# **ASSIGNMENT NO.7**

# **Gaurav\_bodkhe\_2124UCEM1041**

## **1. Introduction**

## In this program, we will develop a simple video player application that plays a pre-selected video file. The app will feature a user-friendly interface with Play, Pause, Forward, and Rewind buttons to control video playback. The video player will automatically load and play the designated video, allowing users to control the playback experience using the provided controls. This project demonstrates how to build a basic video player application with a fixed video source, providing a straightforward way to display video content.

## **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:**

A screenshot of a computer

AI-generated content may be incorrect.

### 

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

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

**Screenshot:**

**A screenshot of a computer

AI-generated content may be incorrect.**

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

* Open Activity\_main.xml
* Implement functionality such as. LinearLayout, VideoView etc.
* Use necessary Android components like LinearLayout, VideoView etc
* **Screenshot:**

A computer screen shot of a program

AI-generated content may be incorrect.

* **XML**

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

*<!--suppress ALL -->*

*<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:background="#1A1818"*

*android:orientation="vertical"*

*tools:context=".MainActivity">*

*<!-- First LinearLayout containing the VideoView -->*

*<LinearLayout*

*android:layout\_width="wrap\_content"*

*android:layout\_height="wrap\_content">*

*<VideoView*

*android:id="@+id/videoview"*

*android:layout\_width="match\_parent"*

*android:layout\_height="250dp" />*

*</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 Forward the Video is get Forworded by some time(10 sec) Whereas On Click the Rewind the song is get Rewind by Some Time(10 sec) And when clicking on pause they get Pause.

**Screenshot:**

**A computer screen shot of a program

AI-generated content may be incorrect.A computer screen shot of a program code

AI-generated content may be incorrect.**

## **Code(java):**

*package com.example.a7;  
  
import android.net.Uri;  
import android.os.Bundle;  
import android.widget.MediaController;  
import android.widget.VideoView;  
  
import androidx.activity.EdgeToEdge;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.graphics.Insets;  
import androidx.core.view.ViewCompat;  
import androidx.core.view.WindowInsetsCompat;  
  
public class MainActivity extends AppCompatActivity {  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 EdgeToEdge.enable(this);  
 setContentView(R.layout.activity\_main);  
  
 VideoView videoView = findViewById(R.id.videoview);  
  
 String vPath = "android.resource://"+getPackageName()+"/raw/three";  
  
  
 Uri videoURI = Uri.parse(vPath);  
  
 //videoView.setVideoPath(vPath);  
 videoView.setVideoURI(videoURI);  
 videoView.start();  
 MediaController mediaController = new MediaController(this);  
 videoView.setMediaController(mediaController);  
 mediaController.setAnchorView(videoView);  
  
 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;  
 });  
 }  
}*

### **Step 5: Adding an Video**

* Right click on the Res file present in the app folder after that create an Another Android Resource directory name as raw
* In that raw file put an mp4 Video and give the Address in MainActivity.java Folder

**Screenshot:**

*A screenshot of a computer

AI-generated content may be incorrect.*

*A screenshot of a computer program

AI-generated content may be incorrect.*

### **Step 6: Running the Application on Emulator**

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

**Screenshot:**

### **A screen shot of a computer AI-generated content may be incorrect.**

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

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

**Screenshot:**

A screen shot of a cell phone

AI-generated content may be incorrect.

A screen shot of a cell phone

AI-generated content may be incorrect.

## **4. Conclusion**

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

## During development, I encountered challenges related to managing media resources and ensuring smooth Video 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.