EXPERIMENT 5B

NAME: Shivam Pawar **UID: 2019230068**

NAME: Vishal Salvi **UID: 2019230069**

NAME: Shreyas Patel **UID: 2018130037**

CLASS: TE COMPS **BATCH: C**

DATE:

Aim: To develop app for mobile device

Theory:

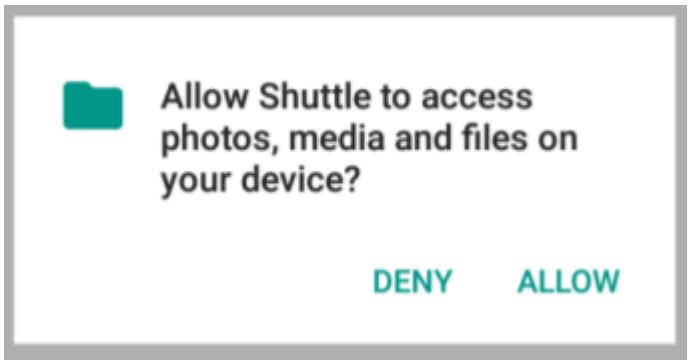
Prob Definition:

We are implementing Music Stream Application, this application help to listen music from external storage. User can listen music without internet access as well as can downloaded music can also added in this application.

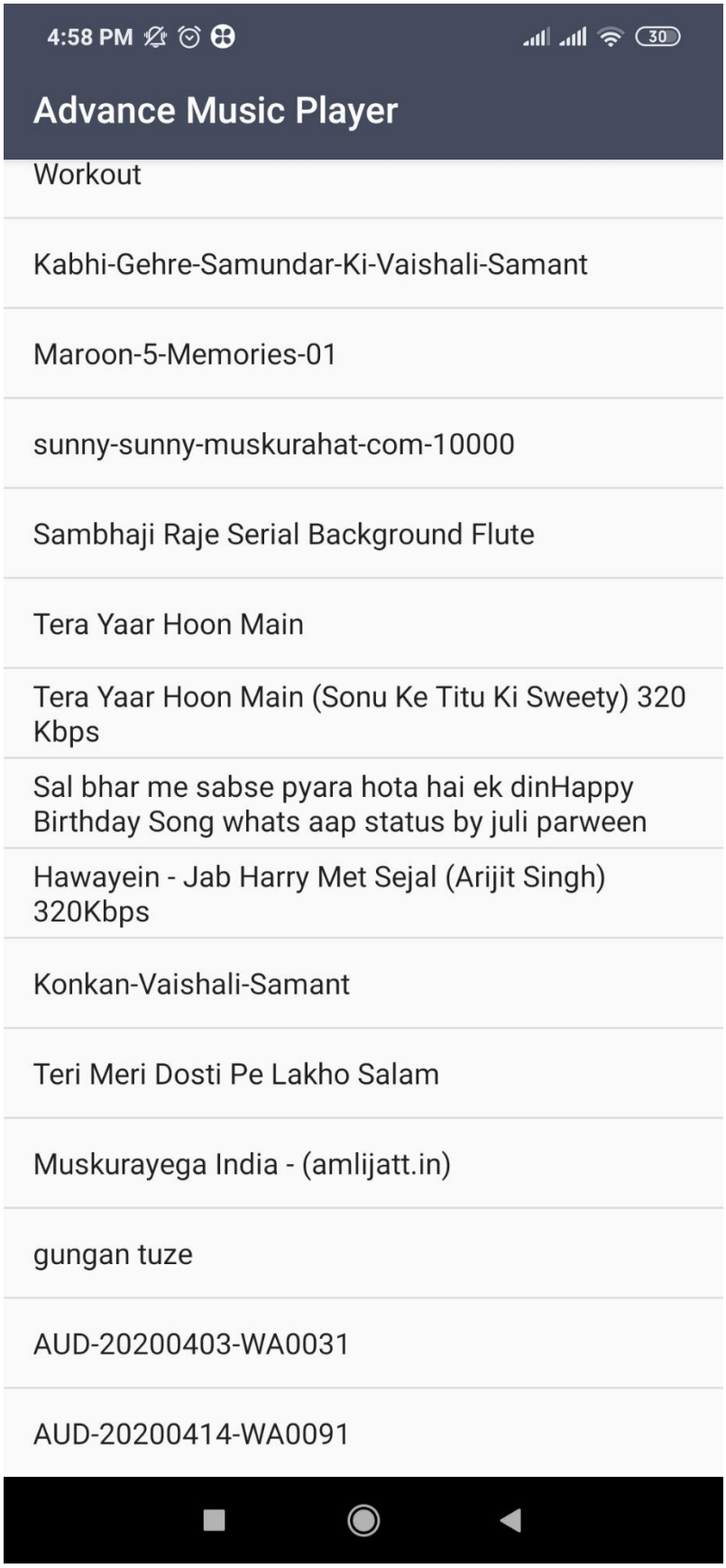
User can Paused at any time, Listen Next as well as previous song.

Screen Shots

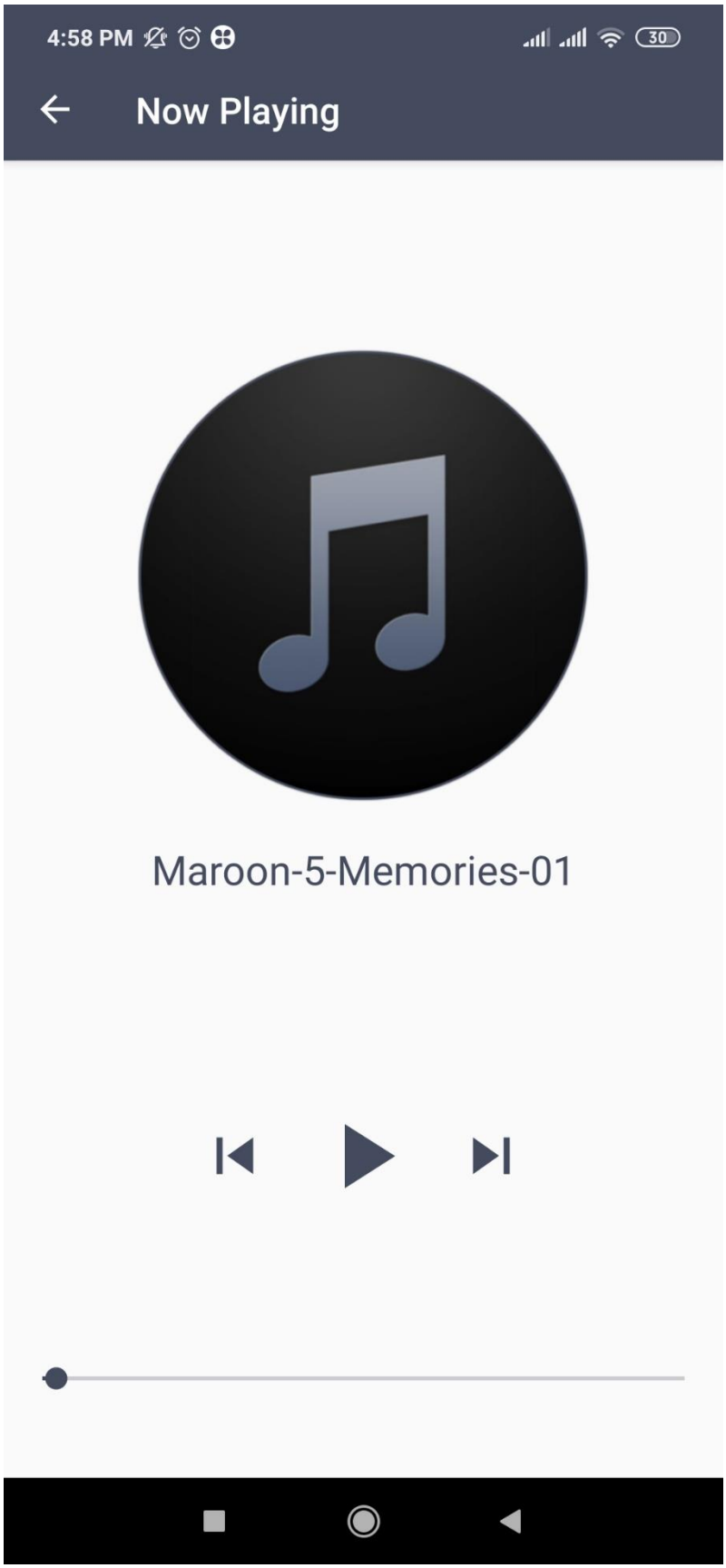
Application allow user to access photos, media and files from mobile device.



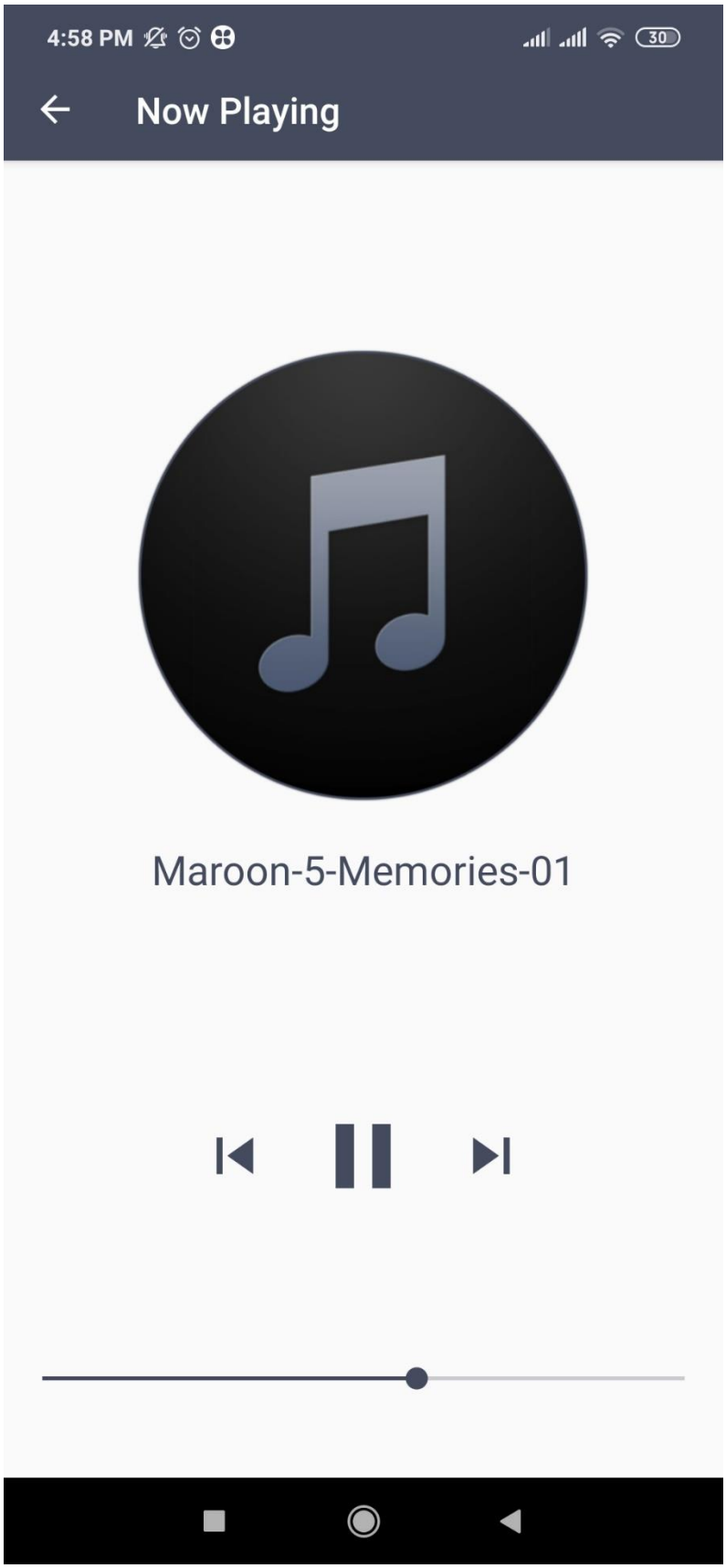
After that list is display on the screen. Following are the music which are form external device storage.



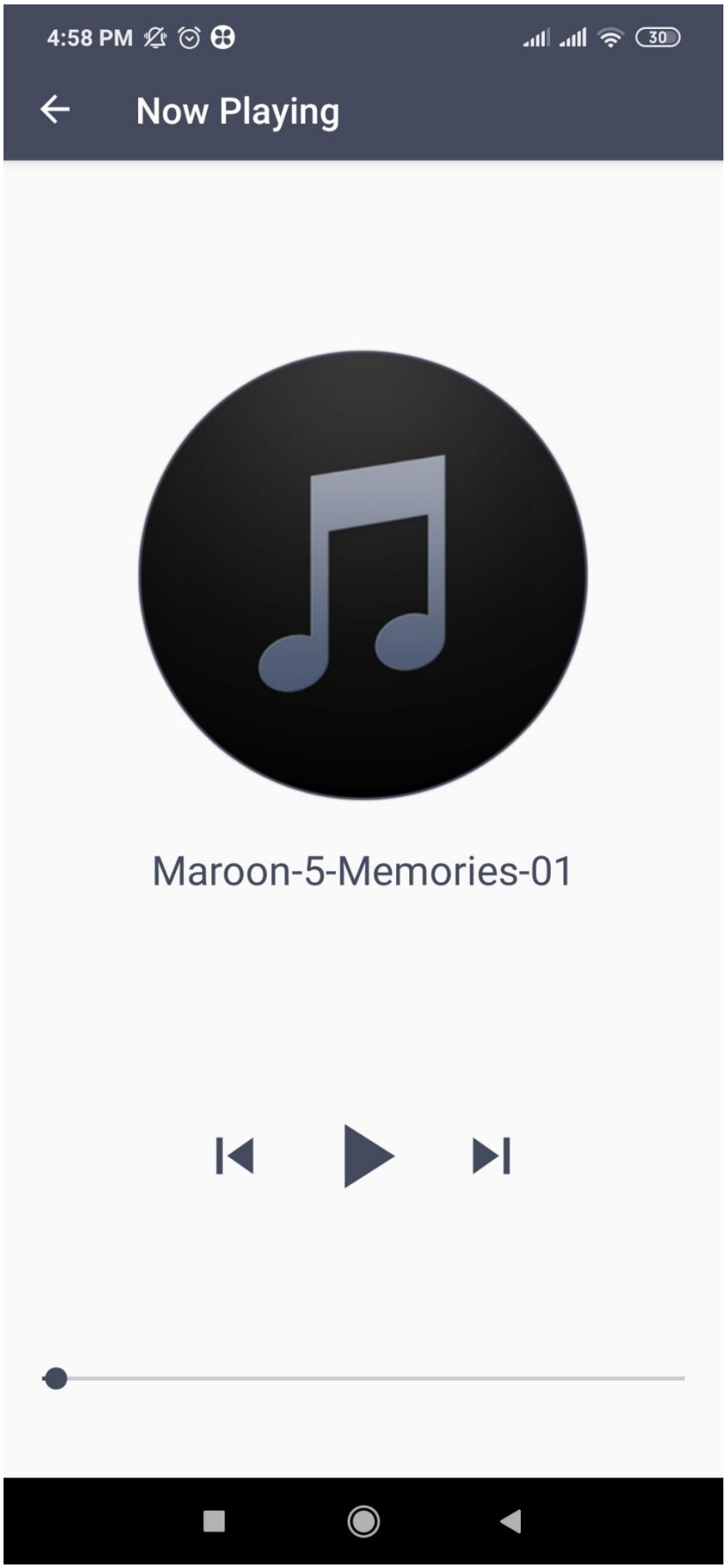
User can select any song from application and listen.



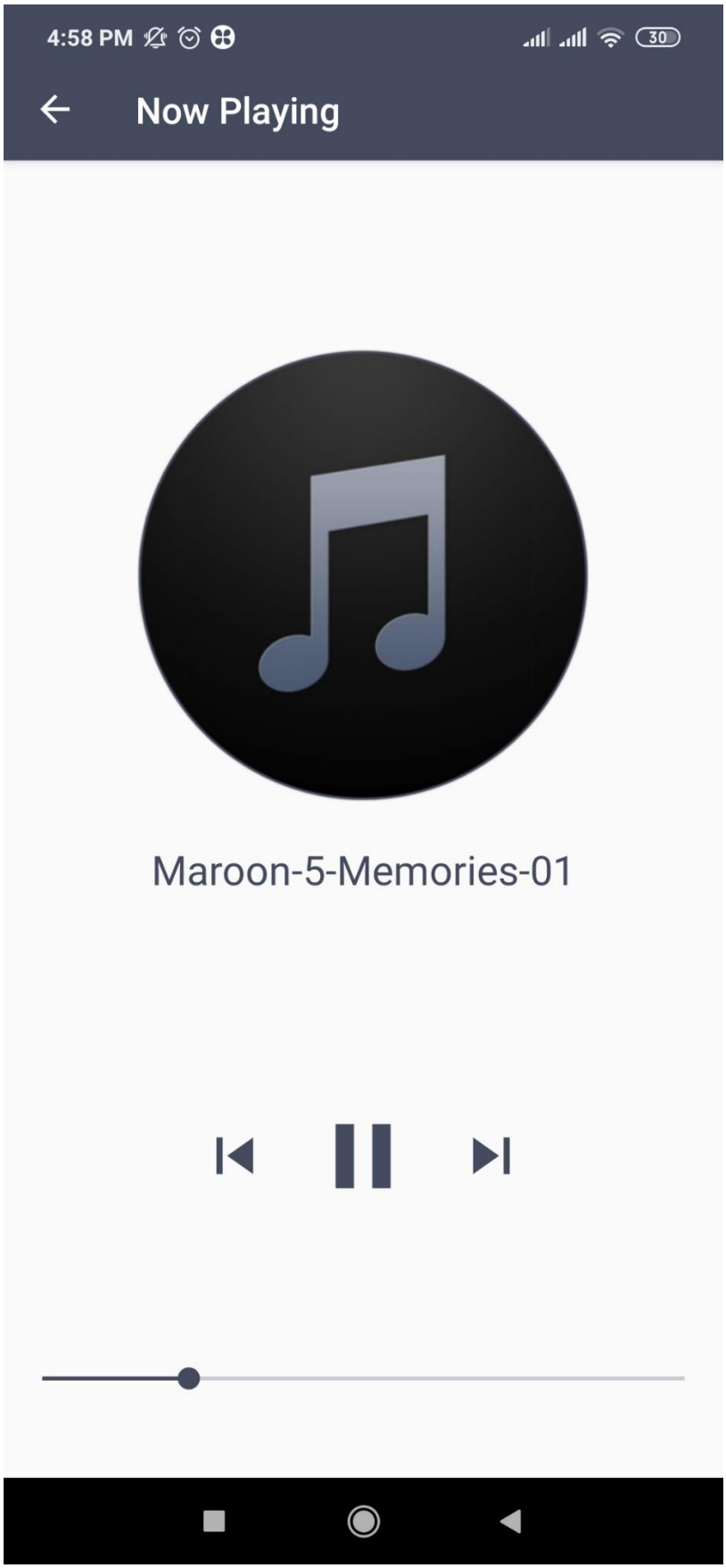
User can move Seekbar and listen on different time position.



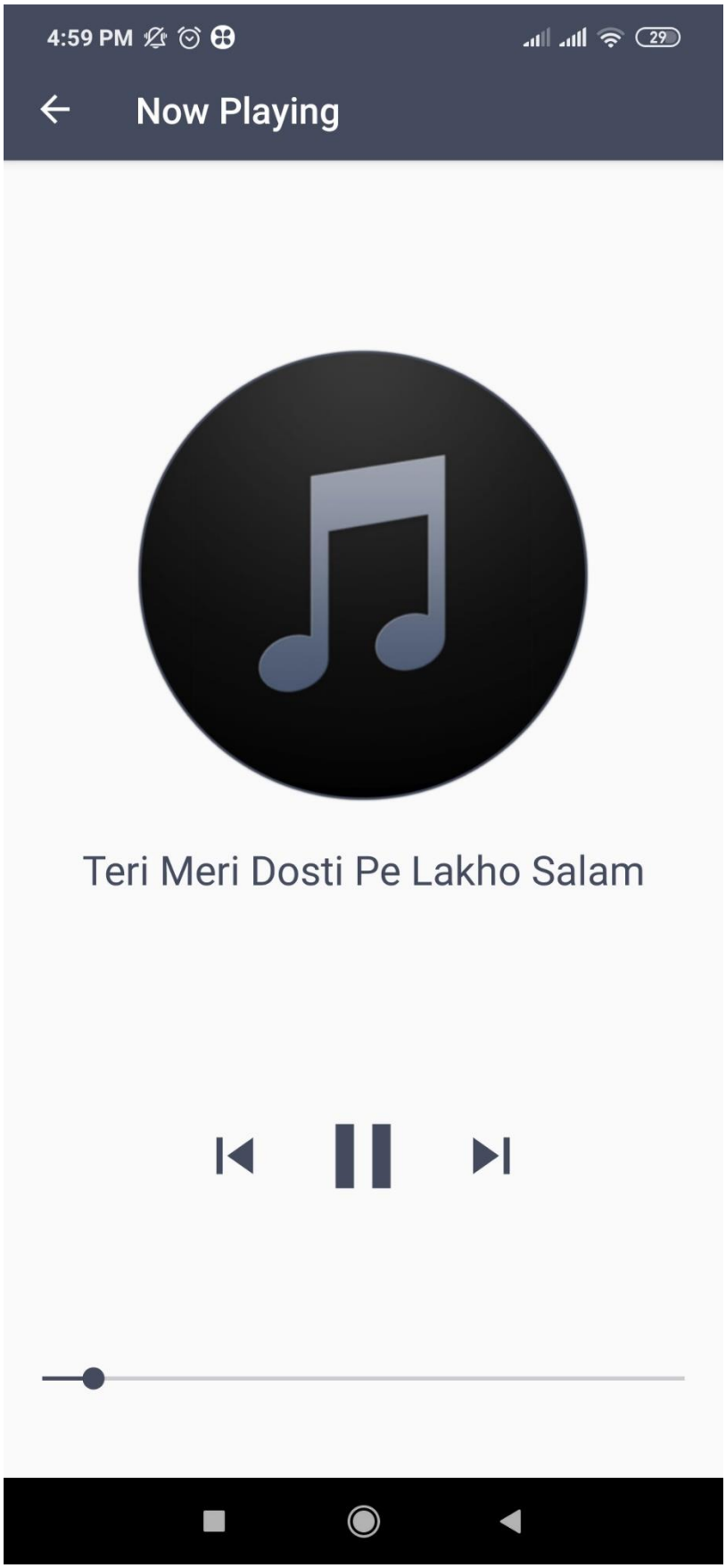
User can stop song(Paused)



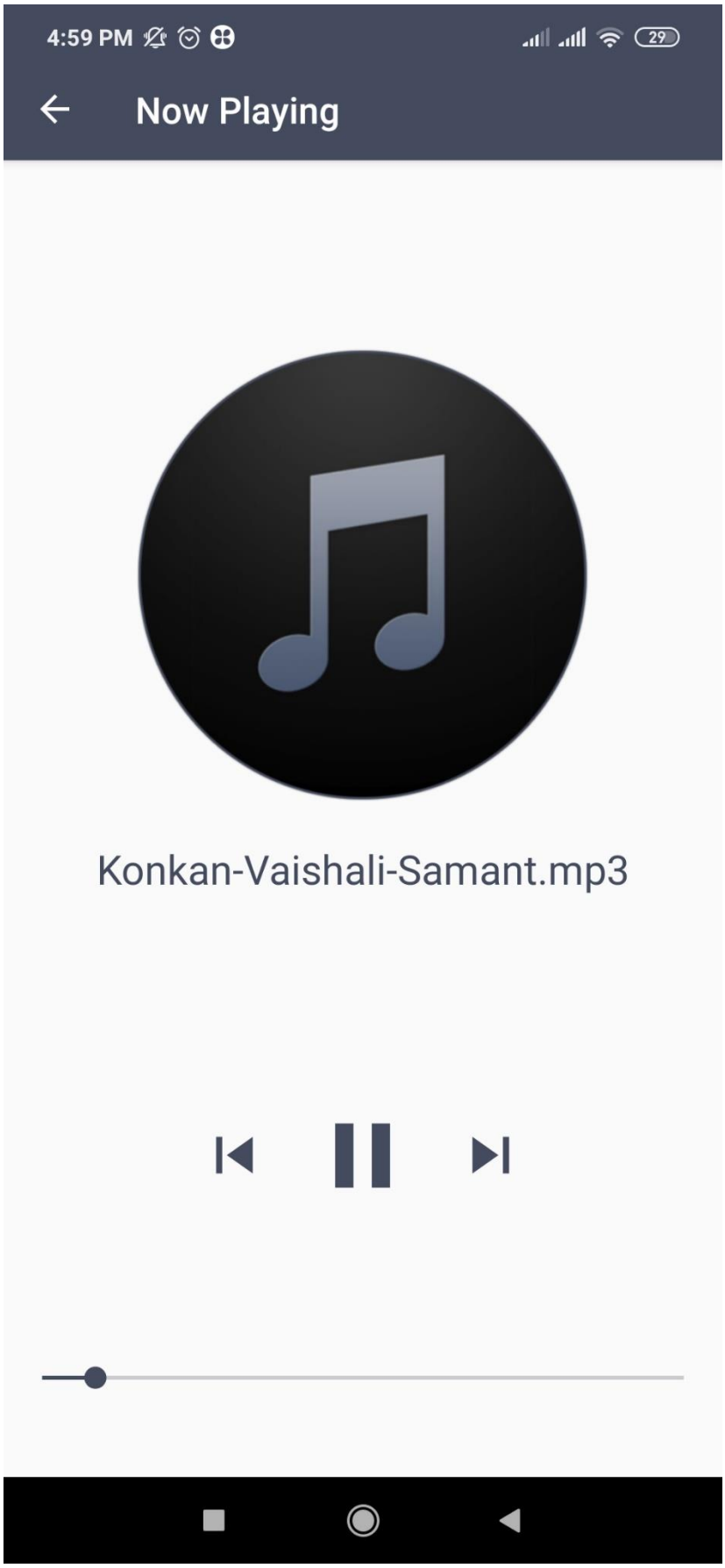
User can again play song

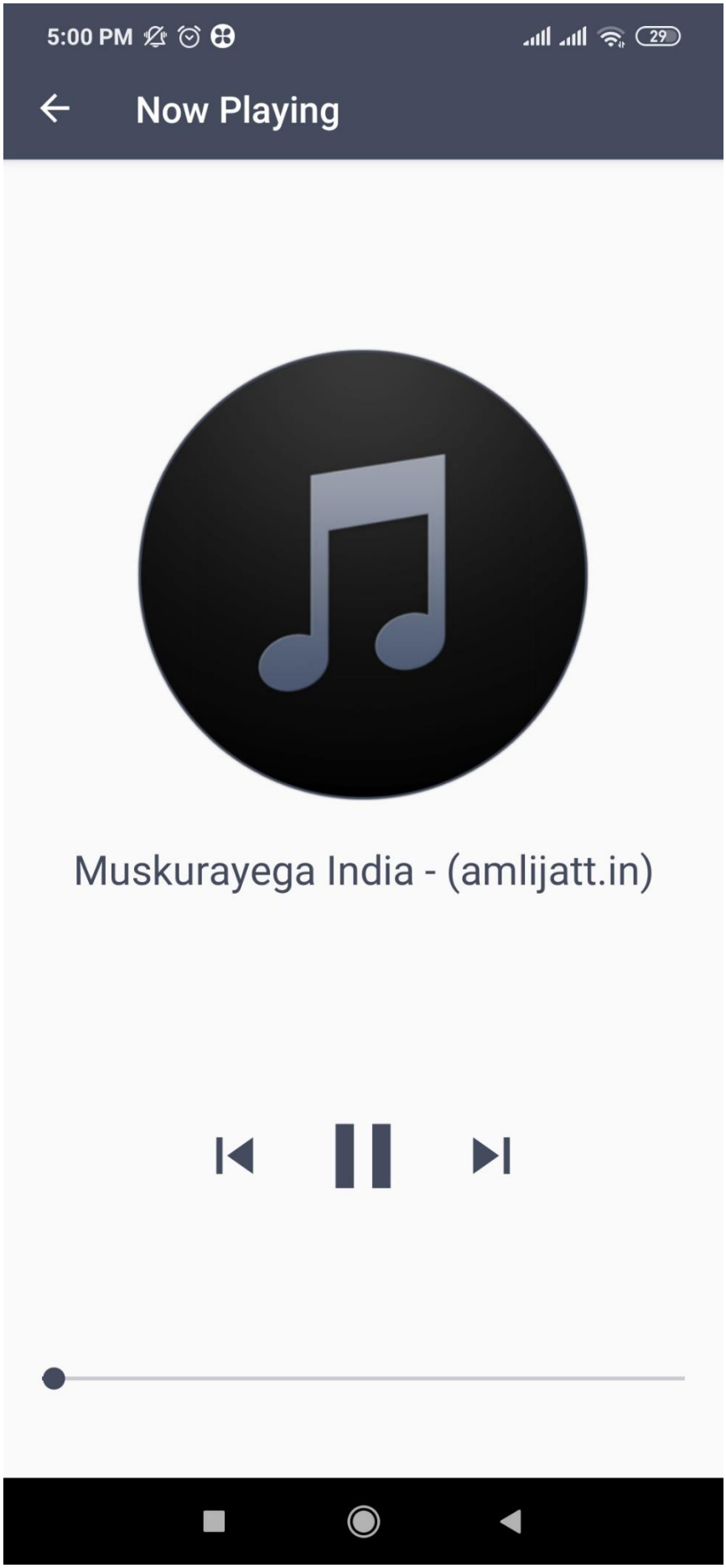


User can click on next arrow to listen next song.

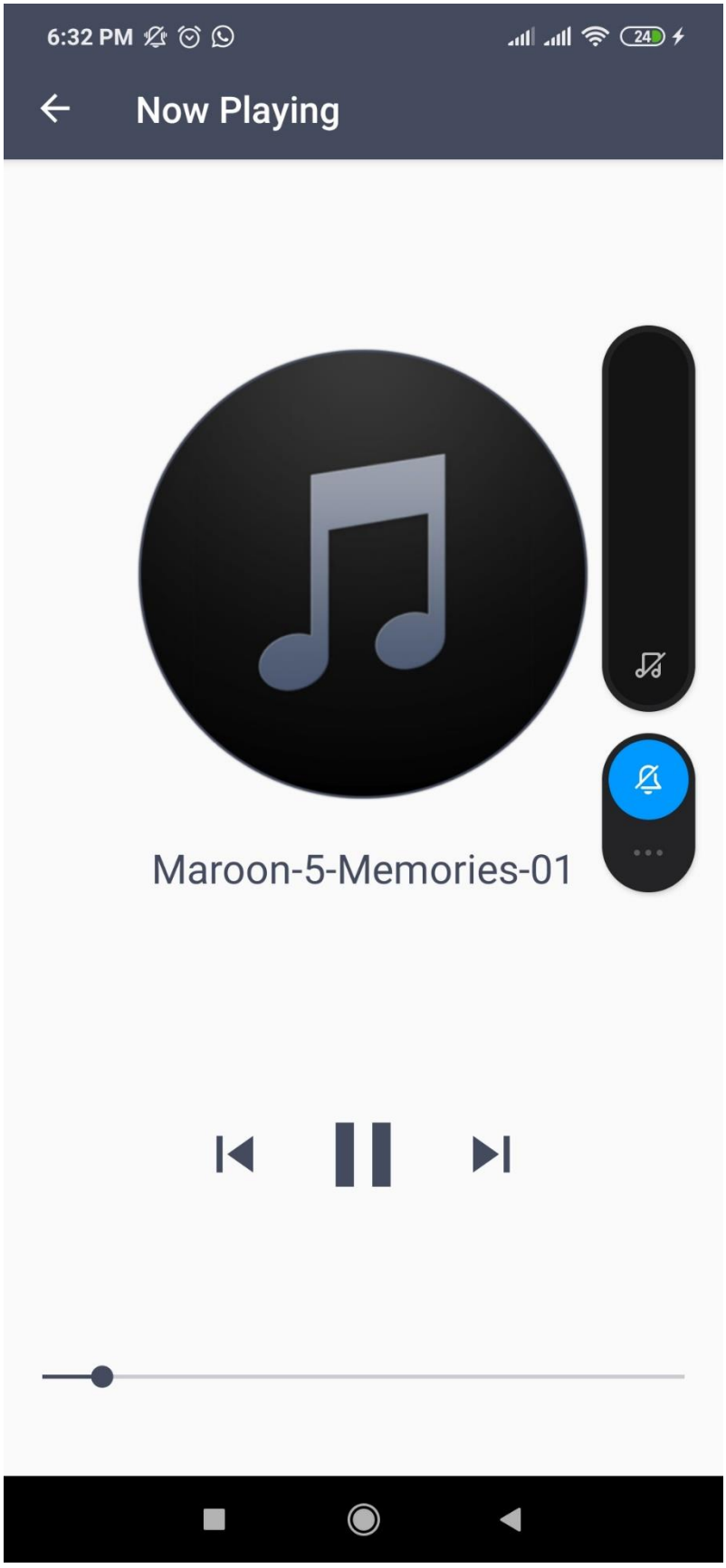


User can click on Back arrow to listen Previous song.

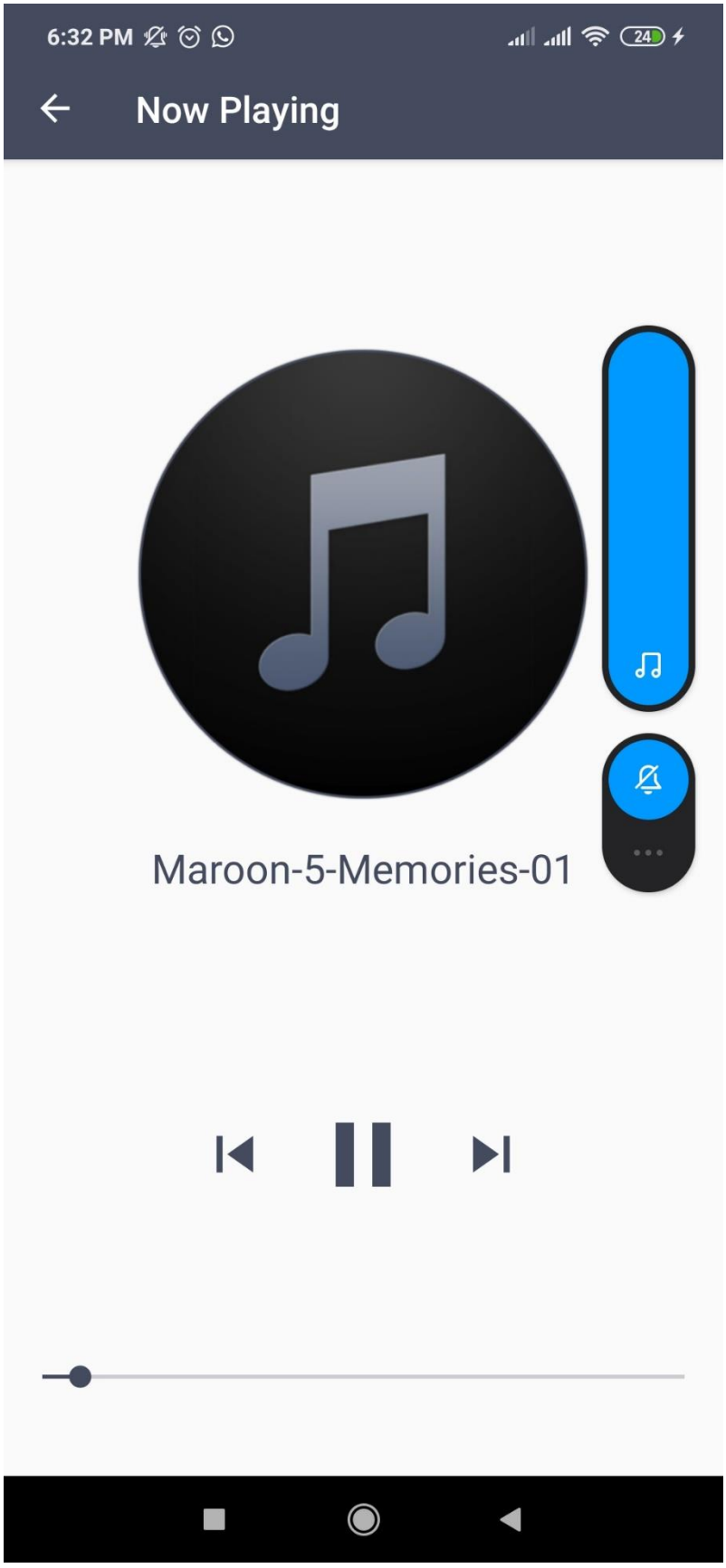



5:00 PM
29
Now Playing
Muskurayega India - (amlijatt.in)

User can adjust Volume (Low)



Volume High



Working Recording:

https://drive.google.com/folderview?id=1MD1oJ6XcYTQaJ1k_lUItEVwaOxPtASoN

Code Snippet:

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="15dp"
    tools:context=".MainActivity">

    <ListView
        android:id="@+id/mySongListView"
        android:layout_width="402dp"
        android:layout_height="722dp">
    </ListView>

</RelativeLayout>
```

MainActivity.java

```
package com.example.shuttle;

import androidx.appcompat.app.AppCompatActivity;
import android.Manifest;
import android.content.Intent;
import android.os.Bundle;
import android.os.Environment;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import com.karumi.dexter.Dexter;
import com.karumi.dexter.PermissionToken;
import com.karumi.dexter.listener.PermissionDeniedResponse;
import com.karumi.dexter.listener.PermissionGrantedResponse;
import com.karumi.dexter.listener.PermissionRequest;
import com.karumi.dexter.listener.single.PermissionListener;
import java.io.File;
import java.util.ArrayList;
import dalvik.system.DexClassLoader;

public class MainActivity extends AppCompatActivity {
    ListView myListViewforSongs;
    String[] items;
```

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

        myListViewforSongs=(ListView) findViewById(R.id.mySongListView);
        runtimePermission();
    }

    public void runtimePermission(){
        Dexter.withActivity(this)
                .withPermission(Manifest.permission.READ_EXTERNAL_STORAGE)
                .withListener(new PermissionListener() {
                    @Override
                    public void onPermissionGranted(PermissionGrantedResponse response) {

                        display();

                    }

                    @Override
                    public void onPermissionDenied(PermissionDeniedResponse response) {

                    }

                    @Override
                    public void onPermissionRationaleShouldBeShown(PermissionRequest
permission, PermissionToken token) {
                        token.continuePermissionRequest();

                    }
                }).check();
    }
    public ArrayList<File> findSong(File file){
        ArrayList<File> arrayList=new ArrayList<>();
        File[] files= file.listFiles();
        for(File singleFile: files){
            if (singleFile.isDirectory() && !singleFile.isHidden()) {

                arrayList.addAll(findSong(singleFile));
            }
            else{
                if(singleFile.getName().endsWith(".mp3") ||
                        singleFile.getName().endsWith(".wav")){
                    arrayList.add(singleFile);
                }}
        }
        return arrayList;
    }
```

```
    void display(){
        final ArrayList<File> mySongs=findSong(Environment.getExternalStorageDirectory());
        items=new String[mySongs.size()];
        for(int i=0;i<mySongs.size();i++){
            items[i] =
mySongs.get(i).getName().toString().replace(".mp3","").replace(".wav","");

        }
        ArrayAdapter<String> myAdapter= new
ArrayAdapter<String>(this,android.R.layout.simple_list_item_1,items);
        myListViewforSongs.setAdapter(myAdapter);

        myListViewforSongs.setOnItemClickListener(new AdapterView.OnItemClickListener()
{
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {

                String songName = myListViewforSongs.getItemAtPosition(i).toString();

                startActivity(new Intent(getApplicationContext(),PlayerActivity.class)
                        .putExtra("songs",mySongs).putExtra("songname",songName)
                        .putExtra("pos",i));


            }


        });

    }
}
```

player_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<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:layout_width="match_parent"
    android:orientation="vertical"
    android:weightSum="10"
    android:layout_height="match_parent"
    tools:context=".PlayerActivity">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="7"
        android:orientation="vertical"
        android:gravity="center"
        >
        <ImageView
```

```
        android:layout_width="250dp"
        android:layout_height="250dp"
        android:src="@drawable/music" />

    <TextView
        android:id="@+id/songLabel"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:ellipsize="marquee"
        android:marqueeRepeatLimit="marquee_forever"
        android:scrollHorizontally="true"
        android:singleLine="true"
        android:text="Song Name"
        android:textColor="#000000"
        android:textAlignment="center"
        android:textStyle="bold"
        android:textSize="30sp" />


</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="3">
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="5dp">
        <SeekBar
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/seekbar"
            android:layout_alignParentBottom="true"
            android:layout_marginBottom="40dp" />
        <Button
            android:layout_width="60dp"
            android:layout_height="60dp"
            android:layout_centerHorizontal="true"
            android:background="@drawable/icon_pause"
            android:id="@+id/pause"
            android:layout_marginTop="5dp" />

        <Button
            android:id="@+id/next"
            android:layout_width="40dp"
            android:layout_height="40dp"
            android:layout_marginLeft="45dp"
            android:layout_marginTop="15dp"
            android:layout_toRightOf="@+id/pause"
```

```
            android:background="@drawable/icon_next"/>

        <Button
            android:id="@+id/previous"
            android:layout_width="40dp"
            android:layout_height="40dp"
            android:layout_marginLeft="45dp"
            android:layout_marginTop="15dp"
            android:layout_marginRight="45dp"
            android:layout_toLeftOf="@+id/pause"
            android:background="@drawable/icon_previous" />
    </RelativeLayout>
  </LinearLayout>
</LinearLayout>
```

PlayerActivity.java

```
package com.example.shuttle;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.annotation.SuppressLint;
import android.content.Intent;
import android.graphics.PorterDuff;
import android.media.MediaPlayer;
import android.net.Uri;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.SeekBar;
import android.widget.TextView;
import java.io.File;
import java.util.ArrayList;
import static android.graphics.PorterDuff.Mode.SRC_IN;
public class PlayerActivity extends AppCompatActivity {
    Button btn_next, btn_previous, btn_pause;
    TextView songText;
    SeekBar songSeekBar;
    String sname;
    static MediaPlayer myMediaPlayer;
    int position;
    ArrayList<File> mySongs;
    Thread updateseekbar;

    @SuppressLint("NewApi")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_player);
        btn_next = (Button) findViewById(R.id.next);
        btn_previous = (Button) findViewById(R.id.previous);
```

```
        btn_pause = (Button) findViewById(R.id.pause);
        songText = (TextView) findViewById(R.id.songLabel);
        songSeekBar = (SeekBar) findViewById(R.id.seekbar);
        getSupportActionBar().setTitle("Now Playing");
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayShowHomeEnabled(true);
        updateseekbar = new Thread() {
            public void run() {
                int totalDuration = myMediaPlayer.getDuration();
                int currentPosition = 0;
                while (currentPosition < totalDuration) {
                    try {
                        sleep(500);
                        currentPosition = myMediaPlayer.getCurrentPosition();
                        songSeekBar.setProgress(currentPosition);
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }

                }

            }
        };
        if (myMediaPlayer != null) {
            myMediaPlayer.stop();
            myMediaPlayer.release();

        }
        Intent i = getIntent();
        Bundle bundle = i.getExtras();
        mySongs = (ArrayList) bundle.getParcelableArrayList("songs");
        sname = mySongs.get(position).getName().toString();
        String songName = i.getStringExtra("songname");
        songText.setText(songName);
        songText.setSelected(true);
        position = bundle.getInt("pos", 0);
        Uri u = Uri.parse(mySongs.get(position).toString());
        myMediaPlayer = MediaPlayer.create(getApplicationContext(), u);
        myMediaPlayer.start();
        songSeekBar.setMax(myMediaPlayer.getDuration());
        updateseekbar.start();

songSeekBar.getProgressDrawable().setColorFilter(getResources().getColor(R.color.colorPr
imary), PorterDuff.Mode.MULTIPLY);

songSeekBar.getThumb().setColorFilter(getResources().getColor(R.color.colorPrimary),
PorterDuff.Mode.SRC_IN);
        songSeekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener()
{
            @Override
```

```
    public void onProgressChanged(SeekBar seekBar, int i, boolean b) {

    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {

    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {
        myMediaPlayer.seekTo(seekBar.getProgress());

    }
});

btn_pause.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        songSeekBar.setMax(myMediaPlayer.getDuration());
        if (myMediaPlayer.isPlaying()) {
            btn_pause.setBackgroundResource(R.drawable.icon_play);
            myMediaPlayer.pause();
        } else {
            btn_pause.setBackgroundResource(R.drawable.icon_pause);
            myMediaPlayer.start();
        }
    }
});
btn_next.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        myMediaPlayer.stop();
        myMediaPlayer.release();
        position = ((position + 1) % mySongs.size());
        Uri u = Uri.parse(mySongs.get(position).toString());
        myMediaPlayer = MediaPlayer.create(getApplicationContext(), u);
        sname = mySongs.get(position).getName().toString();
        songText.setText(sname);
        myMediaPlayer.start();
    }
});
btn_previous.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        myMediaPlayer.stop();
        myMediaPlayer.release();

        position = ((position - 1) < 0) ? (mySongs.size()) - 1 : (position - 1);
        Uri u = Uri.parse(mySongs.get(position).toString());
```

```
            myMediaPlayer = MediaPlayer.create(getApplicationContext(), u);
            sname = mySongs.get(position).getName().toString();
            songText.setText(sname);
            myMediaPlayer.start();

        }
      });
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
      if (item.getItemId() == android.R.id.home) {
        onBackPressed();
      }
      return super.onOptionsItemSelected(item);
    }
}
```

dependencies:

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'androidx.appcompat:appcompat:1.0.2'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test:runner:1.2.0'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
    implementation 'com.karumi:dexter:6.0.0'

}
```

AndroidManifest.xml

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


    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".PlayerActivity"></activity>
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
```

```
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>

</manifest>
```

Conclusion:

Thus From this experiment we built Music Stream Application.