Ex.No: 1 To create a database table and to display the database table field using SQLite Database in Android Studio.

Date: 07.09.2022

AIM:

To create a database table and to display the database table field using SQLite Database in Android Studio.

EQUIPMENTS REQUIRED:

Android Studio(Min.required Arctic Fox)

ALGORITHM:

Step 1: Open Android Studio and then click on File -> New -> New project.

Step 2: Then type the Application name as sqllite and click Next.

Step 3: Then select the Minimum SDK as shown below and click Next.

Step 4: Then select the Empty Activity and click Next. Finally click Finish.

Step 5: Design layout in activity_main.xml.

Step 6: Enter the code in MainActivity file.

Step 7: Save and run the application.

PROGRAM:

MainActivity.java

```
package com.example.sqlliteapp;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;

import android.database.Cursor;

import android.os.Bundle;

import  android.util.Log;

import android.view.View;

import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

    EditText editUserID;

    EditText editUserName;

    EditText editUserPassword;
```

```
    DatabaseManager dbManager;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        editUserID= findViewById(R.id.editTextID);

        editUserName=findViewById(R.id.editTextUserName);

        editUserPassword=findViewById(R.id.editTextPassword);

        dbManager=new DatabaseManager(this);

        try{

            dbManager.open();

        }

        catch(Exception e){

            e.printStackTrace();

        }

    }

    public void btnInsertPressed(View
v){   dbManager.insert(editUserName.getText().toString(),editUserPassword.getTe
xt().toString());

    }

    public void btnFetchPressed(View v){

        Cursor cursor=dbManager.fetch();

        if (cursor.moveToFirst())

        {

            do{

                @SuppressLint("Range") String
ID=cursor.getString(cursor.getColumnIndex(DatabaseHelper.USER_ID));

                @SuppressLint("Range") String
username=cursor.getString(cursor.getColumnIndex(DatabaseHelper.USER_NAME));

                @SuppressLint("Range") String
password=cursor.getString(cursor.getColumnIndex(DatabaseHelper.USER_PASSWORD));

                Log.i("DATABASE_TAG","I have read ID : "+ID+" Username :
"+username+" password : "+password);

            }while(cursor.moveToNext());

        }

    }
```

```
    public void btnUpdatePressed(View
v){   dbManager.update(Long.parseLong(editUserID.getText().toString()),editUser
Name.getText().toString(),editUserPassword.getText().toString());

    }

    public void btnDeletePressed(View v){

        dbManager.delete(Long.parseLong(editUserID.getText().toString()));

    }

}
```

DatabaseManager.java

```
package com.example.sqlliteapp;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import java.sql.SQLDataException;

public class DatabaseManager {

    private DatabaseHelper dbHelper;

    private Context context;

    private SQLiteDatabase database;

    public DatabaseManager(Context ctx){

        context=ctx;

    }

    public DatabaseManager open() throws SQLDataException {

        dbHelper = new DatabaseHelper(context);

        database = dbHelper.getWritableDatabase();

        return this;

    }

    public void close(){

        dbHelper.close();

    }

    public void insert (String username, String password){

        ContentValues contentValues=new ContentValues();

        contentValues.put(DatabaseHelper.USER_NAME,username);
```

```
        contentValues.put(DatabaseHelper.USER_PASSWORD,password);

        database.insert(DatabaseHelper.DATABASE_TABLE,null,contentValues);

    }

    public Cursor fetch(){

        String [] columns= new
String[]{DatabaseHelper.USER_ID,DatabaseHelper.USER_NAME,DatabaseHelper.USER_PA
SSWORD};

        Cursor
cursor=database.query(DatabaseHelper.DATABASE_TABLE,columns,null,null,null,null,
null);

        if(cursor!=null){

            cursor.moveToFirst();

        }

        return cursor;

    }

    public int update(long _id,String username,String password){

        ContentValues contentValues=new ContentValues();

        contentValues.put(DatabaseHelper.USER_NAME,username);

        contentValues.put(DatabaseHelper.USER_PASSWORD,password);

        int ret =
database.update(DatabaseHelper.DATABASE_TABLE,contentValues,DatabaseHelper.USER
_ID+"="+_id,null);

        return ret;

    }

    public void delete(long id){


database.delete(DatabaseHelper.DATABASE_TABLE,DatabaseHelper.USER_ID,null);

    }

}
```

DatabaseHelper.java

```
package com.example.sqlliteapp;

import android.content.Context;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;
```

```
public class DatabaseHelper extends SQLiteOpenHelper {

    static final String DATABASE_NAME="DataBASE.DB";

    static final int DATABASE_VERSION=1;

    static final String DATABASE_TABLE= "USERS";

    static final String USER_ID= "_ID";

    static final String USER_NAME="user_name";

    static final String USER_PASSWORD="password";

    private static final String CREATE_DB_QUERY = "CREATE TABLE
"+DATABASE_TABLE +" ( "+ USER_ID+ " INTEGER PRIMARY KEY AUTOINCREMENT,
"+USER_NAME+ " TEXT NOT NULL,   "+ USER_PASSWORD+ " TEXT NOT NULL );";

    public DatabaseHelper(Context context) {

        super(context, DATABASE_NAME, null, DATABASE_VERSION);

    }

    @Override

    public void onCreate(SQLiteDatabase db) {

        db.execSQL(CREATE_DB_QUERY);

    }

    @Override

    public void onUpgrade(SQLiteDatabase db, int i, int i1) {

    db.execSQL("DROP TABLE IF EXISTS "+ DATABASE_TABLE);

    }

}
```

activity_main.xml

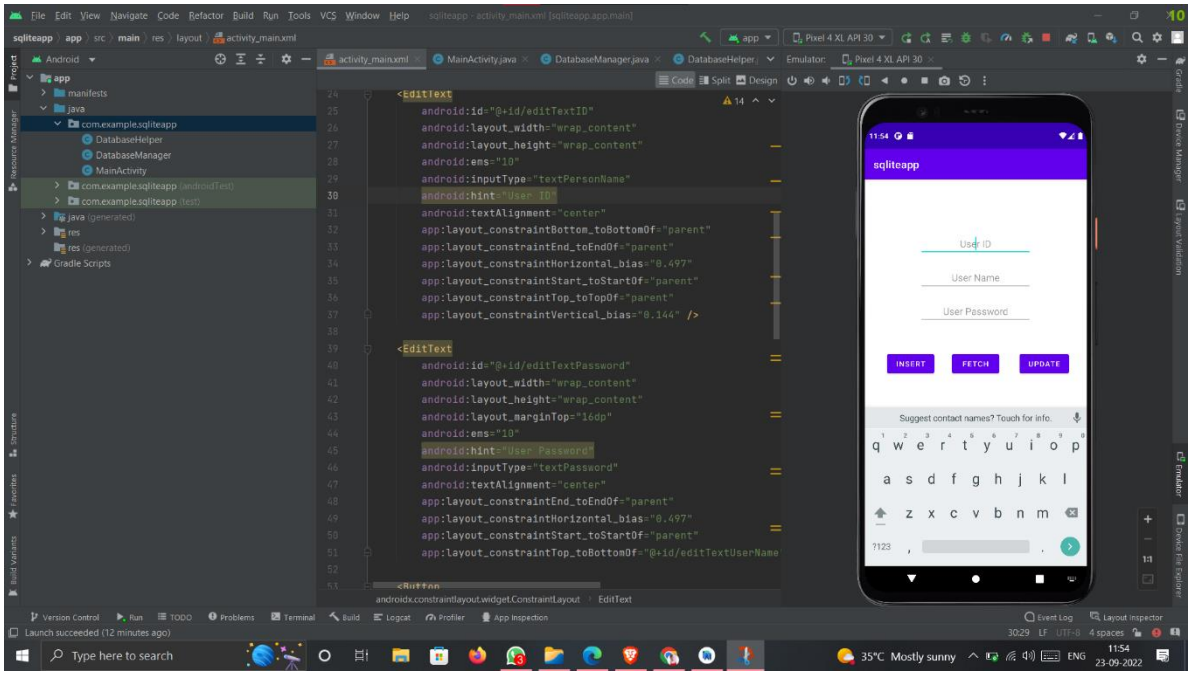
```
<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout
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"

    tools:context=".MainActivity">

    <EditText

        android:id="@+id/editTextUserName"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"
```

```
    android:layout_marginTop="16dp"
    android:ems="10"
    android:inputType="textPersonName"
    android:minHeight="48dp"
    android:hint="User Name"
    android:textAlignment="center"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editTextID" />
<EditText
    android:id="@+id/editTextID"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="textPersonName"
    android:hint="User ID"
    android:textAlignment="center"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.144" />
<EditText
    android:id="@+id/editTextPassword"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:ems="10"
    android:hint="User Password"
    android:inputType="textPassword"
    android:textAlignment="center"
    app:layout_constraintEnd_toEndOf="parent"
```

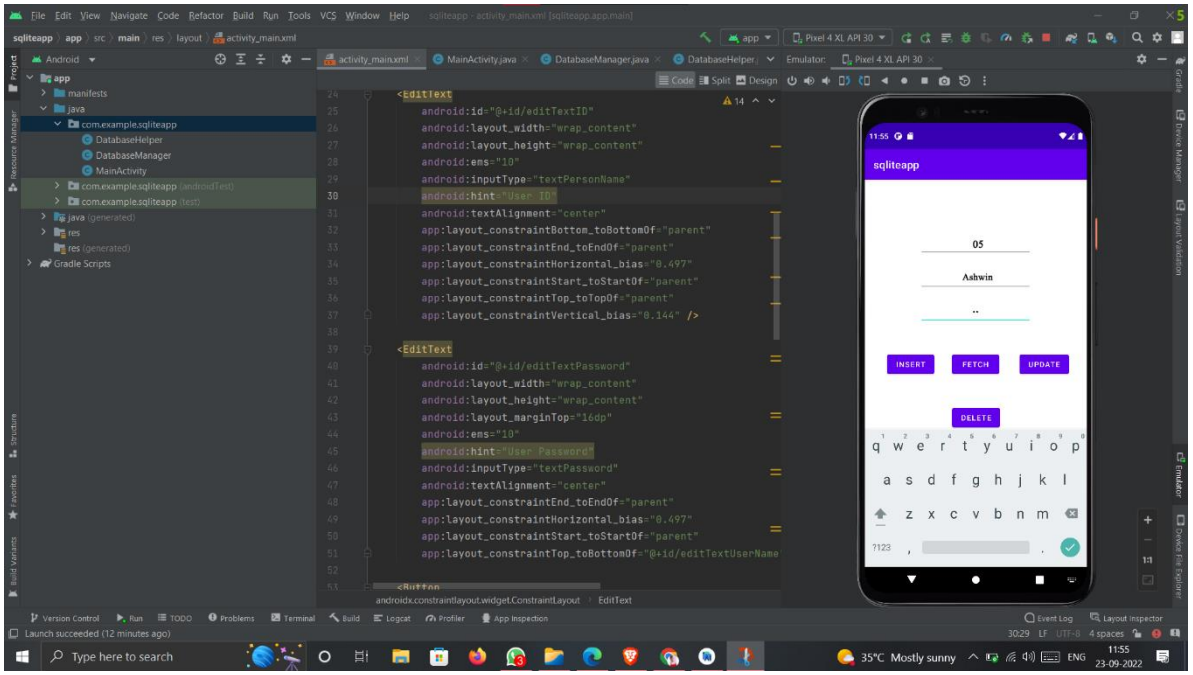
```
    app:layout_constraintHorizontal_bias="0.497"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editTextUserName" />
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="btnFetchPressed"
    android:text="Fetch"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.475" />
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="btnInsertPressed"
    android:text="Insert"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.126"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.475" />
<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="btnUpdatePressed"
    android:text="Update"
    app:layout_constraintBottom_toBottomOf="parent"
```

```
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.896"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.475" />
    <Button
        android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="btnDeletePressed"
        android:text="Delete"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.501"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.623" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

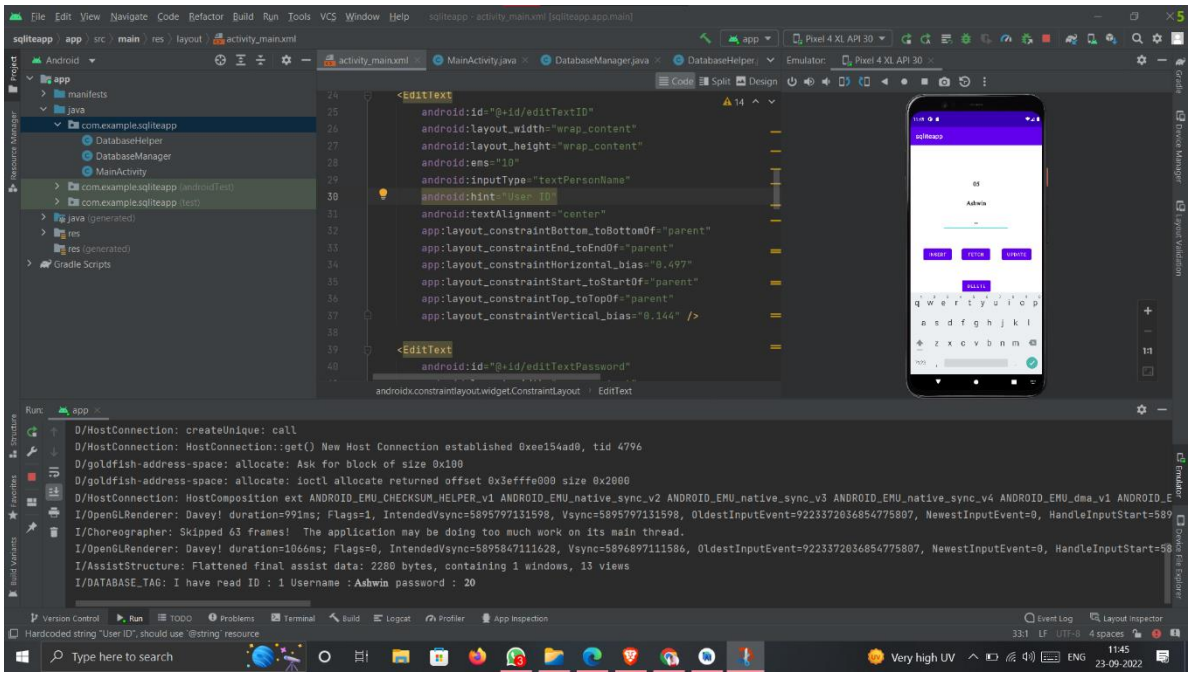
OUTPUT:



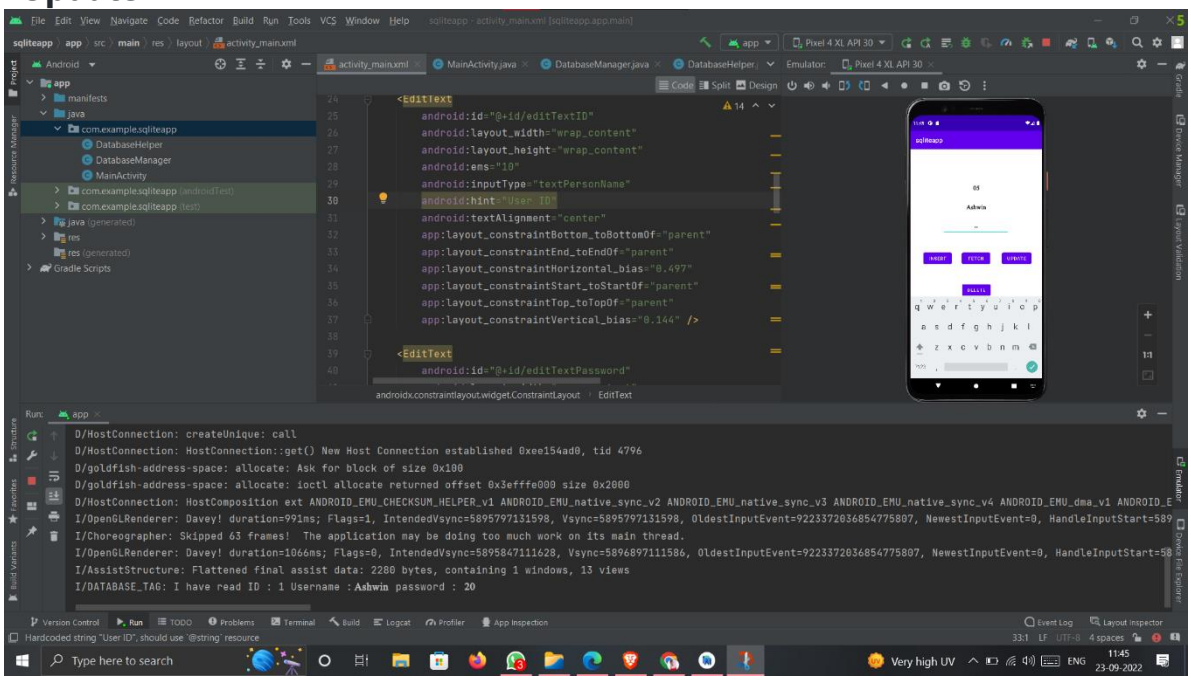
Insert



Fetch



Update



RESULT:

Thus, a Simple Android Application to create a database table and to display the database table using SQLite Database in Android Studio is developed and executed successfully.

Ex.No.: 02 Create a simple application client and server service using AIDL interface in android studio.

Date: 20.09.2022

AIM:

To create a AIDL interface and communicate the process between client and server using AIDL interface in Android Studio.

EQUIPMENTS REQUIRED:

Android Studio(Min.required Artic Fox)

ALGORITHM:

Step 1:

Open Android Stdio and then click on File -> New -> New project.

Step 2:

Then type the pplication name as CSAIDL and click Next.

Step 3:

Then select the Minimum SDK as shown below and click Next.

Step 4:

Then select the Empty Activity and click Next. Finally click Finish.

Step 5:

Design layout in activity_main.xml.

Step 6:

Display message give in MainActivity file(client/server).

Step 7:

Launch an emulator and run the application.

PROGRAM:

AIDL Server

```
package com.example.aidlserver;
```

```
import android.annotation.SuppressLint;

import android.app.Service;

import android.app.Service;

import android.content.Intent;

import android.graphics.Color;

import android.os.IBinder;

import android.os.RemoteException;

import android.util.Log;

import java.util.Random;

public class AIDLColorService extends Service {

    private static final String TAG = "AIDLColorService";

    public AIDLColorService() {

    }

    @Override

    public IBinder onBind(Intent intent) {

        // TODO: Return the communication channel to the service.

        return binder;

    }

    private final IAIDLColorinterface.Stub binder = new
IAIDLColorinterface.Stub() {

        @Override

        public int getcolor() throws RemoteException {

            Random rnd = new Random();

            int color = Color.argb(255, rnd.nextInt(256), rnd.nextInt(256),
rnd.nextInt(256));

            Log.d(TAG, "getColor: "+ color);

            return color;

        }

    }

}

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

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools"

    package="com.example.aidlserver">
```

```
    <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.AIDLServer"
        tools:targetApi="31">
        <service
            android:name=".AIDLColorService"
            android:enabled="true"
            android:exported="true">
            <intent-filter>
                <action android:name="AIDLColorService"/>
            </intent-filter>
        </service>
    </application>
</manifest>
```

AIDL Client

activity_Main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/button"
        android:layout_width="245dp"
```

```
        android:layout_height="273dp"
        android:text="Change the color"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.aidlclient;
import androidx.appcompat.app.AppCompatActivity;
import android.content.ComponentName;
import android.content.Intent;
import android.content.ServiceConnection;
import android.os.Bundle;
import android.os.IBinder;
import android.os.RemoteException;
import android.util.Log;
import  android.view.View;
import android.widget.Button;
import com.example.aidlclient.R;
import com.example.aidlserver.IAIDLColorinterface;
public class MainActivity extends AppCompatActivity {
    IAIDLColorinterface iADILColorService;
    private static final String TAG ="MainActivity";
    private ServiceConnection mConnection = new ServiceConnection() {
        @Override
        public void onServiceConnected(ComponentName componentName, IBinder
iBinder) {
            iADILColorService =
IAIDLColorinterface.Stub.asInterface(iBinder);
            Log.d(TAG, "Remote config Service Connected!!");
        }
        @Override
        public void onServiceDisconnected(ComponentName componentName) {
```

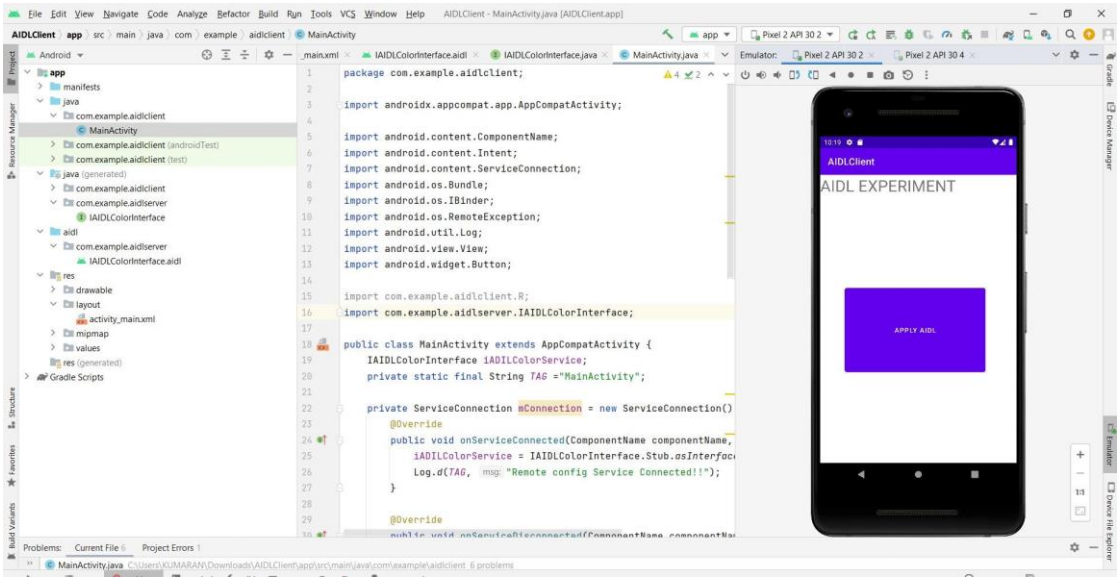
```
        }
    };
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Intent intent = new Intent("AIDLColorService");
        intent.setPackage("com.example.aidlserver");
        bindService(intent,mConnection, BIND_AUTO_CREATE);
        // Create an onclick listener to button
        Log.d(TAG, "bindservice called");
        Button b = findViewById(R.id.button);
        b.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                try {
                    int color = iADILColorService.getcolor();
                    view.setBackgroundColor(color);
                } catch (RemoteException e) {
                }
            }
        });
    }
}
```

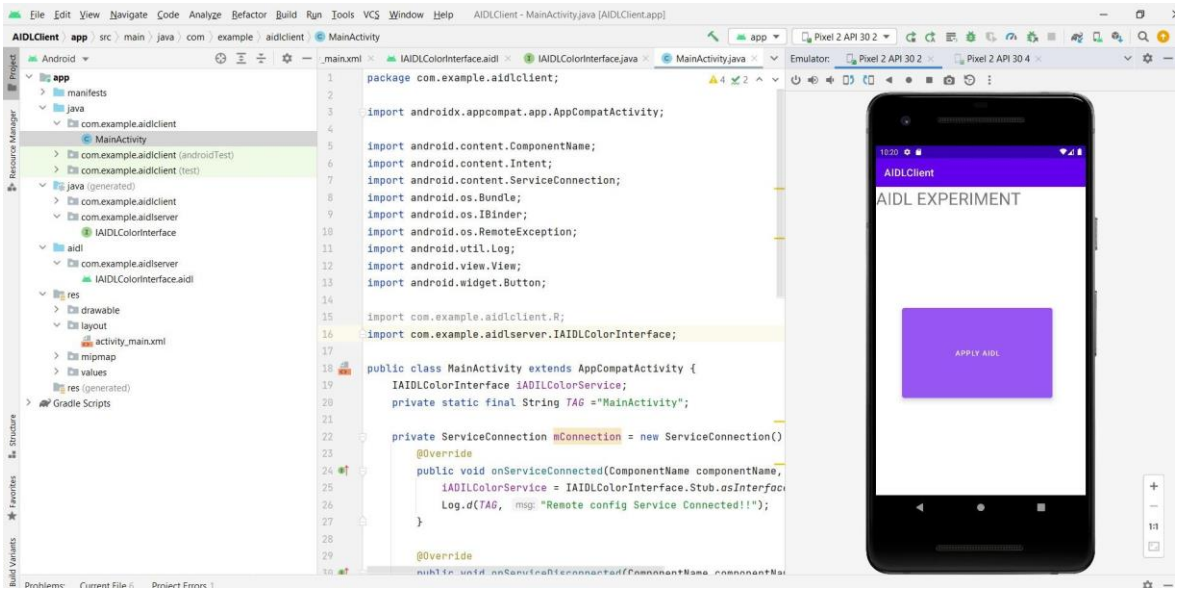
AndroidManifest.xml

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

OUTPUT





RESULT

Thus a Simple Android Application to create a AIDL interface and communicate the process between client and server using AIDL interface in Android Studio is developed and executed successfully.

Ex.No: 3 Develop a simple application to play and control the audio file in android studio.

Date: 11.10.2022

AIM:

To develop a simple application, to play and control the audio file and to perfrom the start,pause and stop opeartion in Android Studio.

EQUIPMENTS REQUIRED:

Android Studio(Min.required Arctic Fox)

ALGORITHM:

Step 1:

Open Android Studio and then click on File -> New -> New project.

Step 2:

Then type the Application name as audiofile and click Next.

Step 3:

Then select the Minimum SDK as shown below and click Next.

Step 4:

Then select the Empty Activity and click Next. Finally click Finish.

Step 5:

Design layout in activity_main.xml and create start,pause and stop button.

Step 6:

Display message give in MainActivity file.

Step 7:

Save and run the application.

PROGRAM:

MainActivity.java

```
package com.example.audio;
```

```
import androidx.appcompat.app.AppCompatActivity;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class MainActivity extends AppCompatActivity {
    Button b1,b2,b3;
    MediaPlayer mp;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        b1=findViewById(R.id.button);
        b2=findViewById(R.id.button2);
        b3=findViewById(R.id.button3);
        mp=MediaPlayer.create(getApplicationContext(),R.raw.music);
        b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.start();
            }
        });
        b2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.pause();
            }
        });
        b3.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mp.stop();
            }
        });
```

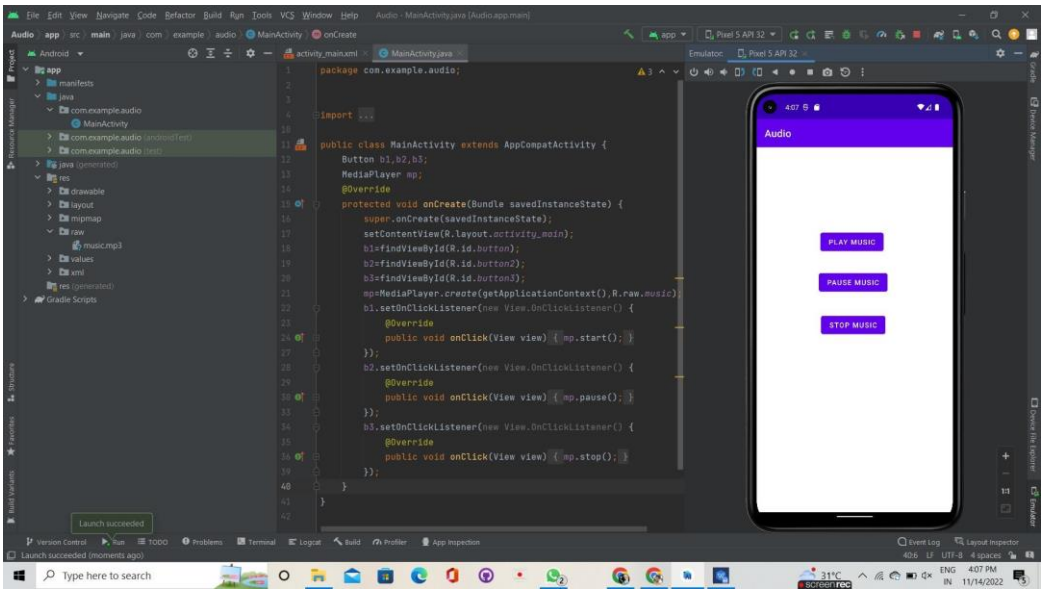
```
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="PLAY MUSIC"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.479"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.243" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="PAUSE MUSIC"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.488"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
```

```
        app:layout_constraintVertical_bias="0.361" />


    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="STOP MUSIC"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.488"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.486" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

OUTPUT:



RESULT:

Thus, a simple application to play and control the audio file and to perform the start,pause and stop opeartion in Android Studio is developed and executed successfully.

Ex.No: 4 Develop a simple application to display the available sensor in android mobile devices using Sensor Manager in android studio.

Date: 29.10.2022

AIM:

To develop a sensor application to use the sensor manager class to identify and get the list of available sensors on a device in Android Studio.

EQUIPMENTS REQUIRED:

Android Studio(Min.required Arctic Fox)

ALGORITHM:

Step 1:

Open Android Studio and then click on File -> New -> New project.

Step 2:

Then type the Application name as Sensor and click Next.

Step 3:

Then select the Minimum SDK as shown below and click Next.

Step 4:

Then select the Empty Activity and click Next. Finally click Finish.

Step 5:

Design layout in activity_main.xml.

Step 6:

Display avaliable sensor in android mobile devices.

Step 7:

Launch an emulator and run the application.

PROGRAM:

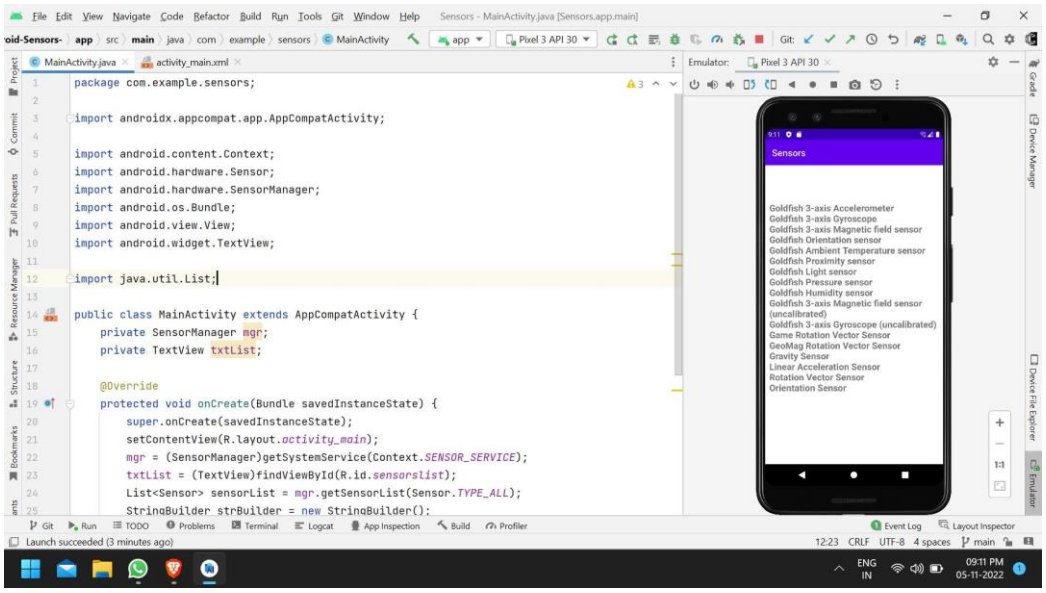
MainActivity.java

```
package com.example.sensors;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Context;
import android.hardware.Sensor;
import android.hardware.SensorManager;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import java.util.List;
public class MainActivity extends AppCompatActivity {
    private SensorManager mgr;
    private TextView txtList;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mgr = (SensorManager)getSystemService(Context.SENSOR_SERVICE);
        txtList = (TextView)findViewById(R.id.sensorslist);
        List<Sensor> sensorList = mgr.getSensorList(Sensor.TYPE_ALL);
        StringBuilder strBuilder = new StringBuilder();
        for(Sensor s: sensorList){
            strBuilder.append(s.getName()+"\n");
        }
        txtList.setVisibility(View.VISIBLE);
        txtList.setText(strBuilder);
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="10dp"
    android:paddingRight="10dp">
    <TextView
        android:id="@+id/sensorslist"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="80dp"
        android:text="Sensors"
        android:textSize="20dp"
        android:textStyle="bold"
        android:layout_gravity="center"
        android:visibility="gone"/>
</LinearLayout>
```

OUTPUT:



RESULT:

Thus, a Simple Android Application to display the avaliable sensor in android mobile devices using Sensor Manager in Android Studio is developed and executed successfully.

Ex.No: 5 Develop a simple application for proximity sensor using Sensor Manager in android studio.

Date: 09.11.2022

AIM:

To develop a sensor application for proximity sensor using sensor manager in Android Studio.

EQUIPMENTS REQUIRED:

Android Studio(Min.required Arctic Fox)

ALGORITHM:

Step 1:

Open Android Studio and then click on File -> New -> New project.

Step 2:

Then type the Application name as proximitysensor and click Next.

Step 3:

Then select the Minimum SDK as shown below and click Next.

Step 4:

Then select the Empty Activity and click Next. Finally click Finish.

Step 5:

Design layout in activity_main.xml.

Step 6:

Display process of proximity sensor in android mobile devices.

Step 7:

Launch an emulator and run the application.

PROGRAM:

MainActivity.java

```
package com.example.proximitysensor;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;import android.hardware.Sensor;import
android.hardware.SensorEvent;import android.hardware.SensorEventListener;import
android.hardware.SensorManager;import android.os.Bundle;import
android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    TextView ProximitySensor, data;

    SensorManager mySensorManager;

    Sensor myProximitySensor;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        ProximitySensor = (TextView) findViewById(R.id.proximitySensor);

        data = (TextView) findViewById(R.id.data);

        mySensorManager = (SensorManager) getSystemService(

                Context.SENSOR_SERVICE);

        myProximitySensor = mySensorManager.getDefaultSensor(

                Sensor.TYPE_PROXIMITY);

        if (myProximitySensor == null) {

            ProximitySensor.setText("No Proximity Sensor!");

        } else {

            mySensorManager.registerListener(proximitySensorEventListener,

                    myProximitySensor,

                    SensorManager.SENSOR_DELAY_NORMAL);

        }

    }

    SensorEventListener proximitySensorEventListener

            = new SensorEventListener() {
```

```
        @Override
        public void onAccuracyChanged(Sensor sensor, int accuracy) {
            // TODO Auto-generated method stub
        }
        @Override
        public void onSensorChanged(SensorEvent event) {
            // TODO Auto-generated method stub
            if (event.sensor.getType() == Sensor.TYPE_PROXIMITY) {
                if (event.values[0] == 0) {
                    data.setText("Near");
                } else {
                    data.setText("Away");
                }
            }
        }
    };
}
```

AndroidManifest.xml

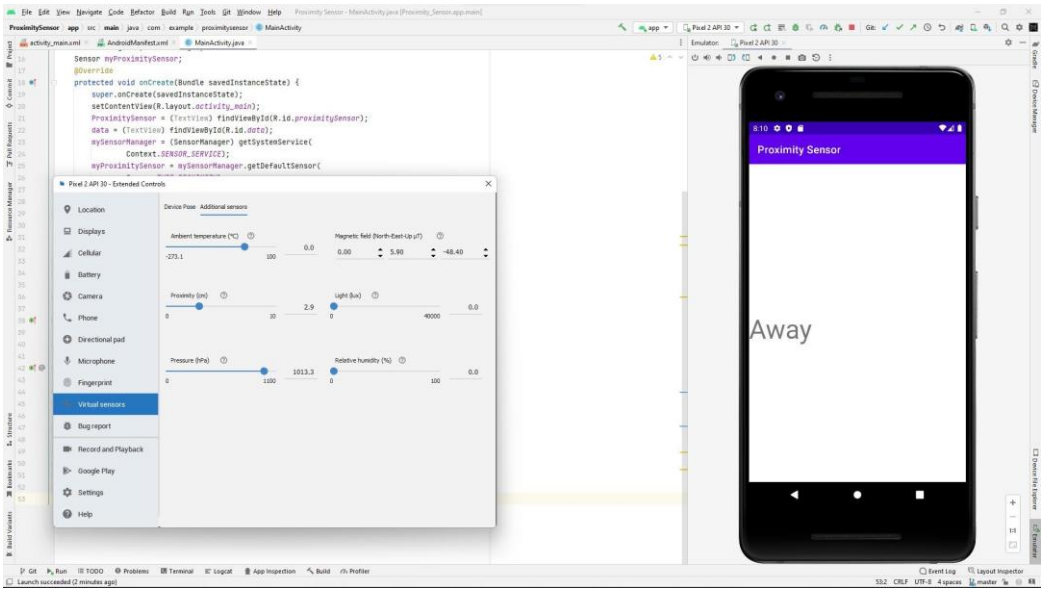
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.hardware.sensor.proximity"/>
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        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.ProximitySensor"
        tools:targetApi="31">
        <activity
```

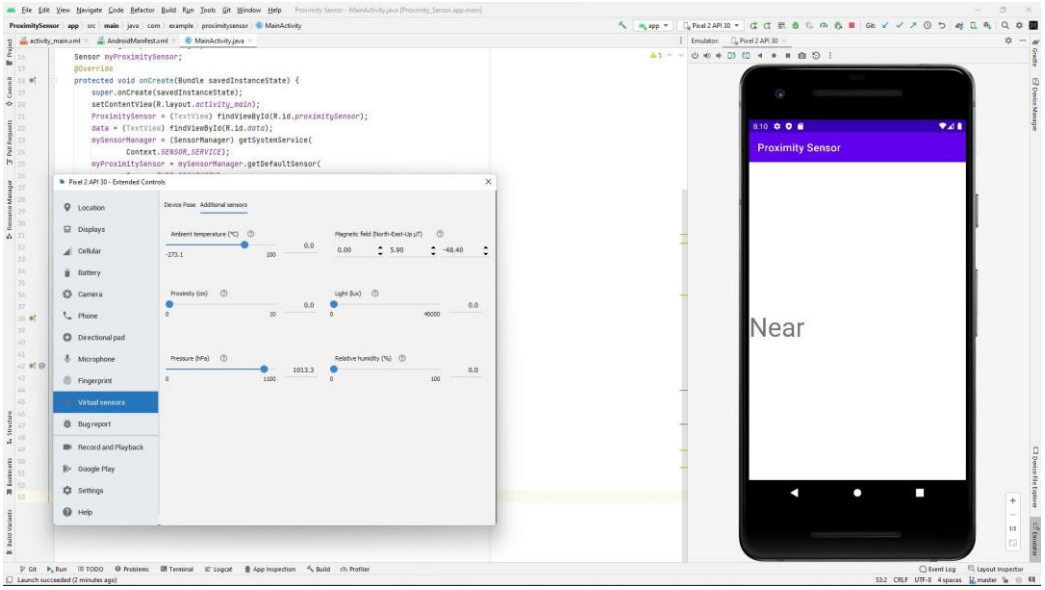
```
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
            <meta-data
                android:name="android.app.lib_name"
                android:value="" />
        </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="fill_parent"
    android:layout_height="fill_parent"
    android:gravity="center"
    android:orientation="vertical">
    <TextView
        android:id="@+id/proximitySensor"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
    <TextView
        android:id="@+id/data"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="50dp" />
</LinearLayout>
```

OUTPUT:





RESULT:

Thus, a Simple Android Application to display the process of proximity sensor using sensor manager in Android Studio is developed and executed successfully.

Ex.No: 6 Create a simple application to request storage and camera permission at RunTime using android studio.

Date: 15.11.2022

AIM:

To develop a simple application for RunTime Permission in Android Studio.

EQUIPMENTS REQUIRED:

Android Studio(Min.required Arctic Fox)

ALGORITHM:

Step 1: Open Android Studio and then click on File -> New -> New project.

Step 2: Then type the Application name as runtimepermission and click Next.

Step 3: Then select the Minimum SDK as shown below and click Next.

Step 4: Then select the Empty Activity and click Next. Finally click Finish.

Step 5: Design layout in activity_main.xml.

Step 6: Display process of runtimepermission in android mobile devices.

Step 7: Save and run the application.

PROGRAM:

MainActivity.java

```
package com.example.runtimepermission;

import android.Manifest;

import android.content.pm.PackageManager;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import androidx.core.content.ContextCompat;

public class MainActivity extends AppCompatActivity {
```

```
    private Button storage, camera;
    private static final int CAMERA_PERMISSION_CODE = 100;
    private static final int STORAGE_PERMISSION_CODE = 101;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        storage = findViewById(R.id.storage);
        camera = findViewById(R.id.camera);
        storage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v)
            {
                checkPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE,
STORAGE_PERMISSION_CODE);
            }
        });
        camera.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v)
            {
                checkPermission(Manifest.permission.CAMERA,
CAMERA_PERMISSION_CODE);
            }
        });
    }
    public void checkPermission(String permission, int requestCode)
    {
        if (ContextCompat.checkSelfPermission(MainActivity.this, permission) ==
PackageManager.PERMISSION_DENIED) {

            // Requesting the permission
            ActivityCompat.requestPermissions(MainActivity.this, new String[]
{ permission }, requestCode);
```

```
        }
        else {
            Toast.makeText(MainActivity.this, "Permission already granted",
Toast.LENGTH_SHORT).show();
        }
    }
    @Override
    public void onRequestPermissionsResult(int requestCode,
                                           @NonNull String[] permissions,
                                           @NonNull int[] grantResults)
    {
        super.onRequestPermissionsResult(requestCode,
                permissions,
                grantResults);
        if (requestCode == CAMERA_PERMISSION_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                Toast.makeText(MainActivity.this, "Camera Permission Granted",
Toast.LENGTH_SHORT) .show();
            }
            else {
                Toast.makeText(MainActivity.this, "Camera Permission Denied",
Toast.LENGTH_SHORT) .show();
            }
        }
        else if (requestCode == STORAGE_PERMISSION_CODE) {
            if (grantResults.length > 0
                    && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                Toast.makeText(MainActivity.this, "Storage Permission Granted",
Toast.LENGTH_SHORT).show();
            } else {
                Toast.makeText(MainActivity.this, "Storage Permission Denied",
Toast.LENGTH_SHORT).show();
            }
        }
    }
```

```
}
```

activity_main.xml

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

<androidx.constraintlayout.widget.ConstraintLayout
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"

    tools:context=".MainActivity">

    <Button

        android:id="@+id/storage"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_centerHorizontal="true"

        android:padding="8dp"

        android:text="Request Storage Permission"

        app:layout_constraintBottom_toTopOf="@+id/camera"

        app:layout_constraintEnd_toEndOf="parent"

        app:layout_constraintStart_toStartOf="parent"

        app:layout_constraintTop_toTopOf="parent"

        app:layout_constraintVertical_bias="0.837" />

    <Button

        android:id="@+id/camera"

        android:layout_width="wrap_content"

        android:layout_height="wrap_content"

        android:layout_below="@id/storage"

        android:layout_centerHorizontal="true"

        android:layout_marginBottom="260dp"

        android:padding="8dp"

        android:text="Request Camera Permission"

        app:layout_constraintBottom_toBottomOf="parent"

        app:layout_constraintEnd_toEndOf="parent"

        app:layout_constraintHorizontal_bias="0.522"
```

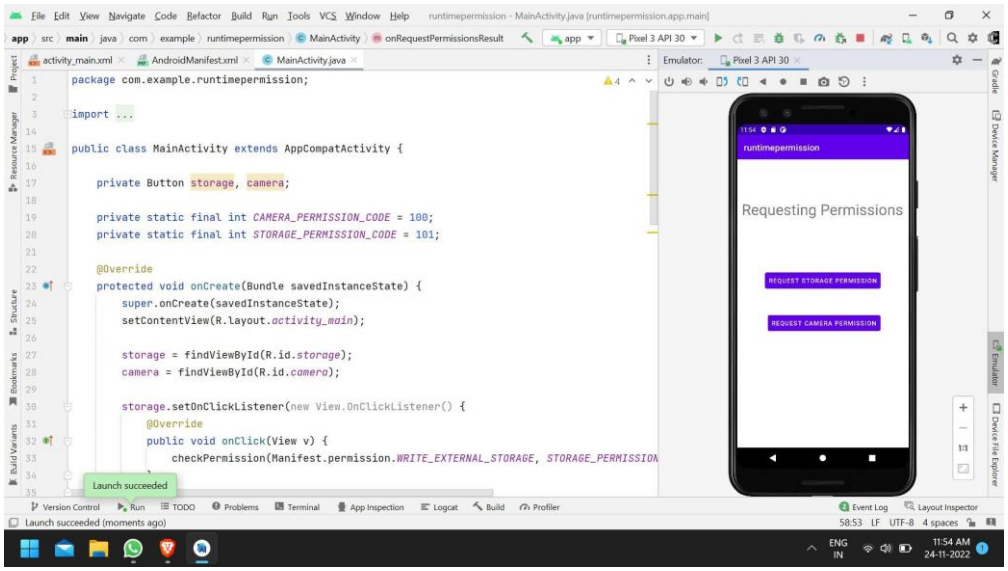
```
        app:layout_constraintStart_toStartOf="parent" />
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Requesting Permissions"
        android:textSize="34sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.488"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.148" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

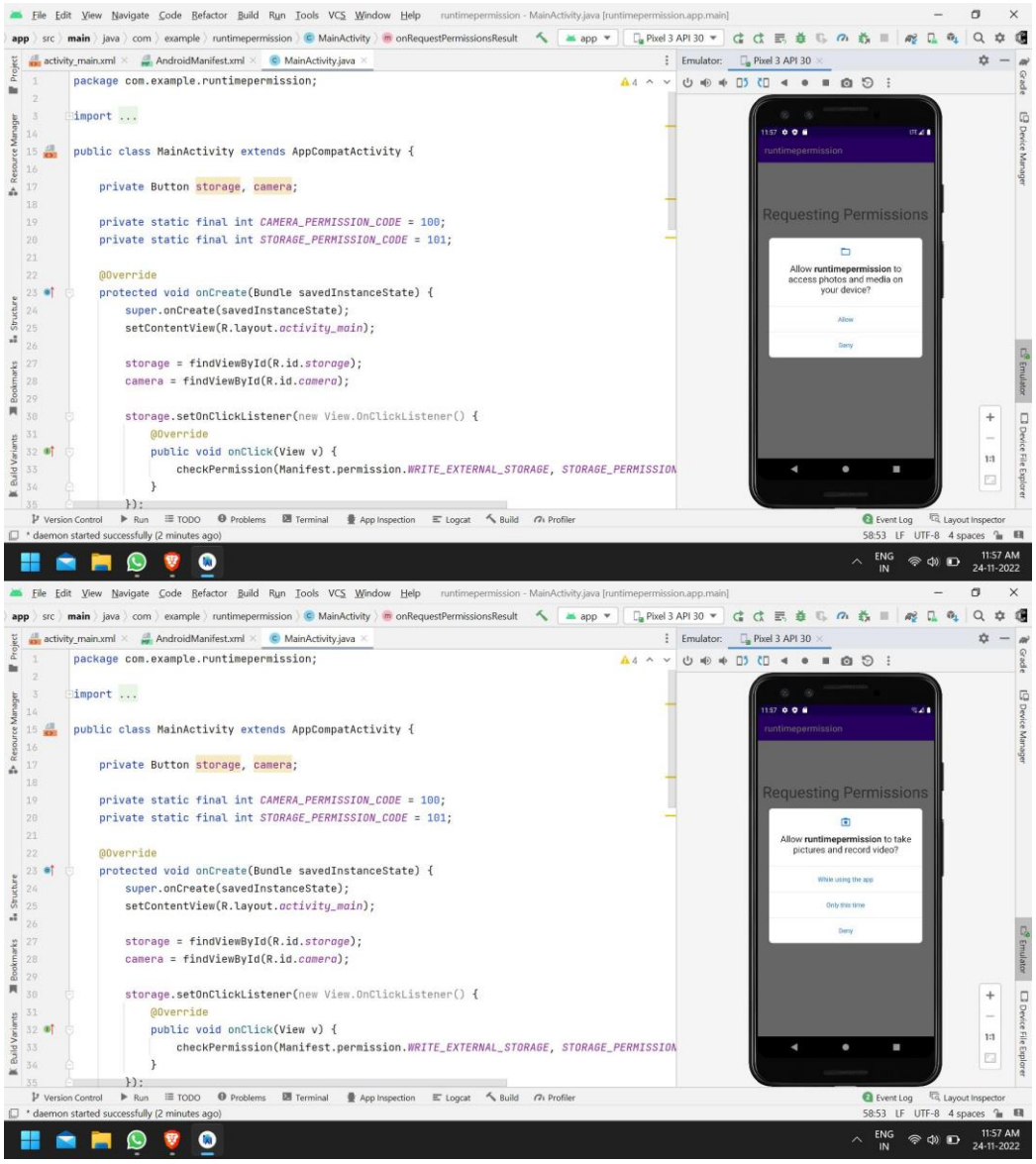
AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <!--Declaring the required permissions-->
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
/>
    <uses-permission android:name="android.permission.CAMERA" />
    <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.RuntimePermission"
        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>
            <meta-data
                android:name="android.app.lib_name"
                android:value="" />
        </activity>
    </application>
</manifest>
```

OUTPUT:





RESULT:

Thus, a Simple Android Application to request storage and camera permission at RunTime in Android Studio is developed and executed successfully.