Experiment 2.1

Student Name: Debdulal Das **UID:** 21BCS9011
Branch: CSE **Section/Group:**21BCS-CC-646-A
Semester: 6
Subject Name: Mobile Application Development
Subject Code: 21CSH-355

Aim:

Create an Android app that uses Intent with button to create a page and passes values from one activity to another.

Objective:

The objective of an Android app that uses Intent with a button to create a page and passes values from one activity to another could be to demonstrate and implement a simple data communication flow between different activities within an Android application. This type of app is commonly used to understand and showcase the concept of passing data between different screens or pages in Android.

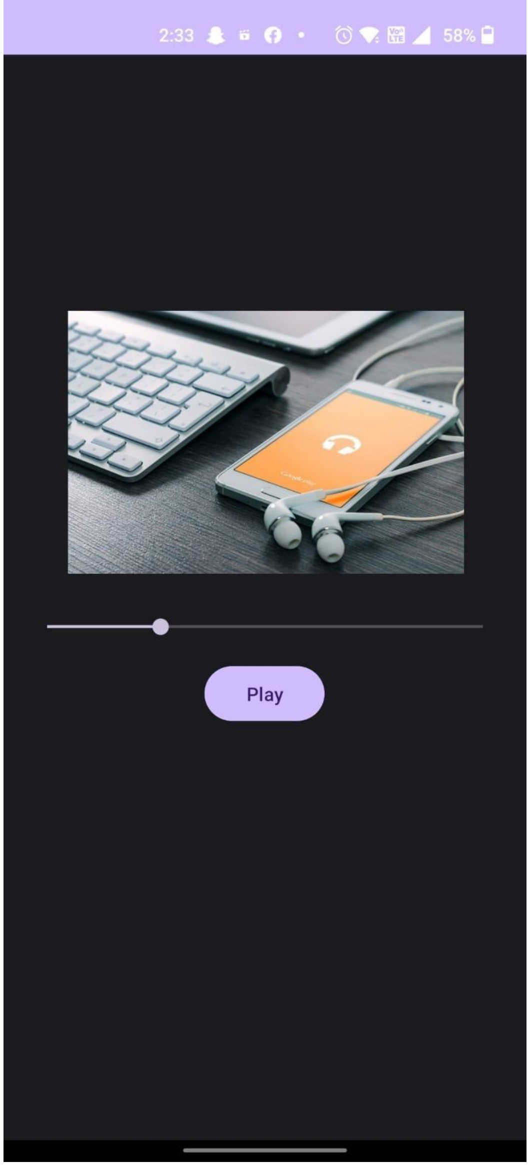
Code:

AndroidManifest.java:

```
package com.example.mad4;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

import android.content.pm.PackageManager;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.os.Environment;
import android.os.Handler;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.SeekBar;

import java.io.File;
import java.io.IOException;
```

```
public class MainActivity extends AppCompatActivity {

    private static final int REQUEST_PERMISSION_READ_EXTERNAL_STORAGE = 1;
    private Button playButton;
    private SeekBar seekBar;
    private MediaPlayer mediaPlayer;

    private boolean isPlaying = false;
    private Handler mHandler;
    private Runnable mUpdateSeekbar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Check for permission to read external storage
        if (ContextCompat.checkSelfPermission(this,
android.Manifest.permission.READ_EXTERNAL_STORAGE) !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new
String[]{android.Manifest.permission.READ_EXTERNAL_STORAGE},
REQUEST_PERMISSION_READ_EXTERNAL_STORAGE);
        }

        // Construct the path to your music file
        File downloadsFolder =
Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_MUSIC);
        String musicFileName = "Song1.mp3";
        File musicFile = new File(downloadsFolder, musicFileName);

        //Create media player
        mediaPlayer = new MediaPlayer();
        try {
            mediaPlayer.setDataSource(musicFile.getAbsolutePath());
            mediaPlayer.prepare();
        } catch (IOException e) {
            e.printStackTrace();
        }
        //Handle seekbar progress
        seekBar = findViewById(R.id.seekBar);
        mHandler = new Handler();
        mUpdateSeekbar = new Runnable() {
            @Override
            public void run() {
                System.out.println("inside run method");
                if(mediaPlayer != null){
                    System.out.println(mediaPlayer);
                    System.out.println("inside media player");
                    int currentPosition = mediaPlayer.getCurrentPosition();
                    int totalDuration = mediaPlayer.getDuration();
                    if(totalDuration==0){
                        return;
                    }
                    seekBar.setProgress(currentPosition * 100 / totalDuration);
                    mHandler.postDelayed(this, 1000);
                }
            }
        };
        seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
            @Override
            public void onProgressChanged(SeekBar seekBar, int progress, boolean
fromUser) {
                if(fromUser){
                    int newPosition = mediaPlayer.getDuration() * progress / 100;
                    mediaPlayer.seekTo(newPosition);
                }

            }

            @Override
            public void onStartTrackingTouch(SeekBar seekBar) {
                // Pause the audio when user starts seeking
                if(mediaPlayer.isPlaying()){
                    mediaPlayer.pause();
                }

            }

            @Override
            public void onStopTrackingTouch(SeekBar seekBar) {
                // Resume audio playback when user stops seeking
                mediaPlayer.start();
            }
        });


        //Setup playback button
        playButton = findViewById(R.id.playButton);
        playButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if(isPlaying){
                    mediaPlayer.pause();
                    playButton.setText("Play");
     //remove the callbacks to the Runnable when the MediaPlayer stops
                    mHandler.removeCallbacks(mUpdateSeekbar);
                }else{
                    mediaPlayer.start();
                    playButton.setText("Pause");
                    //start the Runnable when the MediaPlayer starts playing
                    mHandler.postDelayed(mUpdateSeekbar, 1000);
                }
                isPlaying = !isPlaying;
            }
        });

    }
    //Release the MediaPlayer resources when the activity is destroyed
    @Override
    protected void onDestroy(){
        super.onDestroy();
        if(mediaPlayer != null){
            mediaPlayer.release();
            mediaPlayer = null;
            mHandler.removeCallbacks(mUpdateSeekbar);
        }
    }
```

OUTPUT:



Learning Outcomes:

- Successful USE of your Android development environment.
- Project Workspace.
- Configuration Intent Completion.
- How to add new Intent