|  |
| --- |
| **BATCH AND ROLL NO:** |
| **EXPERIMENT NO.8** |
| **TITLE:** Design a mobile application for media player. |
| **DATE OF PERFORMANCE:** |
| **DATE OF SUBMISSION:** |

**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:**

**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">

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

android:maxSdkVersion="32" />

<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.MediaPlayer"

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>

**activity\_main.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:padding="16dp">

<!-- Title -->

<TextView

android:id="@+id/title"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Sample Media Player"

android:gravity="center"

android:textSize="24sp"

android:textStyle="bold"

/>

<!-- Play Button -->

<Button

android:id="@+id/playButton"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Play" />

<!-- Pause Button -->

<Button

android:id="@+id/pauseButton"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Pause"

android:layout\_marginTop="10dp" />

<!-- Stop Button -->

<Button

android:id="@+id/stopButton"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Stop"

android:layout\_marginTop="10dp" />

</LinearLayout>

**Java Code:**

package com.example.mediaplayer;

import androidx.appcompat.app.AppCompatActivity;

import android.media.MediaPlayer;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

private MediaPlayer mediaPlayer;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

// Initialize Buttons

Button playButton = findViewById(R.id.playButton);

Button pauseButton = findViewById(R.id.pauseButton);

Button stopButton = findViewById(R.id.stopButton);

// Initialize MediaPlayer with a sample audio file from res/raw

mediaPlayer = MediaPlayer.create(this, R.raw.sample\_audio);

// Play Button Listener

playButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if (!mediaPlayer.isPlaying()) {

mediaPlayer.start();

Toast.makeText(MainActivity.this, "Playing Audio", Toast.LENGTH\_SHORT).show();

}

}

});

// Pause Button Listener

pauseButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if (mediaPlayer.isPlaying()) {

mediaPlayer.pause();

Toast.makeText(MainActivity.this, "Audio Paused", Toast.LENGTH\_SHORT).show();

}

}

});

// Stop Button Listener

stopButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if (mediaPlayer.isPlaying() || mediaPlayer.getCurrentPosition() > 0) {

mediaPlayer.stop();

mediaPlayer = MediaPlayer.create(MainActivity.this, R.raw.sample\_audio); // Reset MediaPlayer

Toast.makeText(MainActivity.this, "Audio Stopped", Toast.LENGTH\_SHORT).show();

}

}

});

}

@Override

protected void onDestroy() {

super.onDestroy();

if (mediaPlayer != null) {

mediaPlayer.release(); // Release resources

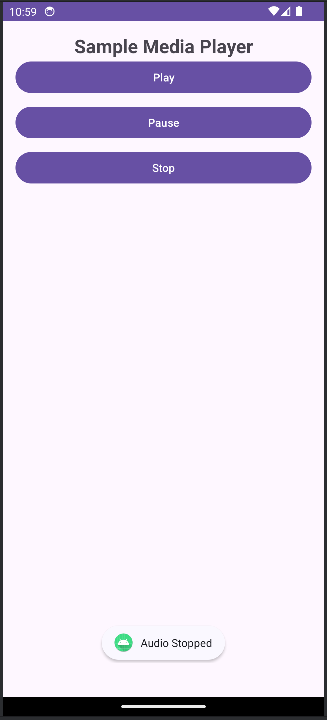
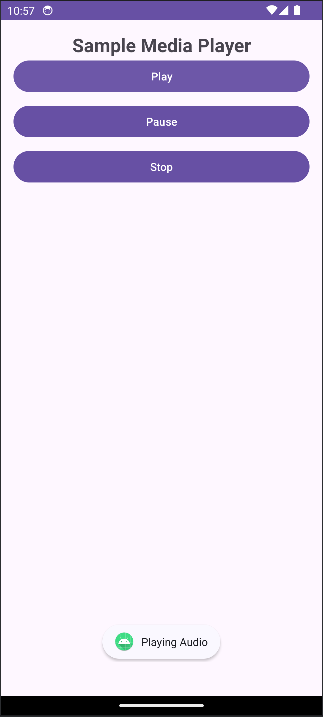
mediaPlayer = null;

}

}

}

**Output:**

****

#### Conclusion:

#### ………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………