

Topic : Develop an application of music player .

Code :

⇒ **Activity_main.xml**

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

<ScrollView

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

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:background="#000000"

    android:fillViewport="true">

    <LinearLayout

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:orientation="vertical"

        android:padding="16dp">

        <TextView

            android:layout_width="wrap_content"

            android:layout_height="wrap_content"

            android:text="Song Categories"

            android:textSize="22sp"

            android:textColor="#FFFFFF"

            android:textStyle="bold"

            android:paddingBottom="10dp" />

        <ExpandableListView

            android:id="@+id/lvSongs"

            android:layout_width="wrap_content"

            android:layout_height="456dp"

            android:background="#1E1E1E"

            android:divider="@android:color/darker_gray"

            android:dividerHeight="1dp"

            android:padding="10dp"

            android:scrollbars="vertical" />
```

<LinearLayout

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center_horizontal"
    android:orientation="vertical"
    android:padding="16dp">
```

<TextView

```
    android:id="@+id/tvSongTitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:gravity="center_horizontal"
    android:text="Select a song to play"
    android:textColor="#FFFFFF"
    android:textSize="20sp"
    android:textStyle="bold" />
```

<SeekBar

```
    android:id="@+id/seekBar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:progressTint="#FF4081"
    android:thumbTint="#FF4081" />
```

<TextView

```
    android:id="@+id/tvCurrentTime"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="8dp"
    android:text="0:00"
    android:textColor="#CCCCCC"
    android:textSize="12sp" />
```

```
<LinearLayout

    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="horizontal"
    android:padding="8dp">

    <Button

        android:id="@+id/btnPlay"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:backgroundTint="#FFEB3B"
        android:text="▶ Play"
        android:textColor="#000000" />

    <Button

        android:id="@+id/btnPause"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:backgroundTint="#3F51B5"
        android:text="⏸ Pause"
        android:textColor="#FFFFFF" />

</LinearLayout>
```

```
<LinearLayout

    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="horizontal"
    android:padding="8dp">

    <Button

        android:id="@+id/btnStop"
```

```
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:backgroundTint="#F44336"
        android:text="■ Stop"
        android:textColor="#FFFFFF" />
```

```
<Button
    android:id="@+id/btnNext"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:backgroundTint="#FF9800"
    android:text="▶ Next"
    android:onClick="playNextSong"
    android:textColor="#FFFFFF" />
```

```
</LinearLayout>
```

```
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="horizontal">
```

```
<View
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1" />
```

```
<Button
    android:id="@+id/btnShuffle"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
```

```

        android:layout_weight="1"

        android:backgroundTint="#4CAF50"

        android:text="🔀 Shuffle"

        android:onClick="toggleShuffle"

        android:textColor="#FFFFFF" />

    <View

        android:layout_width="0dp"

        android:layout_height="wrap_content"

        android:layout_weight="1" />

    </LinearLayout>

</LinearLayout>

</ScrollView>

```

⇒ MainActivity.java

```

package com.example.music_player;

import android.media.MediaPlayer;
import android.os.Bundle;
import android.os.Handler;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ExpandableListView;
import android.widget.SeekBar;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashMap;
import java.util.List;

public class MainActivity extends AppCompatActivity {
    private static final String TAG = "MainActivity";
    private ExpandableListView lvSongs;
    private TextView tvSongTitle, tvCurrentTime;
    private SeekBar seekBar;
    private Button btnPlay, btnPause, btnStop, btnShuffle, btnNext;
    private MediaPlayer mediaPlayer;
    private Handler handler = new Handler();
    private List<String> categories;
    private HashMap<String, List<String>> songList;
    private String currentSong = "";
    private boolean isShuffle = false;
    private List<String> shuffledSongs = new ArrayList<>();
    private int songPosition = 0; // Add a variable to store the current song position
    private boolean isSeekBarTracking = false; // Add a variable to track SeekBar changes

```

```

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

    lvSongs = findViewById(R.id.lvSongs);
    tvSongTitle = findViewById(R.id.tvSongTitle);
    tvCurrentTime = findViewById(R.id.tvCurrentTime);
    seekBar = findViewById(R.id.seekBar);
    btnPlay = findViewById(R.id.btnPlay);
    btnPause = findViewById(R.id.btnPause);
    btnStop = findViewById(R.id.btnStop);
    btnShuffle = findViewById(R.id.btnShuffle);
    btnNext = findViewById(R.id.btnNext);

    initializeSongs();
    setupExpandableListView();
    setupSeekBarListener(); // Add method to set up SeekBar listener

    btnPlay.setOnClickListener(v -> playSong());
    btnPause.setOnClickListener(v -> pauseSong());
    btnStop.setOnClickListener(v -> stopSong());
    btnShuffle.setOnClickListener(v -> toggleShuffle());
    btnNext.setOnClickListener(v -> playNextSong());
}

private void initializeSongs() {
    categories = new ArrayList<>();
    songList = new HashMap<>();

    categories.add("Marathi Songs");
    categories.add("English Songs");
    categories.add("Hindi Songs");

    List<String> marathiSongs = new ArrayList<>();
    marathiSongs.add("kaakan_marathi");
    marathiSongs.add("kadhi_tu_marathi");
    marathiSongs.add("kitidya_navane_marathi");
    marathiSongs.add("mla_ved_lagale");
    marathiSongs.add("saaj_hyo_tuza_marathi");
    marathiSongs.add("tik_tik_vajte_marathi");
    marathiSongs.add("tola_tola_marathi");

    List<String> englishSongs = new ArrayList<>();
    englishSongs.add("die_with_a_smile");
    englishSongs.add("night_changes");
    englishSongs.add("what_makes_you_beautiful");
    englishSongs.add("perfect");
    englishSongs.add("until_i_found_you");
    englishSongs.add("somebody_to_you");

    List<String> hindiSongs = new ArrayList<>();
    hindiSongs.add("mayya_mainu");
    hindiSongs.add("pehla_pyaar");
    hindiSongs.add("pehli_dafa");

```

```

hindiSongs.add("o_saathi");
hindiSongs.add("all_do_is_think_about_you");
hindiSongs.add("humdard");
hindiSongs.add("labon_pe_naam");
hindiSongs.add("sukoon_mila");
hindiSongs.add("gazab_ka_hai_din");

songList.put(categories.get(0), marathiSongs);
songList.put(categories.get(1), englishSongs);
songList.put(categories.get(2), hindiSongs);

// Flatten song list for shuffle mode
shuffledSongs.addAll(marathiSongs);
shuffledSongs.addAll(englishSongs);
shuffledSongs.addAll(hindiSongs);
}

private void setupExpandableListView() {
    ExpandableListAdapter adapter = new ExpandableListAdapter(this, categories, songList);
    lvSongs.setAdapter(adapter);

    lvSongs.setOnChildClickListener((parent, v, groupPosition, childPosition, id) -> {
        currentSong = songList.get(categories.get(groupPosition)).get(childPosition);
        tvSongTitle.setText(currentSong);
        playSong();
        return true;
    });
}

private void setupSeekBarListener() {
    seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
        @Override
        public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
            if (fromUser && mediaPlayer != null) {
                mediaPlayer.seekTo(progress);
            }
        }

        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {
            isSeekBarTracking = true;
        }

        @Override
        public void onStopTrackingTouch(SeekBar seekBar) {
            isSeekBarTracking = false;
            if (mediaPlayer != null) {
                mediaPlayer.seekTo(seekBar.getProgress());
            }
        }
    });
}

private void playSong() {
    if (mediaPlayer != null) {
        mediaPlayer.release();
    }
}

```

```

    }

    int songResId = getResources().getIdentifier(currentSong, "raw", getPackageName());
    Log.d(TAG, "Playing song: " + currentSong + " with resource ID: " + songResId);

    if (songResId == 0) {
        Log.e(TAG, "Error: Song not found - " + currentSong);
        tvSongTitle.setText("Error: Song not found");
        return;
    }

    mediaPlayer = MediaPlayer.create(this, songResId);
    mediaPlayer.seekTo(songPosition); // Resume from the current position
    mediaPlayer.start();

    seekBar.setMax(mediaPlayer.getDuration());
    handler.postDelayed(updateSeekBar, 1000);
}

private void pauseSong() {
    if (mediaPlayer != null && mediaPlayer.isPlaying()) {
        songPosition = mediaPlayer.getCurrentPosition(); // Save the current position
        mediaPlayer.pause();
    }
}

private void stopSong() {
    if (mediaPlayer != null) {
        mediaPlayer.stop();
        mediaPlayer.release();
        mediaPlayer = null;
        seekBar.setProgress(0);
        tvCurrentTime.setText("0:00");
        songPosition = 0; // Reset the position
    }
}

private void toggleShuffle() {
    isShuffle = !isShuffle;
    btnShuffle.setText(isShuffle ? "Shuffle: ON" : "Shuffle: OFF");

    if (isShuffle) {
        Collections.shuffle(shuffledSongs);

        // Ensure the new song is different from the current song
        String newSong;
        do {
            newSong = shuffledSongs.get(0);
        } while (newSong.equals(currentSong) && shuffledSongs.size() > 1);

        currentSong = newSong;
        tvSongTitle.setText(currentSong);
        playSong();
    }
}

```



```

private List<String> getAllSongs() {
    List<String> allSongs = new ArrayList<>();
    if (songList != null && !songList.isEmpty()) {
        for (List<String> songs : songList.values()) {
            allSongs.addAll(songs);
        }
    }
    return allSongs;
}

private void playNextSong() {
    List<String> currentList = isShuffle ? shuffledSongs : getAllSongs();

    // Prevent crash: Ensure the list is not empty
    if (currentList == null || currentList.isEmpty()) {
        tvSongTitle.setText("No songs available");
        return;
    }

    // Find the current song index
    int currentIndex = currentList.indexOf(currentSong);

    // Prevent crash: Ensure currentSong is valid
    if (currentIndex == -1) {
        currentSong = currentList.get(0); // Start with the first song if not found
    } else if (currentIndex >= currentList.size() - 1) {
        currentSong = currentList.get(0); // Loop back to the first song
    } else {
        currentSong = currentList.get(currentIndex + 1); // Move to the next song
    }

    tvSongTitle.setText(currentSong);
    playSong();
}

private Runnable updateSeekBar = new Runnable() {
    @Override
    public void run() {
        if (mediaPlayer != null && !isSeekBarTracking) { // Check if SeekBar is being tracked
            seekBar.setProgress(mediaPlayer.getCurrentPosition());
            tvCurrentTime.setText(formatTime(mediaPlayer.getCurrentPosition()));
            handler.postDelayed(this, 1000);
        }
    }
};

private String formatTime(int millis) {
    int minutes = (millis / 1000) / 60;
    int seconds = (millis / 1000) % 60;
    return String.format("%d:%02d", minutes, seconds);
}

@Override
protected void onDestroy() {
    if (mediaPlayer != null) {
        mediaPlayer.release();
    }
}

```

```

        mediaPlayer = null;
    }
    super.onDestroy();
}
}

```

⇒ **AndroidManifest.xml**

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <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:supportRtl="true"
        android:theme="@style/Theme.Music_Player"
        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>

```