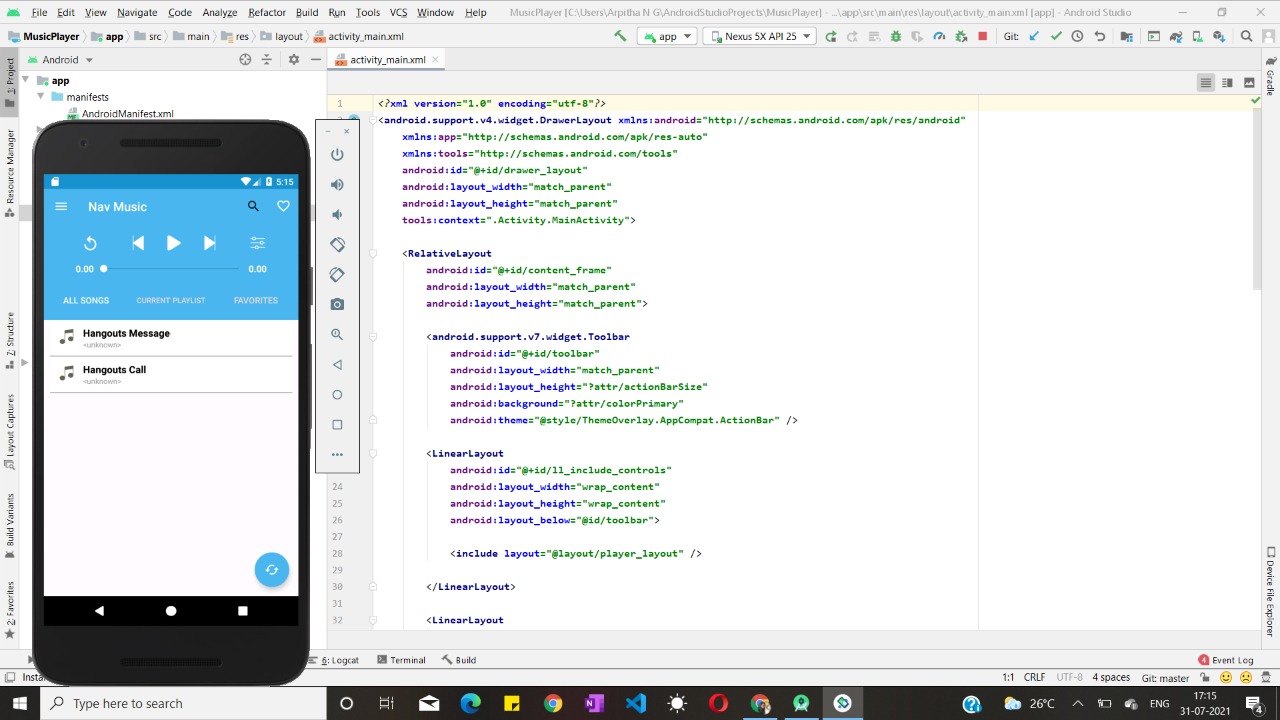
public String getSubTitle() {

return subTitle;

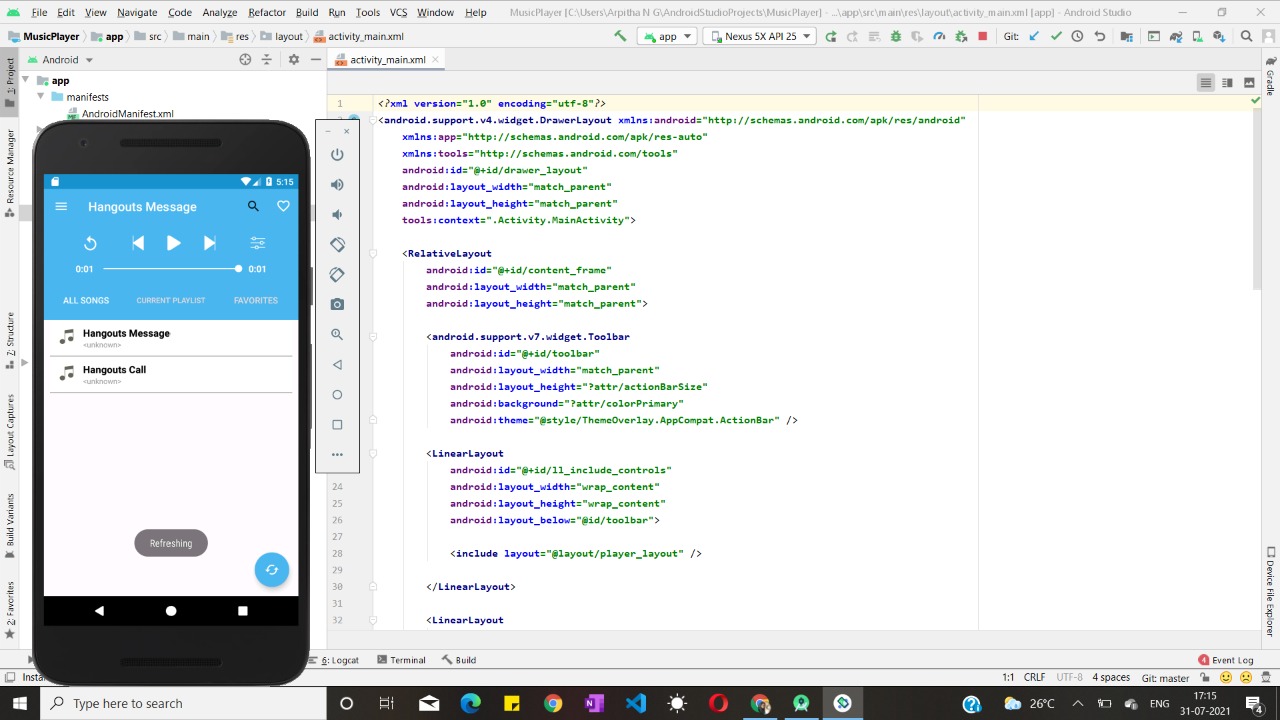
}

}

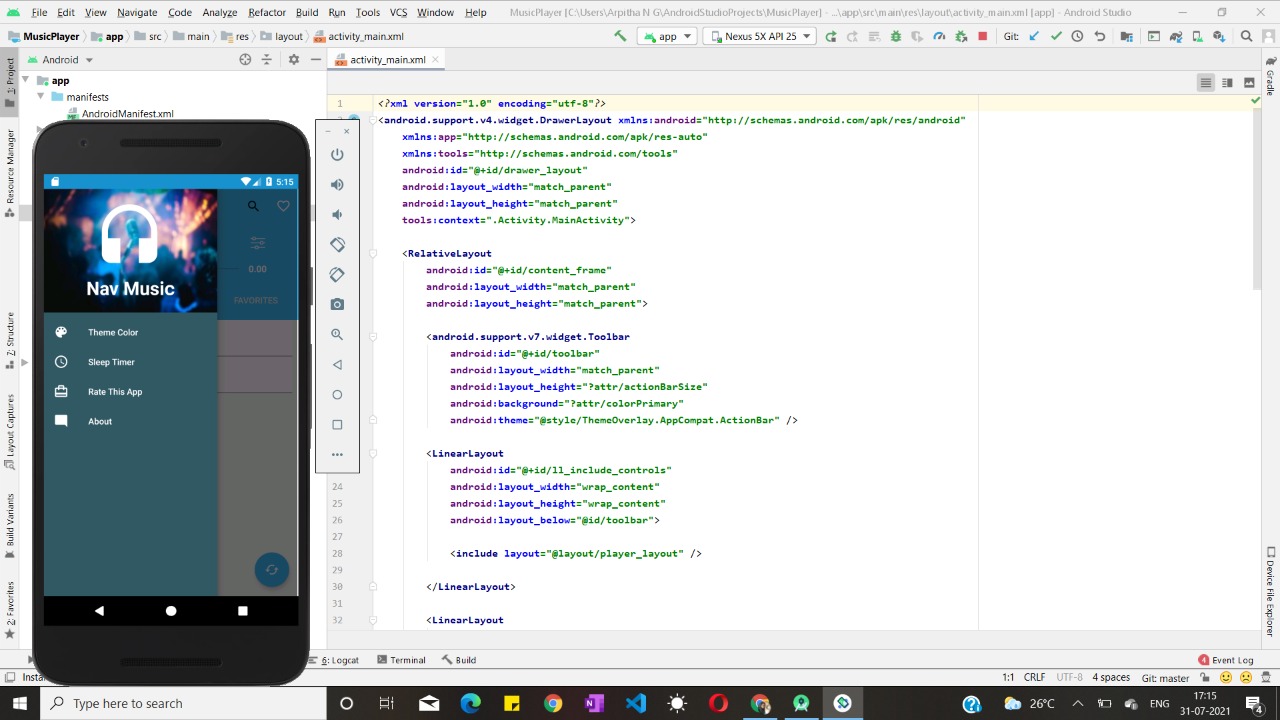
**RESULTS**

****

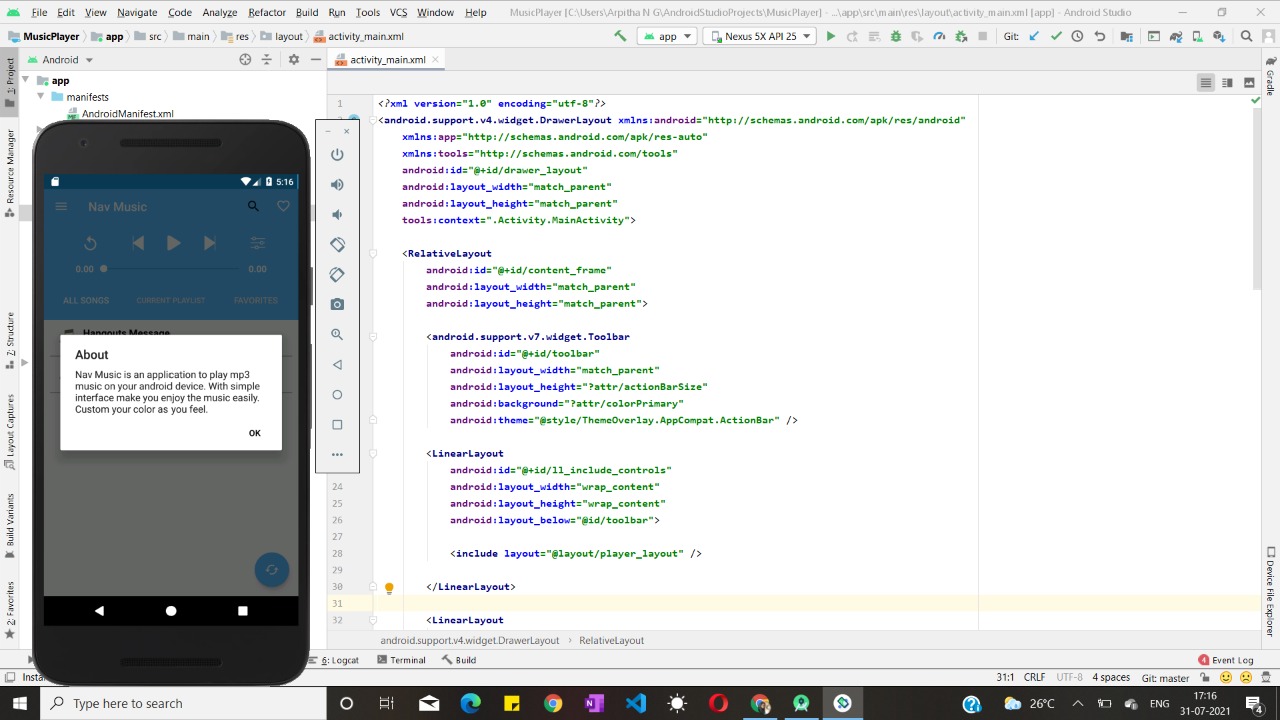
***Snapshot 1: PlayList***

****

***Snapshot 2 : Refreshing***

****

***Snapshot 3 : Dashboard***

****

***Snapshot 4 : About pop-up***