



<b>BATCH AND ROLL NO:</b> Q5 42441
<b>EXPERIMENT NO.8</b>
<b>TITLE:</b> Design a mobile application for media player.
<b>DATE OF PERFORMANCE:</b>
<b>DATE OF SUBMISSION:</b>

**Title:** Design a mobile application for media player.

**Requirements:**

1 Android studio

**Theory:**

**Introduction**

In the realm of mobile application development, creating a media player application provides an avenue to deliver immersive and interactive experiences for users. This lab focuses on the design and implementation of a mobile media player application, empowering users to enjoy audio or video content seamlessly. The integration of a media player not only enhances the entertainment aspect of an application but also showcases the utilization of multimedia capabilities in modern mobile devices.

**Objective of the Lab:** The primary objective of this lab is to guide you through the process of designing a mobile application equipped with a media player component. By the end of this lab, you should be proficient in implementing features such as playing, pausing, and controlling media playback. Additionally, you will explore aspects like handling media files, implementing user controls, and providing a seamless and engaging media playback experience.

**Components of the Application:**

**1. Media Player Component:**

- The media player component serves as the core element responsible for handling and controlling audio or video playback.
- It includes functionalities such as play, pause, stop, forward, and rewind, contributing to a user-friendly and feature-rich media experience.

**Lab Prerequisites:**

- Basic understanding of mobile application development concepts.
- Familiarity with the chosen development environment (e.g., Android Studio, Xcode).
- Prior knowledge of programming languages such as Java or Kotlin (for Android) or Swift (for iOS).



**Steps:**

**Step 1: Set Up Your Development Environment**

- Ensure that you have Android Studio installed and configured on your machine.

**Step 2: Create a New Project**

- Open Android Studio and create a new project.
- Choose an appropriate project template, such as "Empty Activity" or "Basic Activity."

**Step 3: Design the Main Activity Layout**

- Open the XML layout file associated with your main activity (e.g., activity\_main.xml).
- Design the layout with relevant UI elements, such as buttons for play, pause, stop, and a SeekBar for progress tracking.

**Step 4: Implement the Java Code**

- Open the Java file associated with your main activity (e.g., MainActivity.java )
- Implement the logic for initializing the media player, handling button clicks, and updating the SeekBar.

**Step 5: Test Your Application**

- Run your application on an emulator or a physical device.
- Verify that the media player buttons function correctly, and the SeekBar updates as the media plays.



**XML Code:**

activity\_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#121212"
    tools:context=".MainActivity">

    <ImageView
        android:id="@+id/albumArtImageView"
        android:layout_width="300dp"
        android:layout_height="300dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="50dp"
        android:layout_marginVertical="120dp"
        android:scaleType="centerCrop"/>

    <TextView
        android:id="@+id/songTitleTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/albumArtImageView"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20dp"
        android:textColor="#FFFFFF"
        android:textSize="20sp"
        android:textStyle="bold"/>

    <TextView
        android:id="@+id/artistNameTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/songTitleTextView"
        android:layout_centerHorizontal="true"
        android:textColor="#B0B0B0"
        android:textSize="16sp"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:orientation="vertical"
        android:layout_marginBottom="50dp">

        <SeekBar
            android:id="@+id/songProgressSeekBar"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:progressBackgroundTint="#FFFFFF"
            android:layout_marginHorizontal="30dp"
            android:layout_marginBottom="20dp"/>

    </LinearLayout>

</RelativeLayout>
```



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

    <ImageButton
        android:id="@+id/previousButton"
        android:layout_width="60dp"
        android:layout_height="60dp"
        android:background="?attr/selectableItemBackgroundBorderless"
        android:src="@drawable/ic_previous"
        android:layout_marginEnd="30dp"/>

    <ImageButton
        android:id="@+id/playPauseButton"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:background="?attr/selectableItemBackgroundBorderless"
        android:src="@drawable/ic_play"/>

    <ImageButton
        android:id="@+id/nextButton"
        android:layout_width="60dp"
        android:layout_height="60dp"
        android:background="?attr/selectableItemBackgroundBorderless"
        android:src="@drawable/ic_next"
        android:layout_marginStart="30dp"/>
</LinearLayout>
</LinearLayout>
</RelativeLayout>
```

default\_album\_art.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="300dp"
    android:height="300dp"
    android:viewportWidth="100"
    android:viewportHeight="100">
    <path
        android:fillColor="#808080"
        android:pathData="M0,0 L100,0 L100,100 L0,100 Z"/>
</vector>
```



ic\_play.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:viewportWidth="24"
    android:viewportHeight="24">
    <path
        android:fillColor="#FFFFFF"
        android:pathData="M8,5v14l11,-7z"/>
</vector>
```

ic\_pause.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:viewportWidth="24"
    android:viewportHeight="24">
    <path
        android:fillColor="#FFFFFF"
        android:pathData="M6,19h4L10,5L6,5v14zM14,5v14h4L18,5h-4z"/>
</vector>
```

ic\_next.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:viewportWidth="24"
    android:viewportHeight="24">
    <path
        android:fillColor="#FFFFFF"
        android:pathData="M10,6L8.59,7.41 13.17,12l-4.58,4.59L10,18l6,-6z"/>
</vector>
```



ic\_previous.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:viewportWidth="24"
    android:viewportHeight="24">
    <path
        android:fillColor="#FFFFFF"
        android:pathData="M15.41,7.41L14,6l-6,6 6,6 1.41,-1.41L10.83,12z"/>
</vector>
```

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.expt08_42441">

    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.MediaPlayer">

        <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>
```



### **Java Code:**

MainActivity.java:

```
package com.example.expt08_42441;

import android.media.MediaPlayer;
import android.os.Bundle;
import android.os.Handler;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.SeekBar;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import java.util.ArrayList;
import java.util.List;

public class MainActivity extends AppCompatActivity {
    private MediaPlayer mediaPlayer;
    private ImageButton playPauseButton, nextButton, previousButton;
    private SeekBar songProgressSeekBar;
    private TextView songTitleTextView, artistNameTextView;
    private ImageView albumArtImageView;
    private List<Song> playlist;
    private int currentSongIndex = 0;
    private Handler handler = new Handler();

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

        // Initialize views
        playPauseButton = findViewById(R.id.playPauseButton);
        nextButton = findViewById(R.id.nextButton);
        previousButton = findViewById(R.id.previousButton);
        songProgressSeekBar = findViewById(R.id.songProgressSeekBar);
        songTitleTextView = findViewById(R.id.songTitleTextView);
        artistNameTextView = findViewById(R.id.artistNameTextView);
        albumArtImageView = findViewById(R.id.albumArtImageView);

        // Setup playlist
        playlist = new ArrayList<>();
        playlist.add(new Song("Perfect", "Ed Sheeran", R.raw.perfect_ed_sheeran, R.drawable.image1));
        playlist.add(new Song("Yellow", "Coldplay", R.raw.yellow_coldplay, R.drawable.image2));

        // Initial song setup
        setupSong(playlist.get(currentSongIndex));
```



```
// Play/Pause listener
playPauseButton.setOnClickListener(v -> {
    if (mediaPlayer.isPlaying()) {
        mediaPlayer.pause();
        playPauseButton.setImageResource(R.drawable.ic_play);
    } else {
        mediaPlayer.start();
        playPauseButton.setImageResource(R.drawable.ic_pause);
    }
});

// Next song listener
nextButton.setOnClickListener(v -> {
    mediaPlayer.stop();
    currentSongIndex = (currentSongIndex + 1) % playlist.size();
    setupSong(playlist.get(currentSongIndex));
});

// Previous song listener
previousButton.setOnClickListener(v -> {
    mediaPlayer.stop();
    currentSongIndex = (currentSongIndex - 1 + playlist.size()) % playlist.size();
    setupSong(playlist.get(currentSongIndex));
});
}

private void setupSong(Song song) {
    // Release previous MediaPlayer
    if (mediaPlayer != null) {
        mediaPlayer.release();
    }

    // Create new MediaPlayer
    mediaPlayer = MediaPlayer.create(this, song.getResourceId());

    // Update UI
    songTitleTextView.setText(song.getTitle());
    artistNameTextView.setText(song.getArtist());
    albumArtImageView.setImageResource(song.getCoverResourceId());

    // Setup SeekBar
    songProgressSeekBar.setMax(mediaPlayer.getDuration());
    songProgressSeekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
        @Override
        public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
            if (fromUser) {
                mediaPlayer.seekTo(progress);
            }
        }
        @Override
        public void onStartTrackingTouch(SeekBar seekBar) {}
        @Override
        public void onStopTrackingTouch(SeekBar seekBar) {}
    });
};
```





```
// Start playback
mediaPlayer.start();
playPauseButton.setImageResource(R.drawable.ic_pause);

// Update SeekBar
updateSeekBar();
}

private void updateSeekBar() {
    handler.postDelayed(new Runnable() {
        @Override
        public void run() {
            if (mediaPlayer != null) {
                songProgressSeekBar.setProgress(mediaPlayer.getCurrentPosition());
            }
            handler.postDelayed(this, 1000);
        }
    }, 1000);
}

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

**Song.java:**

```
package com.example.expt08_42441;

public class Song {
    private String title;
    private String artist;
    private int resourceId;
    private int coverResourceId;

    public Song(String title, String artist, int resourceId, int coverResourceId) {
        this.title = title;
        this.artist = artist;
        this.resourceId = resourceId;
        this.coverResourceId = coverResourceId;
    }

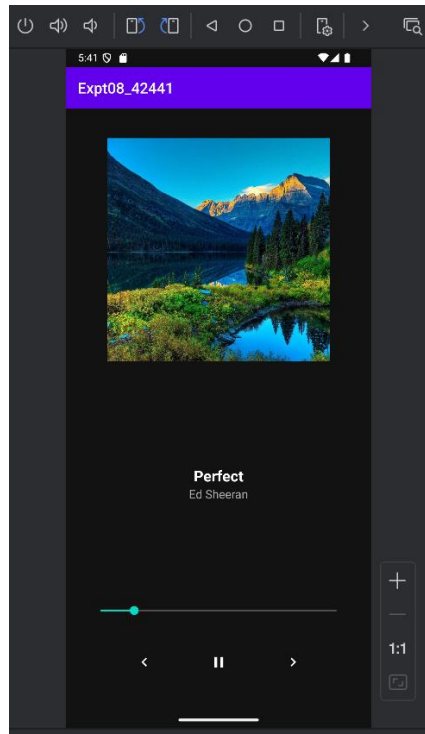
    public String getTitle() { return title; }
    public String getArtist() { return artist; }
    public int getResourceId() { return resourceId; }
    public int getCoverResourceId() { return coverResourceId; }
}
```



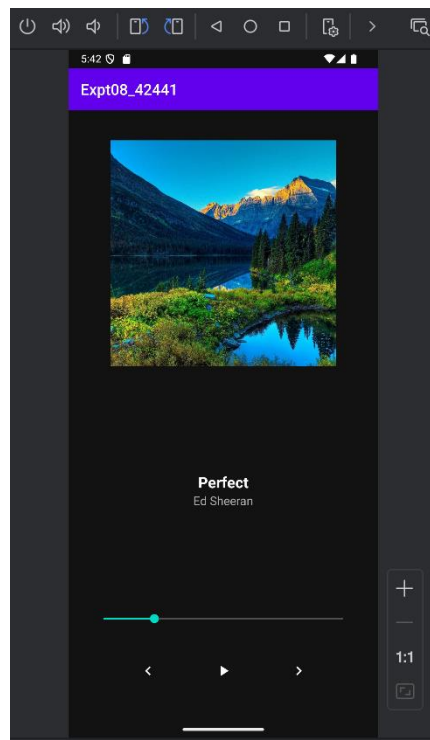
**PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE – 411043**  
**Department of Electronics & Telecommunication Engineering**

**Output:**

1) Play:



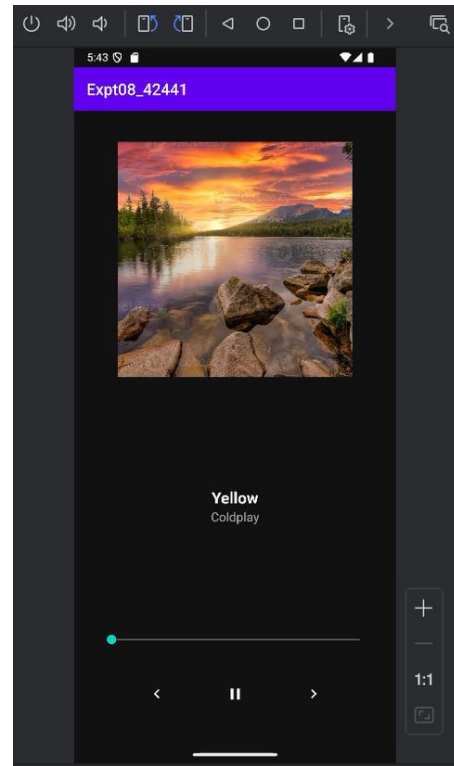
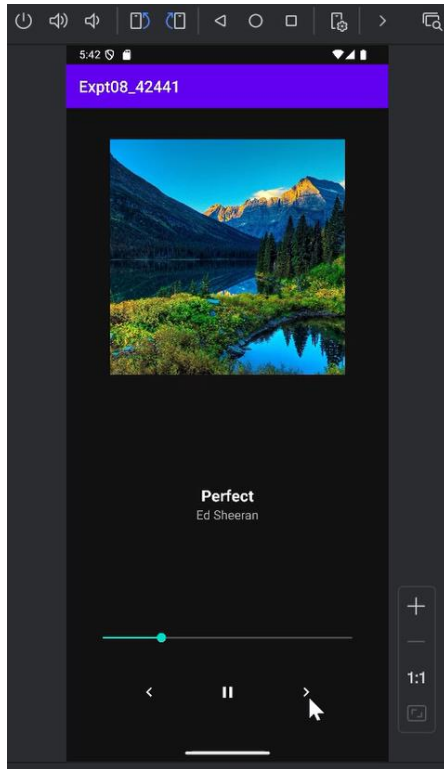
2) Pause:



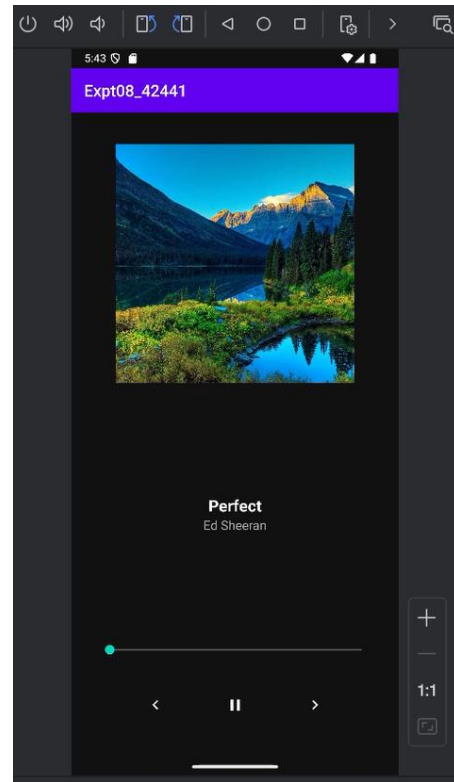
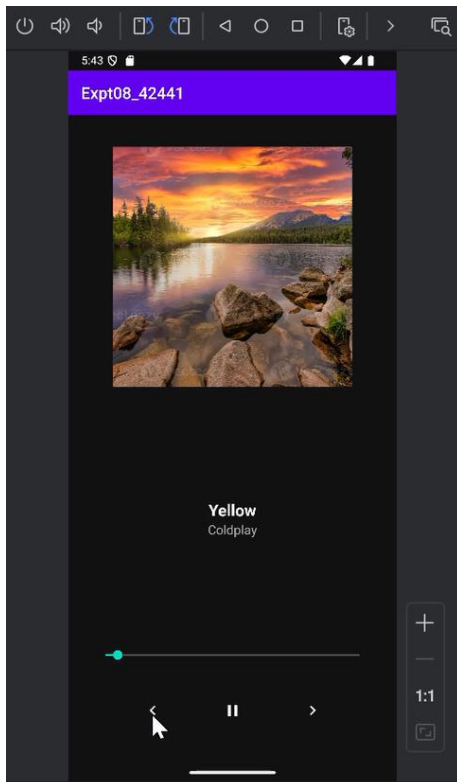


**PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE – 411043**  
**Department of Electronics & Telecommunication Engineering**

3) Forward:



4) Backward:





**PUNE INSTITUTE OF COMPUTER TECHNOLOGY, PUNE – 411043**  
**Department of Electronics & Telecommunication Engineering**

**Conclusion:**

.....

.....

.....