ASSIGNMENT 3:

DATABASE CONNECTIVITY & REST API INTEGRATION

Name: Sayli Rajendra Dholam

Roll No: 16

Subject: Mobile Computing Lab

Semester: III

Division: A

Q1. Write a program to read and write content in internal storage.

Aim: To create an Android application that reads and writes content to the internal storage of the device.

Objective: Learn how to read from and write data into the internal storage. Understand the use of FileInputStream and FileOutputStream for internal storage operations.

Theory: Internal Storage is a private storage space available for an app. Data stored here is not accessible by other apps, and it persists even if the app is closed. Files are stored in the app's private directory, and when the app is uninstalled, the stored data is deleted.

FileOutputStream fos: Used to write data to a file in the internal storage. openFileOutput() opens the file for writing.

FileInputStream fis: Used to read data from a file in the internal storage. openFileInput() opens the file for reading.

InputStreamReader: Converts byte stream (from FileInputStream) into a character stream, allowing text manipulation.

BufferedReader: Reads text from a character-input stream and buffers the characters, improving reading efficiency.

StringBuilder: Collects and efficiently manipulates strings.

Code:

MainActivity.xml

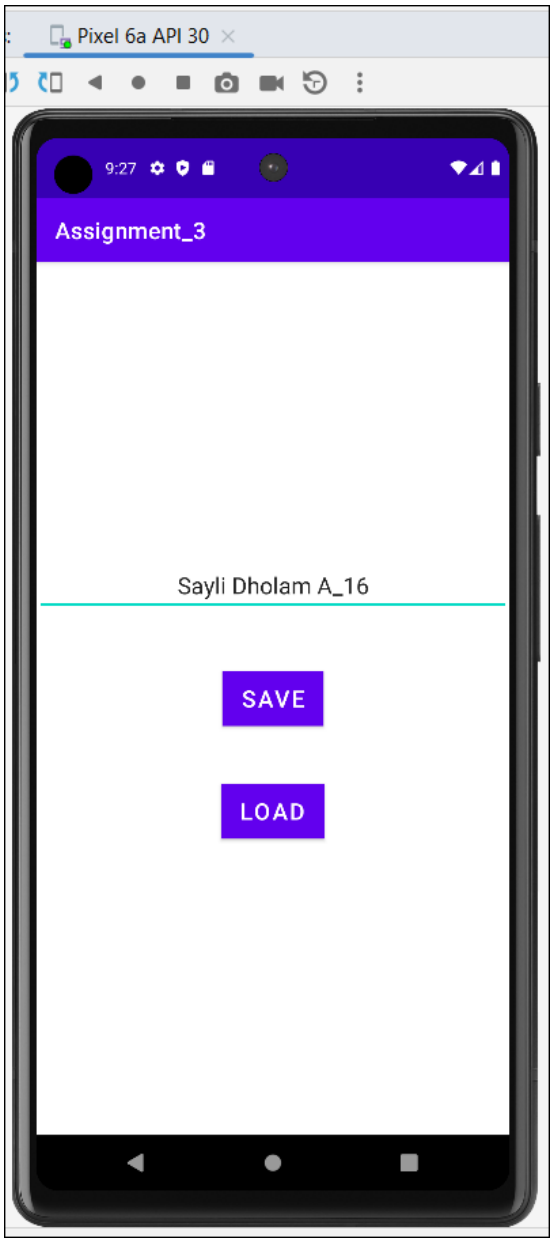
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_gravity="center"
    android:gravity="center"
    tools:context=".MainActivity_internalstrg">

    <EditText
        android:id="@+id/text1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="20dp"
        android:hint="Enter Data" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btn1"
        android:layout_gravity="center"
        android:layout_marginTop="50dp"
        android:textSize="20dp"
        android:background="#6691A5"
        android:text="Save"
        android:onClick="safe"/>

    <Button
        android:id="@+id/btn2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="50dp"
        android:background="#6691A5"
        android:text="Load"
        android:onClick="loadMethod"
        android:textSize="20dp" />

</LinearLayout>
```

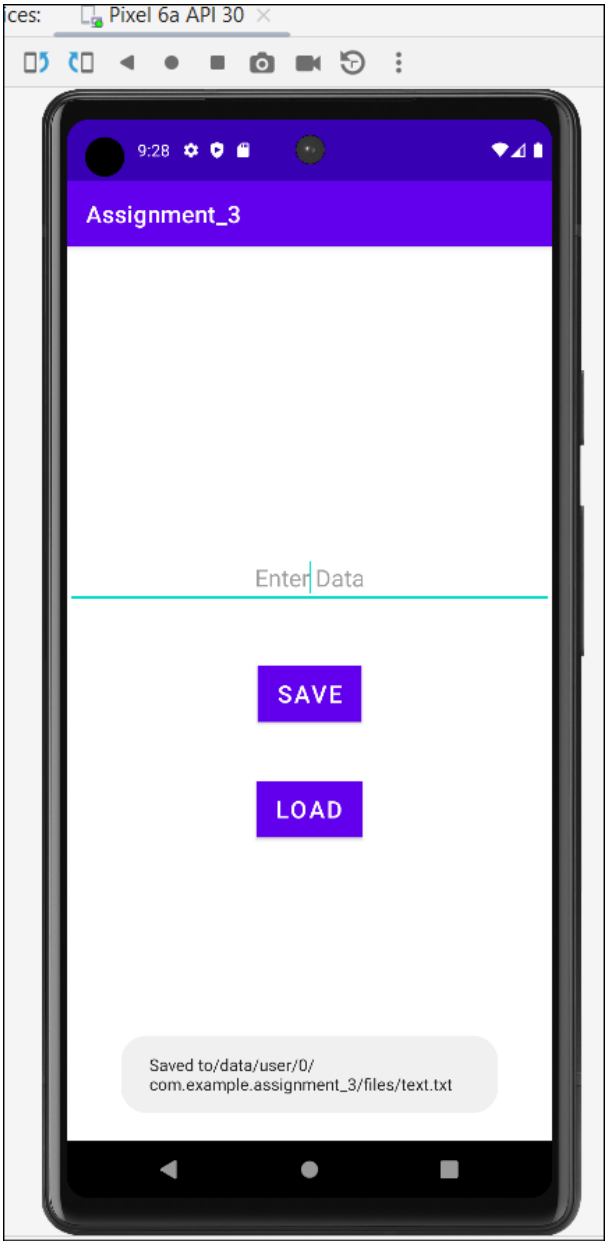
MainActivity.java

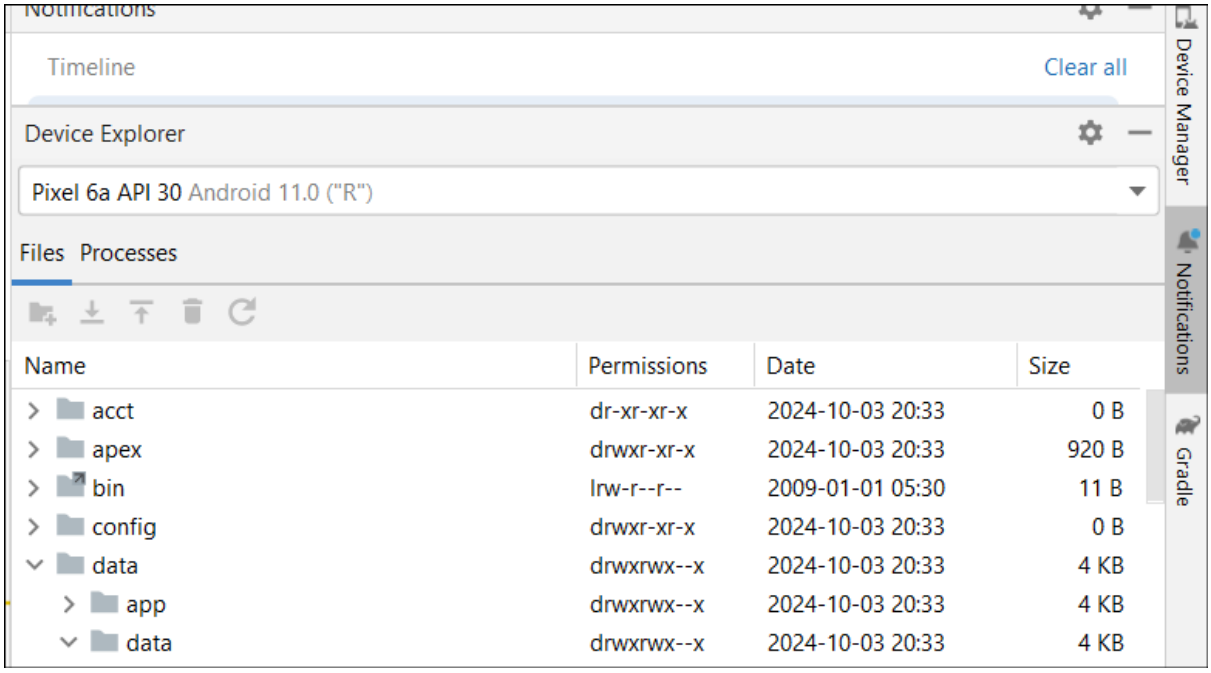
```
package com.example.assignment_3;

import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;

public class MainActivity_internalstrg extends
AppCompatActivity {

    EditText mEditText;
    private static final String FILE_NAME="text.txt";
    @SuppressLint("MissingInflatedId")

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_internalstrg);
        mEditText=findViewById(R.id.text1);
    }

    public void safe(View view) throws IOException {
        String text=mEditText.getText().toString();
        FileOutputStream fos=null;
        try {
            fos=openFileOutput(FILE_NAME,MODE_PRIVATE);
            fos.write(text.getBytes());
            mEditText.getText().clear();
            Toast.makeText(this, "Saved
to"+getFilesDir()+"/"+FILE_NAME, Toast.LENGTH_LONG).show();
        }catch (FileNotFoundException e){
            e.printStackTrace();
        }catch (IOException e){
            e.printStackTrace();
        }finally {
            if (fos!=null){
                fos.close();
```

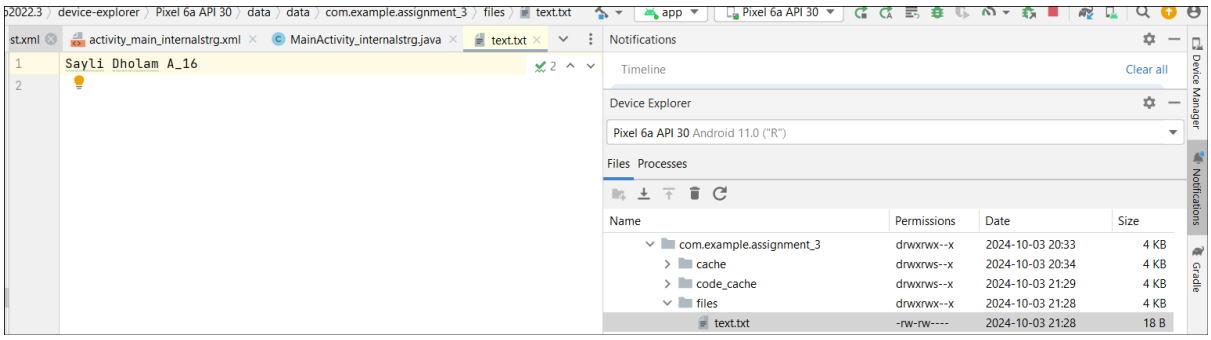
```
            }
        }
    }
    public void loadMethod(View view){
        FileInputStream fis = null;
        try{
            fis = openFileInput(FILE_NAME);
            InputStreamReader isr = new
InputStreamReader(fis);
            BufferedReader br = new BufferedReader(isr);
            StringBuilder sb = new StringBuilder();
            String text;
            while((text = br.readLine()) != null){
                sb.append(text).append("\n");
            }
            mEditText.setText(sb.toString());
        }
        catch (FileNotFoundException e) {
            e.printStackTrace();
        }
        catch (IOException e) {
            e.printStackTrace();
        }
        finally {
            if(fis != null){
                try{
                    fis.close();
                }
                catch(IOException e){
                    e.printStackTrace();
                }
            }
        }
    }
}
```

Output:


Pixel 6a API 30
9:27
Assignment_3
Sayli Dholam A_16
SAVE
LOAD


Pixel 6a API 30
9:28
Assignment_3
Enter Data
SAVE
LOAD
Saved to/data/user/0/
com.example.assignment_3/files/text.txt


Notifications
Timeline
Clear all
Device Explorer
Pixel 6a API 30 Android 11.0 ("R")
Files Processes
Name
Permissions
Date
Size
acct
dr-xr-xr-x
2024-10-03 20:33
0 B
apex
drwxr-xr-x
2024-10-03 20:33
920 B
bin
lrw-r--r--
2009-01-01 05:30
11 B
config
drwxr-xr-x
2024-10-03 20:33
0 B
data
drwxrwx--x
2024-10-03 20:33
4 KB
app
drwxrwx--x
2024-10-03 20:33
4 KB
data
drwxrwx--x
2024-10-03 20:33
4 KB
Device Manager
Notifications
Gradle


device-explorer
Pixel 6a API 30
data
com.example.assignment_3
files
text.txt
activity_main_internalstrg.xml
MainActivity_internalstrg.java
1
Sayli Dholam A_16
2
Notifications
Timeline
Clear all
Device Explorer
Pixel 6a API 30 Android 11.0 ("R")
Files Processes
Name
Permissions
Date
Size
com.example.assignment_3
drwxrwx--x
2024-10-03 20:33
4 KB
cache
drwxrws--x
2024-10-03 20:34
4 KB
code_cache
drwxrws--x
2024-10-03 21:29
4 KB
files
drwxrwx--x
2024-10-03 21:28
4 KB
text.txt
-rw-rw----
2024-10-03 21:28
18 B

Q2. Write a program to read and write content in external storage.

Aim: To create an Android application that reads and writes data to external storage.

Objective: To understand how to manage file operations, including permissions for reading and writing to external storage in Android devices.

Theory: External storage in Android is available for the app and the user to store and manage data like media files, documents, and other large data files. Accessing external storage requires permissions to be granted by the user. The external storage can either be removable (SD card) or non-removable (shared internal storage).

Permissions:

You need to include READ_EXTERNAL_STORAGE and WRITE_EXTERNAL_STORAGE in the AndroidManifest.xml file, as accessing external storage requires explicit permissions from the user.

Environment.getExternalStorageDirectory():

This method retrieves the root directory of the external storage.

FileOutputStream/FileInputStream:

These classes are used for writing and reading the file in external storage.

saveData():

This method writes the contents from the EditText field to a file in external storage.

loadData():

This method reads the contents of the file from external storage and displays it in the EditText.

Code:

AndroidManifest.xml

```
<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

MainActivity.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:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_gravity="center"
    android:gravity="center"
    tools:context=".MainActivity_externalstrg">

    <EditText
        android:id="@+id/text1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="20dp"
        android:hint="Enter Data" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btn1"
        android:layout_gravity="center"
        android:layout_marginTop="50dp"
        android:textSize="20dp"
        android:background="#6691A5"
        android:text="Save"
        android:onClick="safe"/>

    <Button
        android:id="@+id/btn2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="50dp"
```

```
        android:background="#6691A5"
        android:text="Load"
        android:onClick="load"
        android:textSize="20dp" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textview"
        android:textSize="20dp"
        />

</LinearLayout>
```

MainActivity.java

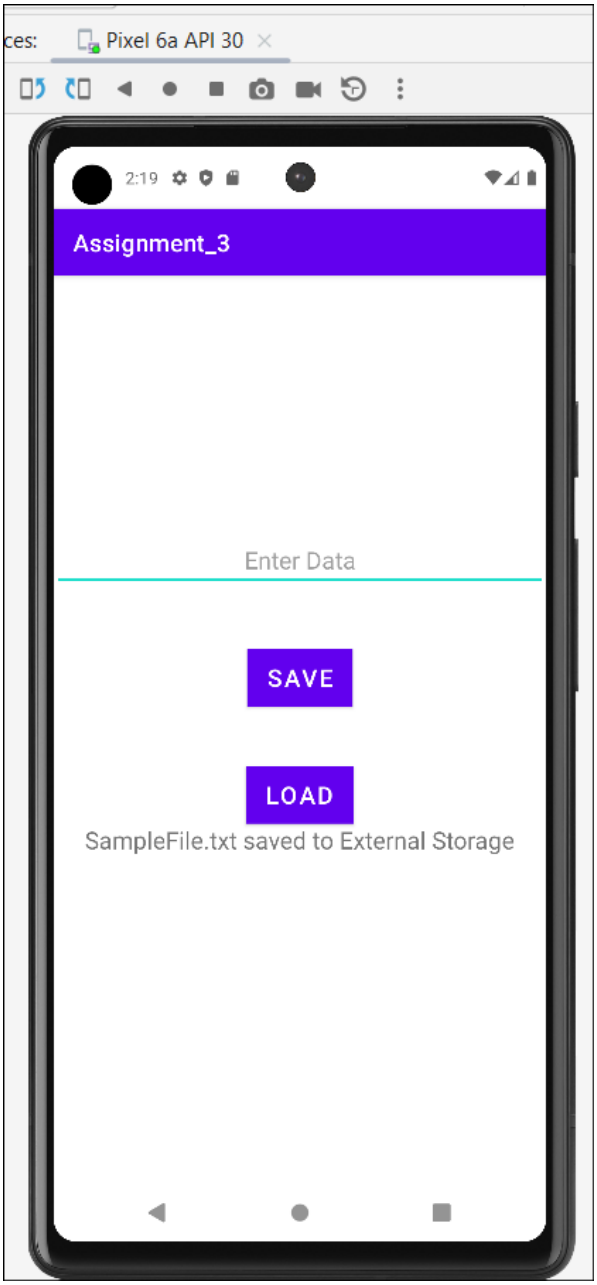
```
package com.example.assignment_3;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.os.Bundle;
import android.os.Environment;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;

public class MainActivity_externalstrg extends
AppCompatActivity {

    EditText mEditText;
    TextView t1;
    Button saveButton,readButton;
    private String filename="Samplefile.txt";
    private String filepath="MyFileStroage";
    File myExternalFile;
    String myData;

    @SuppressLint("MissingInflatedId")
```

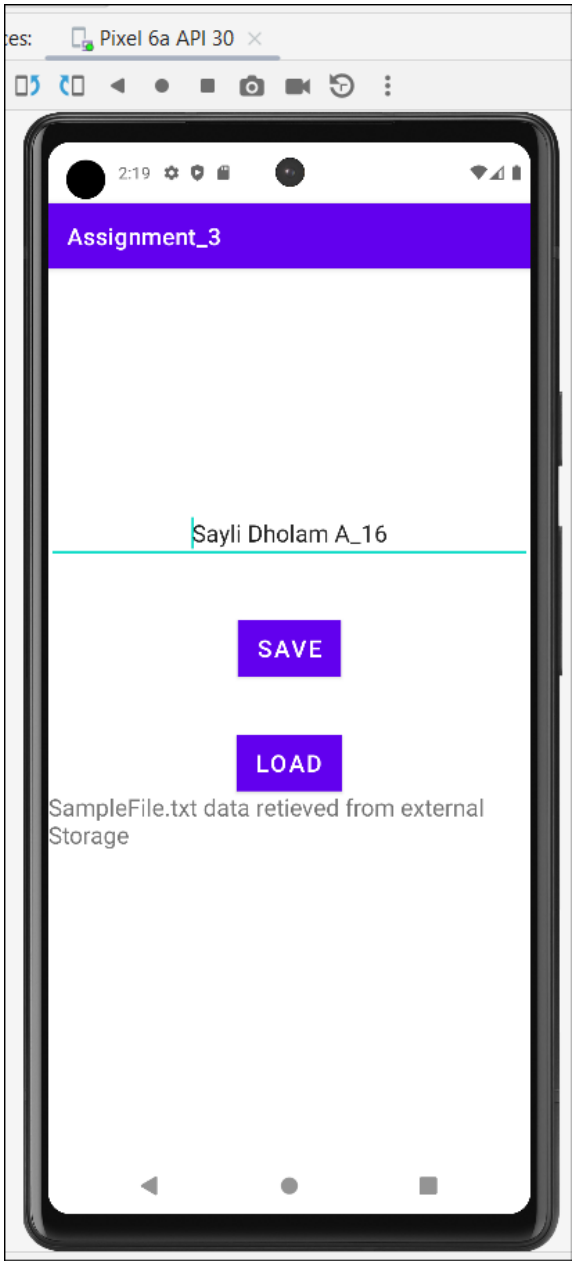
```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main_externalstrg);

        mEditText=findViewById(R.id.text1);
        t1=findViewById(R.id.textview);
        saveButton=findViewById(R.id.btn1);
        readButton=findViewById(R.id.btn2);

        if
(!isExternalStorageAvailable()||isExternalStorageReadOnly()){
            saveButton.setEnabled(false);
        }else {
            myExternalFile=new
File(getExternalFilesDir(filepath),filename);
        }

    }
    private boolean isExternalStorageReadOnly() {
        String extStorageState=
Environment.getExternalStorageState();
        if
(Environment.MEDIA_MOUNTED_READ_ONLY.equals(extStorageState)){
            return true;
        }
        return false;
    }
    private boolean isExternalStorageAvailable() {
        String extStorageState=
Environment.getExternalStorageState();
        if
(Environment.MEDIA_MOUNTED.equals(extStorageState)){
            return true;
        }
        return false;
    }
    public void safe(View view){
        try {
            FileOutputStream fos=new
FileOutputStream(myExternalFile);

fos.write(mEditText.getText().toString().getBytes());
            fos.close();
        }catch (Exception e){
            e.printStackTrace();
        }
        mEditText.setText("");
        t1.setText("SampleFile.txt saved to External
Storage");
```

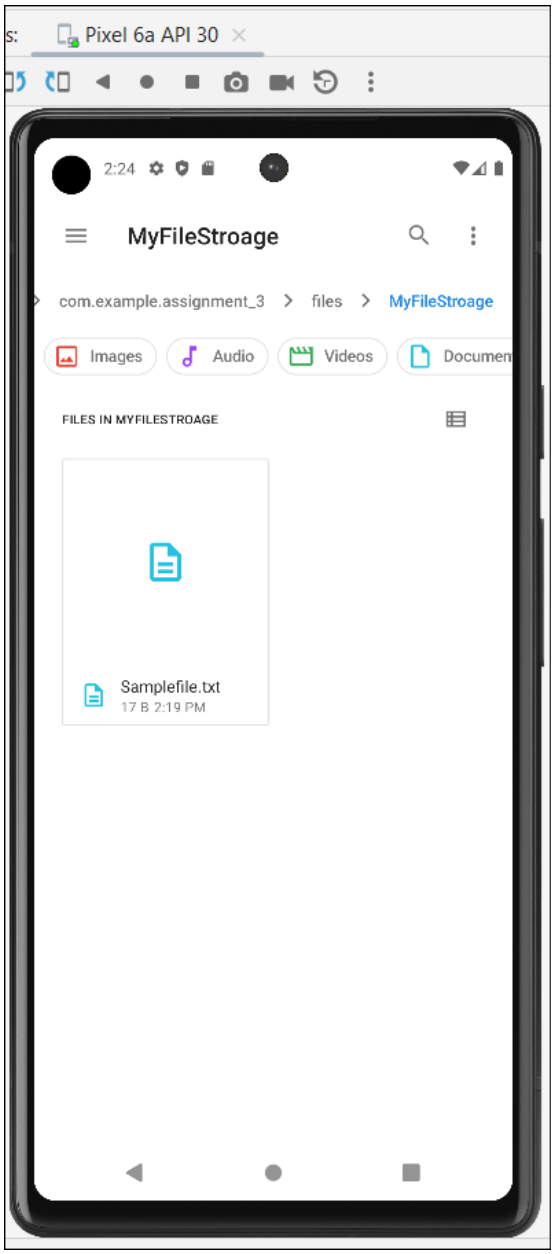
```
    }
    public void load(View view){
        try {
            FileInputStream fis=new
FileInputStream(myExternalFile);
            InputStreamReader i=new InputStreamReader(fis);
            BufferedReader br =new BufferedReader(i);
            String strLine;
            while((strLine=br.readLine())!=null){
                myData=strLine;
            }
            i.close();
        } catch (FileNotFoundException e) {
            throw new RuntimeException(e);
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
        mEditText.setText(myData);
        t1.setText("SampleFile.txt data retieved from external
Storage");
    }
}
```

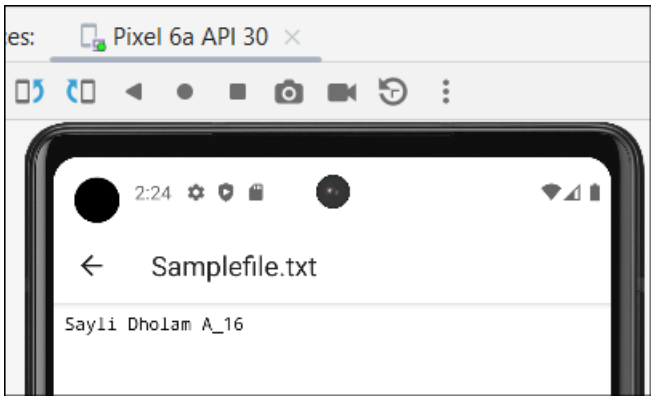
Output:


Pixel 6a API 30
2:18
Assignment_3
Sayli Dholam A_16
SAVE
LOAD


Pixel 6a API 30
2:19
Assignment_3
Enter Data
SAVE
LOAD
SampleFile.txt saved to External Storage







All apps > files > menu > Sdk_gphone_x68 > android > data > com.example.assignment_3 > files > myFileStorage > SampleFile.txt

Q3. Write a program to demonstrate shared preference.

Aim: To create an Android application that demonstrates storing and retrieving, user data using Shared Preferences.

Objective: The objective of this program is to understand how to use Shared Preferences to store key-value pairs, which can be used to retain small amounts of primitive data such as user settings or preferences.

Theory: A Shared Preferences is a lightweight storage option in Android for storing small amounts of primitive data (like boolean, float, int, long, string) in the form of key-value pairs. It's suitable for persisting user settings, application configurations, and other user-related preferences. Shared Preferences persist across app restarts and do not require explicit read/write permissions.

SharedPreferences: Used to store small amounts of data in the form of key-value pairs. This data persists across user sessions and app restarts.

getSharedPreferences(): This method retrieves an instance of SharedPreferences, allowing access to saved data. MODE_PRIVATE ensures that the data is accessible only to your app.

SharedPreferences.Editor: Used to make changes to SharedPreferences. You must use editor.putString() or similar methods to store data and call apply() or commit() to save the changes.

apply() vs commit():

- apply() is asynchronous and doesn't return any result.
- commit() is synchronous and returns a boolean indicating whether the save was successful.

loadData(): Loads stored data from SharedPreferences and displays it in the TextView. If no data is found, default values are shown.

Code:

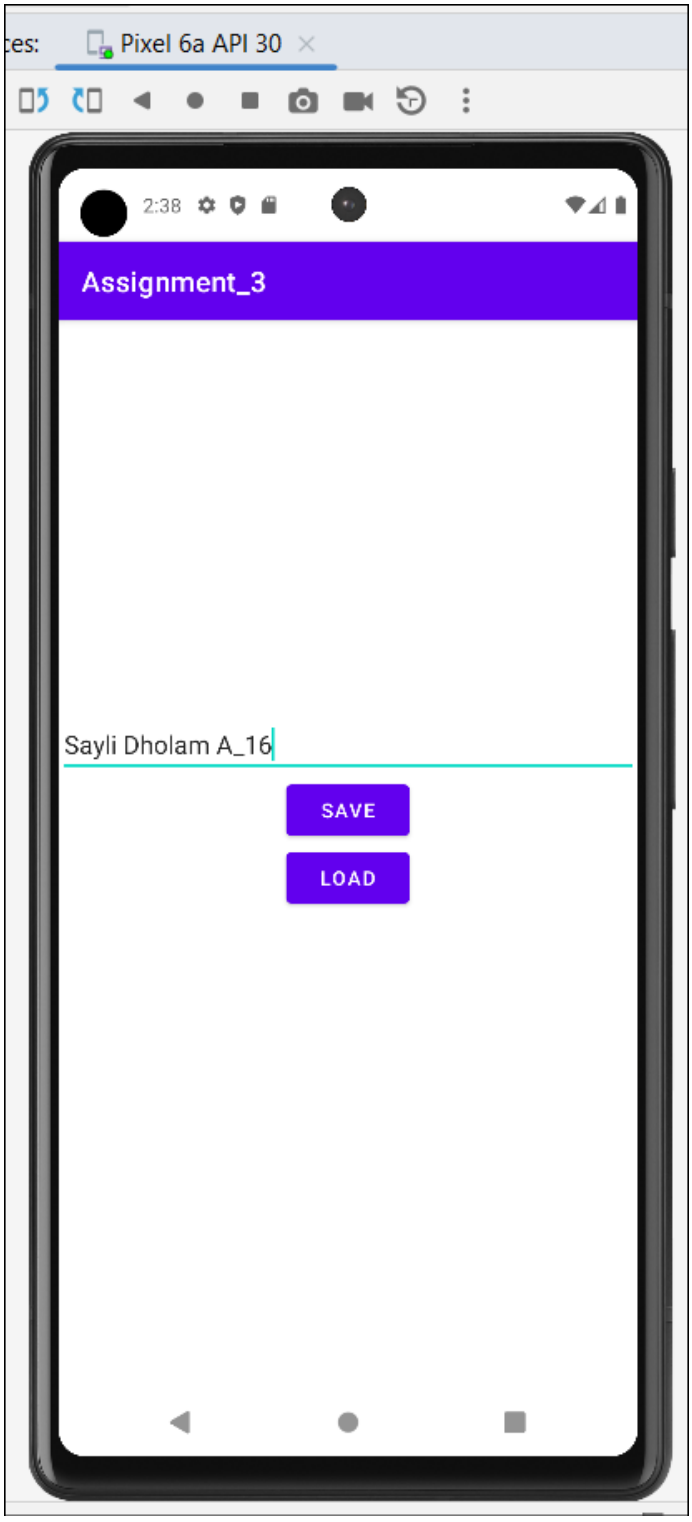
MainActivity.xml

```
<?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:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    tools:context=".MainActivity_sharedpref">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/text1"
        />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btn1"
        android:onClick="saveMethod"
        android:text="Save"
        />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/btn2"
        android:onClick="loadingMethod"
        android:text="Load"
        />

</LinearLayout>
```

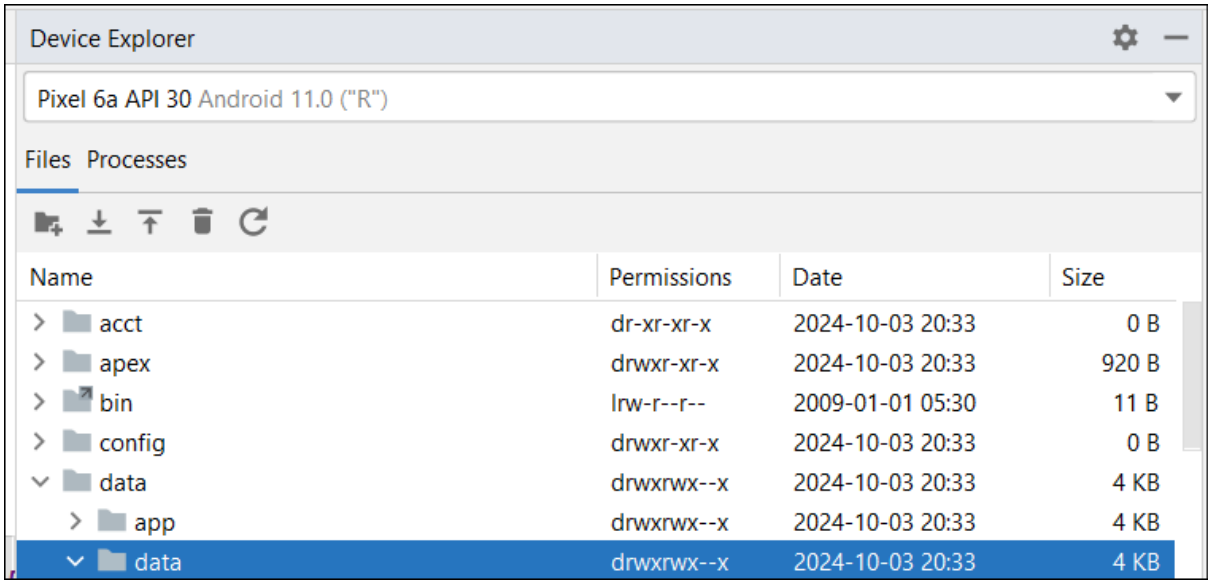
MainActivity.java

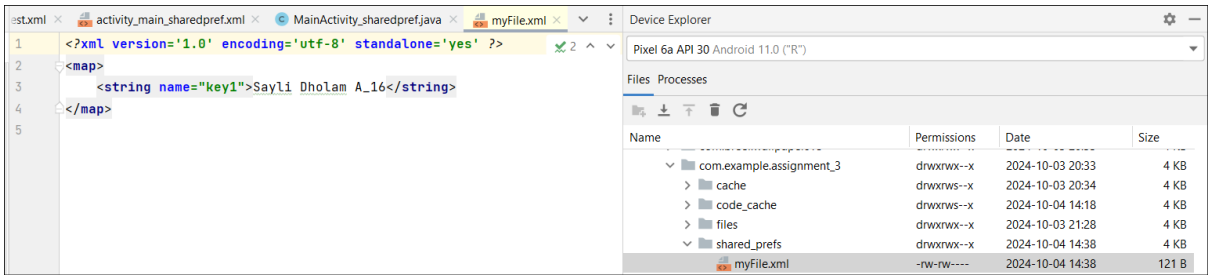
```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

    <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.Assignment_3"
        tools:targetApi="31">

        <activity
            android:name=".MainActivity_sharedpref"
            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:



s


Device Explorer
Pixel 6a API 30 Android 11.0 ("R")
Files Processes
Name
Permissions
Date
Size
acct
dr-xr-xr-x
2024-10-03 20:33
0 B
apex
drwxr-xr-x
2024-10-03 20:33
920 B
bin
lrw-r--r--
2009-01-01 05:30
11 B
config
drwxr-xr-x
2024-10-03 20:33
0 B
data
drwxrwx--x
2024-10-03 20:33
4 KB
app
drwxrwx--x
2024-10-03 20:33
4 KB
data
drwxrwx--x
2024-10-03 20:33
4 KB


activity_main_sharedpref.xml
MainActivity_sharedpref.java
myFile.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
<string name="key1">Sayli Dholam A_16</string>
</map>
Device Explorer
Pixel 6a API 30 Android 11.0 ("R")
Files Processes
Name
Permissions
Date
Size
com.example.assignment_3
drwxrwx--x
2024-10-03 20:33
4 KB
cache
drwxrws--x
2024-10-03 20:34
4 KB
code_cache
drwxrws--x
2024-10-04 14:18
4 KB
files
drwxrwx--x
2024-10-03 21:28
4 KB
shared_prefs
drwxrwx--x
2024-10-04 14:38
4 KB
myFile.xml
-rw-rw----
2024-10-04 14:38
121 B

Q4. Write a program to create a login screen and save user credentials in shared preference and display user name in second activity.

Aim: To create an Android application that demonstrates how to save user credentials (username and password) in shared preferences during login and display the username in a second activity.

Objective: The objective is to understand how to create a login screen, store user credentials securely using shared preferences, and pass data between activities.

Theory: Shared Preferences in Android is a lightweight storage mechanism used to store key-value pairs. It is commonly used to store user preferences, settings, or small amounts of persistent data like user login credentials. Shared Preferences stores data in the form of strings, booleans, integers, etc., which remain available even after the app is closed, until explicitly cleared.

In this program:

1. login screen where the user enters their credentials.
2. Save the username in shared preferences after a successful login.
3. Use an intent to switch to another activity, where the username will be displayed.

SharedPreferences: Used to save the username and password locally. This data will persist even when the app is closed.

SharedPreferences.Editor: Responsible for writing key-value pairs into SharedPreferences.

Intent: Used to move from MainActivity (Login Screen) to SecondActivity, passing data through shared preferences.

apply(): Used to save the changes made to SharedPreferences asynchronously.

TextView and EditText: Widgets used to display and input user data, respectively.

Code:

MainActivity1.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:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:layout_gravity="center"
    android:padding="20dp"
    tools:context=".MainActivity_sharedpref_user1">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/username"
        android:textAlignment="center"
        android:inputType="textEmailAddress"
        android:hint="username"
        />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textAlignment="center"
        android:hint="Password"
        android:inputType="textPassword"
        android:id="@+id/pass"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sign in"
        android:id="@+id/sub"
        android:onClick="submitbtn"
        />

</LinearLayout>
```

MainActivity1.java

```
package com.example.assignment_3;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;

public class MainActivity_sharedpref_user1 extends
AppCompatActivity {
    EditText username,password;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);

setContentView(R.layout.activity_main_sharedpref_user1);
        username=findViewById(R.id.username);
        password=findViewById(R.id.pass);
    }

    public void submitbtn(View view) {
        String msg=username.getText().toString();
        String msg1=password.getText().toString();
        SharedPreferences
sv=getSharedPreferences("LoginData",MODE_PRIVATE);
        SharedPreferences.Editor e=sv.edit();
        e.putString("key1",msg);
        e.putString("key2",msg1);
        e.commit();
        username.setText("");
        password.setText("");
        Intent i=new Intent(this,
MainActivity_sharedpref_user2.class);
        startActivity(i);
    }
}
```

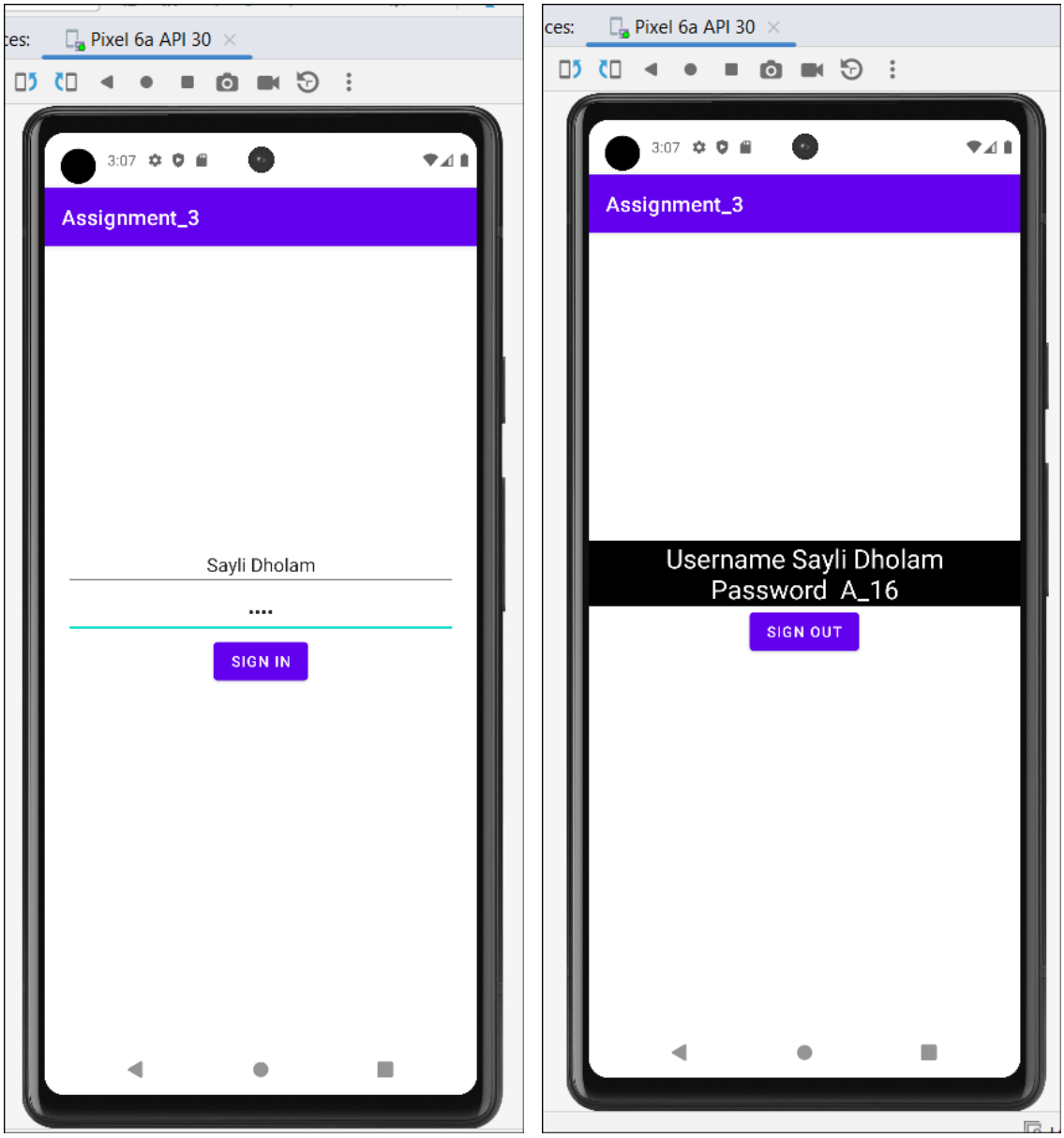
MainActivity2.xml

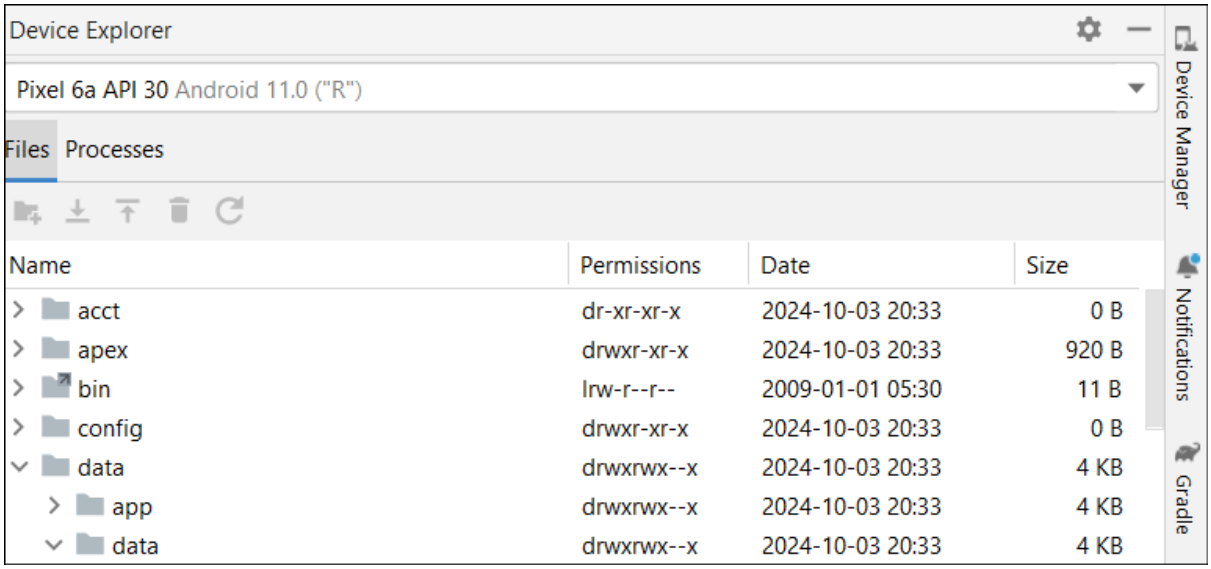
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_gravity="center"
    android:gravity="center"
    tools:context=".MainActivity_sharedpref_user2">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textAlignment="center"
        android:textSize="25dp"
        android:background="@color/black"
        android:textColor="@color/white"
        android:id="@+id/showdata"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="signOut"
        android:text="Sign Out"
        android:id="@+id/signout"/>

</LinearLayout>
```

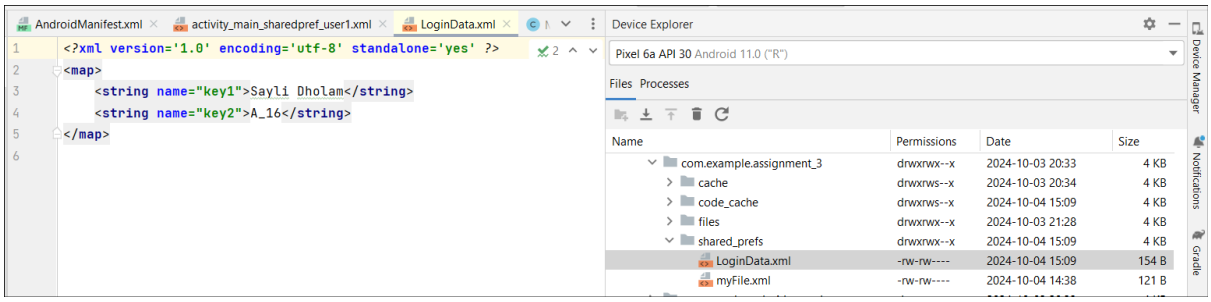
MainActivity2.java

```
package com.example.assignment_3;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;

import android.annotation.SuppressLint;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

public class MainActivity_sharedpref_user2 extends
AppCompatActivity {
    TextView tv;
    @SuppressLint("MissingInflatedId")

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);

setContentView(R.layout.activity_main_sharedpref_user2);
        tv=findViewById(R.id.showdata);
        SharedPreferences
sp=getSharedPreferences("LoginData",MODE_PRIVATE);
        String msg1=sp.getString("key1","Key Doesn't match");
        String msg2=sp.getString("key2","Key Doesn't match");
        tv.setText("Username "+msg1+"\nPassword  "+msg2);
    }

    public void signOut(View view) {
        SharedPreferences
s=getSharedPreferences("LoginData",MODE_PRIVATE);
        SharedPreferences.Editor e=s.edit();
        e.clear();
        e.commit();

        Intent intent=new Intent(this,
MainActivity_sharedpref_user1.class);
        startActivity(intent);
    }
}
```

Output:


Pixel 6a API 30
3:07
Assignment_3
Sayli Dholam
....
SIGN IN
Pixel 6a API 30
3:07
Assignment_3
Username Sayli Dholam
Password A_16
SIGN OUT


Device Explorer
Pixel 6a API 30 Android 11.0 ("R")
Files
Processes
Name
Permissions
Date
Size
acct
dr-xr-xr-x
2024-10-03 20:33
0 B
apex
drwxr-xr-x
2024-10-03 20:33
920 B
bin
lrw-r--r--
2009-01-01 05:30
11 B
config
drwxr-xr-x
2024-10-03 20:33
0 B
data
drwxrwx--x
2024-10-03 20:33
4 KB
app
drwxrwx--x
2024-10-03 20:33
4 KB
data
drwxrwx--x
2024-10-03 20:33
4 KB
Device Manager
Notifications
Gradle


AndroidManifest.xml
activity_main_sharedpref_user1.xml
LoginData.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
<string name="key1">Sayli Dholam</string>
<string name="key2">A_16</string>
</map>
Device Explorer
Pixel 6a API 30 Android 11.0 ("R")
Files
Processes
Name
Permissions
Date
Size
com.example.assignment_3
drwxrwx--x
2024-10-03 20:33
4 KB
cache
drwxrws--x
2024-10-03 20:34
4 KB
code_cache
drwxrws--x
2024-10-04 15:09
4 KB
files
drwxrwx--x
2024-10-03 21:28
4 KB
shared_prefs
drwxrwx--x
2024-10-04 15:09
4 KB
LoginData.xml
-rw-rw----
2024-10-04 15:09
154 B
myFile.xml
-rw-rw----
2024-10-04 14:38
121 B

Q5. Write a program to turn on and off Bluetooth service.

Aim: To create an Android application that allows users to turn on and off the Bluetooth service

Objective: To understand how to work with the BluetoothAdapter class to control Bluetooth functionality and manage necessary permissions for Bluetooth operations in an Android application.

Theory: In Android, the Bluetooth service is managed by the BluetoothAdapter class, which represents the device's Bluetooth radio. This class provides methods to perform several Bluetooth-related operations, such as enabling or disabling Bluetooth, checking whether Bluetooth is supported on the device.

BluetoothAdapter: The central class that manages the Bluetooth functionality. You use this class to enable, disable, and check the status of Bluetooth on the device.

Permissions: Android requires specific permissions to access Bluetooth functionalities.

The permissions required are:

- BLUETOOTH: Allows the app to connect to Bluetooth devices.
- BLUETOOTH_ADMIN: Allows the app to discover and pair with Bluetooth devices.
- BLUETOOTH_CONNECT: Required for enabling and disabling Bluetooth starting from Android 12.

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.BLUETOOTH" />
    <uses-permission
android:name="android.permission.BLUETOOTH_ADMIN" />
    <uses-permission
android:name="android.permission.BLUETOOTH_CONNECT" />

    <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.Assignment_3"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity_bluetooth"
            android:exported="true">
            <intent-filter>
                <action
android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".MainActivity_bluetooth"
            android:exported="true">
            <intent-filter>
                <action
android:name="android.intent.action.MAIN" />

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

```
    </application>

</manifest>
```

MainActivity.xml

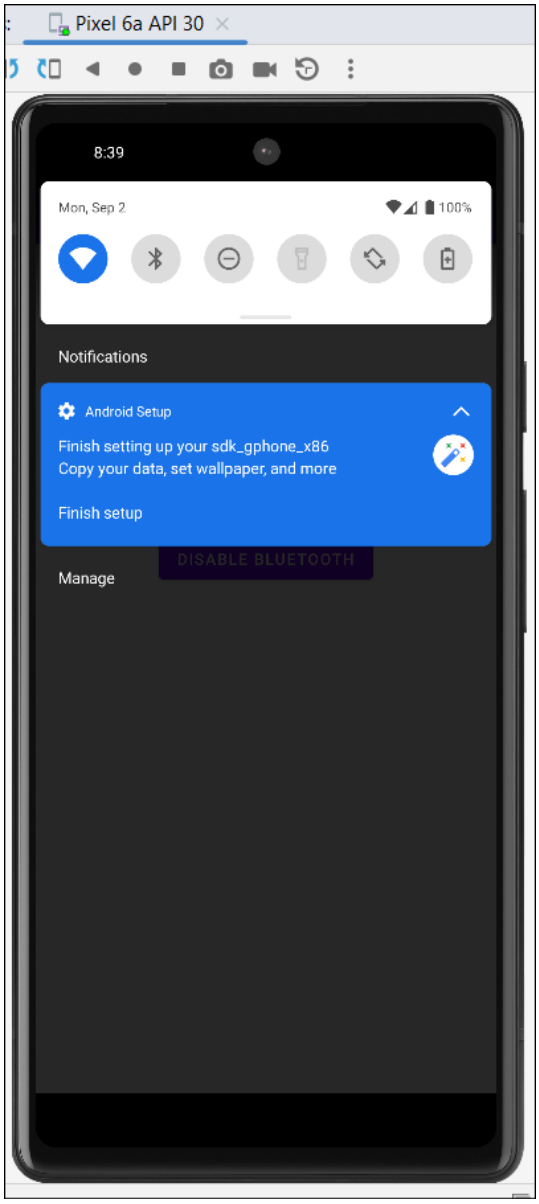
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp"
    tools:context=".MainActivity_bluetooth">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="Sayli Dholam A_16" />


    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="90dp"
        android:text="Enable Bluetooth"
        android:onClick="Enable"/>


    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="90dp"
        android:text="Disable Bluetooth"
        android:onClick="Disable"/>


</LinearLayout>
```

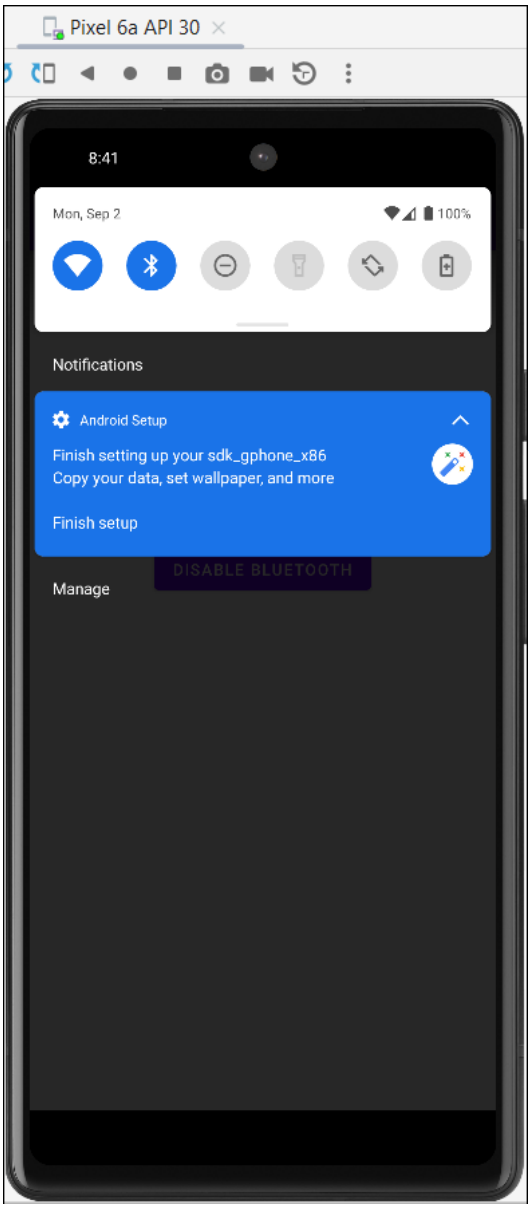
MainActivity.java

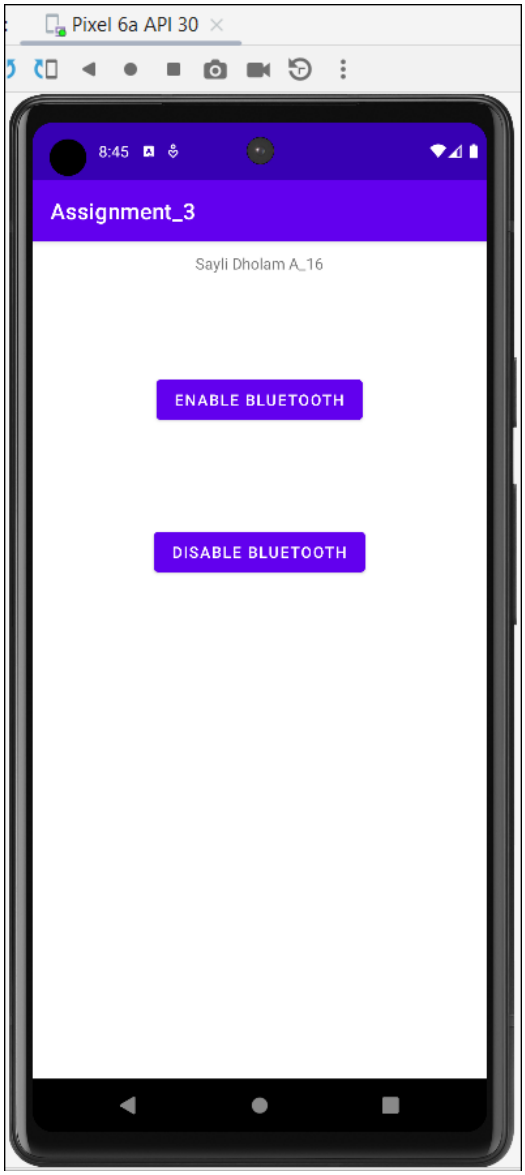
```
package com.example.assignment_3;

import androidx.appcompat.app.AppCompatActivity;

import android.bluetooth.BluetoothAdapter;
import android.os.Bundle;
import androidx.core.app.ActivityCompat;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.view.View;
public class MainActivity_bluetooth extends AppCompatActivity
{

    BluetoothAdapter bAdapter = null;

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

        bAdapter = BluetoothAdapter.getDefaultAdapter();
    }

    public void Enable(View view)
    {
        if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.BLUETOOTH_CONNECT) !=
PackageManager.PERMISSION_GRANTED)
        {
            return;
        }
        Intent intentEnabled = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
        int REQUEST_ENABLE_BT = 2;
        startActivityForResult(intentEnabled,
REQUEST_ENABLE_BT);
    }

    public void Disable(View view) {
        if (ActivityCompat.checkSelfPermission(this,
android.Manifest.permission.BLUETOOTH_CONNECT) !=
PackageManager.PERMISSION_GRANTED) {
            return;
        }
        Intent intentDisabled = new
Intent("android.bluetooth.adapter.action.REQUEST_DISABLE");
        startActivityForResult(intentDisabled, 2);
```

```
        }
}
```

Output:


Pixel 6a API 30
8:39
Mon, Sep 2
100%
Notifications
Android Setup
Finish setting up your sdk_gphone_x86
Copy your data, set wallpaper, and more
Finish setup
DISABLE BLUETOOTH
Manage


Pixel 6a API 30
8:41
Mon, Sep 2
100%
Notifications
Android Setup
Finish setting up your sdk_gphone_x86
Copy your data, set wallpaper, and more
Finish setup
DISABLE BLUETOOTH
Manage


Pixel 6a API 30
8:45
Assignment_3
Sayli Dholam A_16
ENABLE BLUETOOTH
DISABLE BLUETOOTH

Q6. Write a program to turn on and off Wi-Fi service.

Aim: To create an Android application that turns the Wi-Fi service on and off

Objective: Learn how to manipulate the Wi-Fi service in an Android application. Implement the code to toggle Wi-Fi using Android's WifiManager.

Theory: Android provides the WifiManager class, which allows developers to manage Wi-Fi connectivity.

Permissions: To control Wi-Fi on an Android device, the following permissions must be declared in the AndroidManifest.xml file:

- android.permission.ACCESS_WIFI_STATE: Allows the app to view the current state of Wi-Fi.
- android.permission.CHANGE_WIFI_STATE: Allows the app to change the state of Wi-Fi.

WifiManager Class: The WifiManager class provides methods to manage Wi-Fi connectivity. Some of the key methods include:

- setWifiEnabled(boolean enabled): Enables or disables Wi-Fi.
- isWifiEnabled(): Checks whether Wi-Fi is enabled or not.

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.ACCESS_WIFI_STATE" />
    <uses-permission
android:name="android.permission.CHANGE_WIFI_STATE" />

    <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.Assignment_3"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity_wifi"
            android:exported="true">
            <intent-filter>
                <action
android:name="android.intent.action.MAIN" />

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

    </application>

</manifest>
```

MainActivity.xml

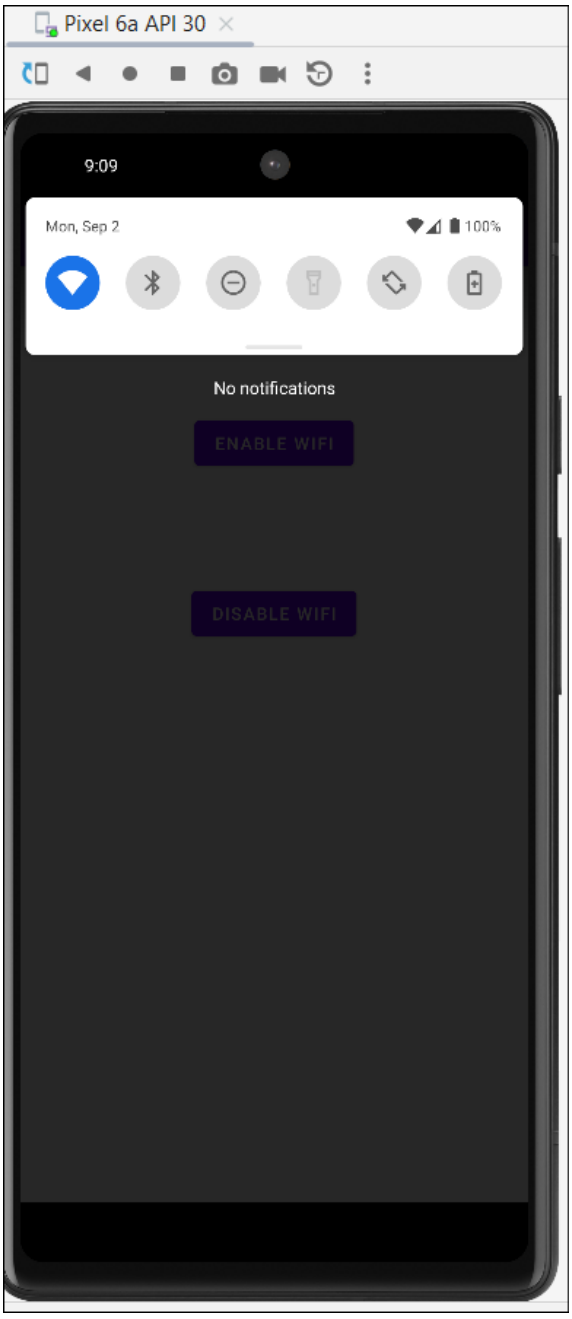
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp"
    tools:context=".MainActivity_wifi">


    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="Sayli Dholam A_16" />


    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="90dp"
        android:text="Enable Wifi"
        android:onClick="Enable"/>


    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="90dp"
        android:text="Disable Wifi"
        android:onClick="Disable"/>


</LinearLayout>
```

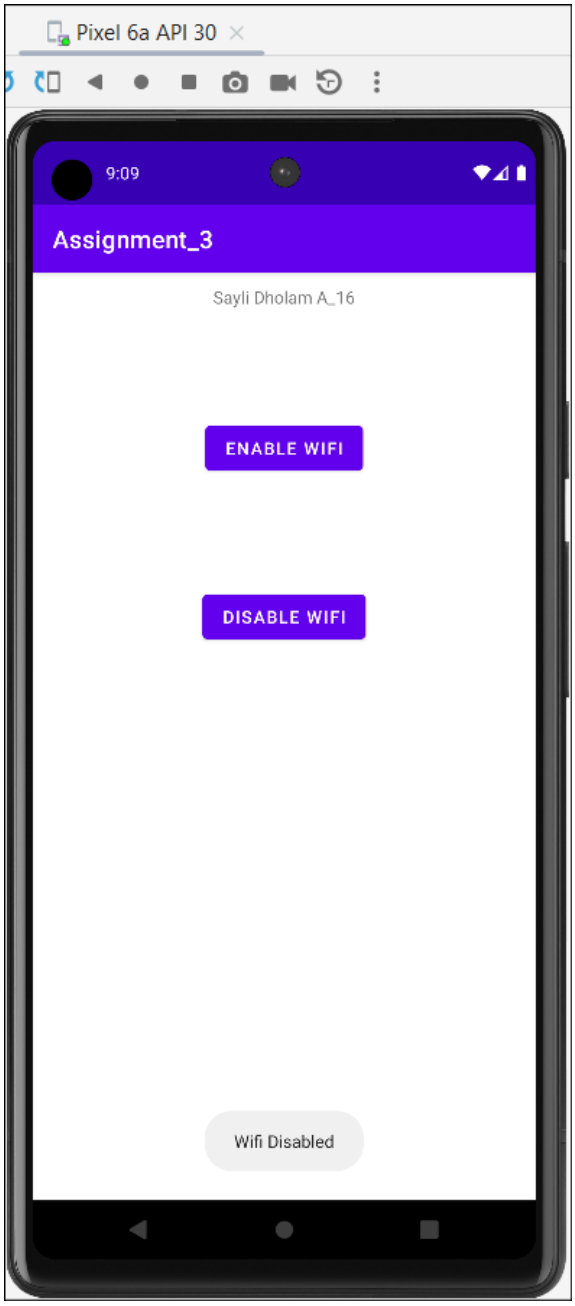
MainActivity.java

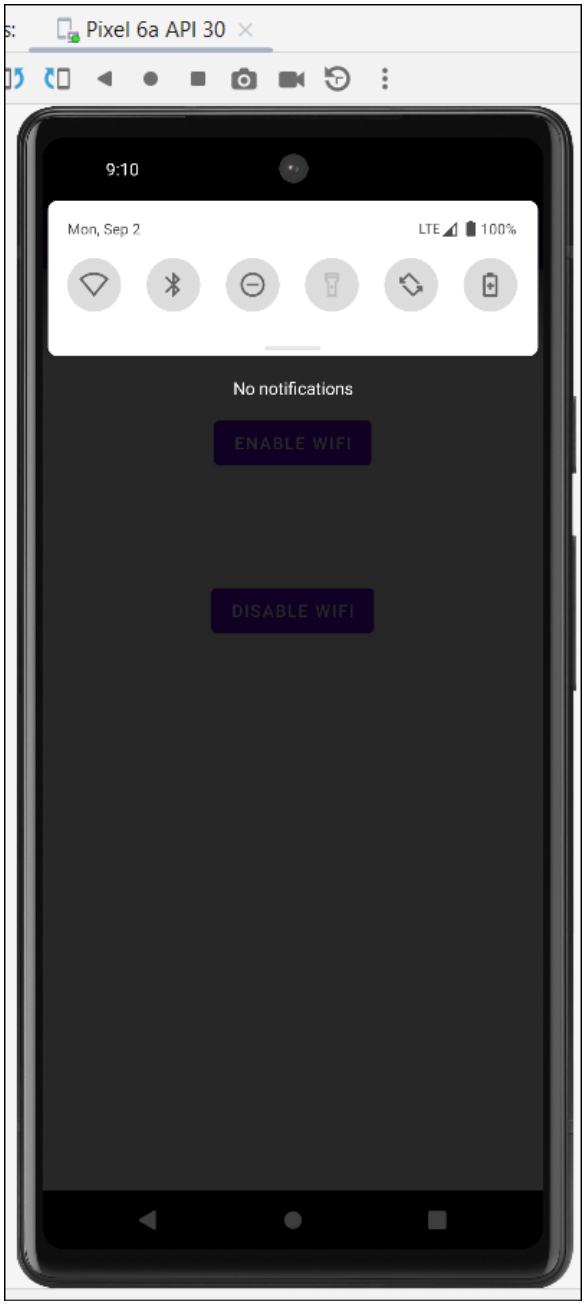
```
package com.example.assignment_3;

import androidx.appcompat.app.AppCompatActivity;

import android.net.wifi.WifiManager;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

public class MainActivity_wifi extends AppCompatActivity {

    WifiManager wifi;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_wifi);

        wifi = (WifiManager)
getApplicationContext().getSystemService(WIFI_SERVICE);
    }
    public void Enable(View view)
    {
        wifi.setWifiEnabled(true);
        Toast.makeText(this, "Wifi Enabled",
Toast.LENGTH_LONG).show();
    }
    public void Disable(View view)
    {
        wifi.setWifiEnabled(false);
        Toast.makeText(this, "Wifi Disabled",
Toast.LENGTH_LONG).show();
    }
}
```

Output:


Pixel 6a API 30
9:09
Assignment_3
Sayli Dholam A_16
ENABLE WIFI
DISABLE WIFI
Wifi Enabled


Pixel 6a API 30
9:09
Mon, Sep 2
100%
No notifications
ENABLE WIFI
DISABLE WIFI


Pixel 6a API 30
9:09
Assignment_3
Sayli Dholam A_16
ENABLE WIFI
DISABLE WIFI
Wifi Disabled


Pixel 6a API 30
9:10
Mon, Sep 2
LTE 100%
No notifications
ENABLE WIFI
DISABLE WIFI

Q7. Write a program to create user registration form after registration data will be inserted in SQLite database and also design activity which displays that information.

Aim: create a android application which accepts user data, stores it in SQLite Database and fetches the same.

Objective: To understand how to design and implement user registration forms in Android.

To learn how to store and retrieve data using SQLite in Android.

To create a user interface that allows users to input their information and view the stored data.

Theory: In Android, SQLite is a lightweight database used for storing and retrieving structured data locally within an application. Each Android application can use SQLite to manage its own private database.

Cursor res = helper.getallData():

Calls the getallData() method of the Helper class, which returns a Cursor object containing the data from the database.

helper.getallData(): Retrieves all data from the SQLite database.

Helper class : The Helper class in your code is an extension of the SQLiteOpenHelper class, which is used to manage database creation and version management. This class includes methods for creating the database, upgrading it, and performing operations like inserting and retrieving data.

SQLiteDatabase db = this.getReadableDatabase():

Opens the database for reading.

Helper(Context context):

This constructor initializes the SQLiteOpenHelper with the provided context, database name, and version. The null passed as the second argument indicates that a default CursorFactory will be used.

onCreate(SQLiteDatabase db):

This method is called when the database is created for the first time. It’s where the creation of tables and initial data insertion should happen.

SQLiteDatabase db = this.getWritableDatabase():

Opens the database for writing.

ContentValues contentvalues = new ContentValues():

A ContentValues object is used to store the data you want to insert into the table. It's essentially a key-value pair where the key is the column name, and the value is the data to be inserted.

Code:

MainActivity.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:layout_height="match_parent"
    android:padding = "10dp"
    android:orientation = "vertical"
    tools:context=".MainActivity_registration">


    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:layout_gravity="center"
        android:text="Sayli Dholam A_16" />

    <!-- Name TextView and EditText -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="50dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Name"
            android:layout_gravity="center_vertical"/>

        <EditText
            android:id="@+id/Name"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="2"
            android:hint="Enter Name"/>
    </LinearLayout>

    <!-- Surname TextView and EditText -->
    <LinearLayout
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="25dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Surname"
            android:layout_gravity="center_vertical"/>

        <EditText
            android:id="@+id/Surname"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="2"
            android:hint="Enter Surname"/>
    </LinearLayout>

    <!-- Marks TextView and EditText -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="25dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Marks"
            android:layout_gravity="center_vertical"/>

        <EditText
            android:id="@+id/Marks"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="2"
            android:hint="Enter marks"/>
    </LinearLayout>

    <!-- Buttons Row -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center"
        android:layout_marginTop="35dp">

        <Button
            android:id="@+id/AddData"
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="ADD DATA"
            android:onClick="add_data"
            android:layout_marginEnd="8dp" />

        <Button
            android:id="@+id/ViewAll"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:onClick="view_data"
            android:text="VIEW ALL" />
    </LinearLayout>

    <!-- Data TextView -->
    <TextView
        android:id="@+id/Data"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Data"
        android:layout_marginTop="50dp"
        android:padding="8dp"
        android:background="@color/purple_200" />

</LinearLayout>
```

MainActivity.java

```
package com.example.assignment_3;

import androidx.appcompat.app.AppCompatActivity;

import android.database.Cursor;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity_registration extends
AppCompatActivity {

    EditText name, surname, marks;
    TextView data;
    Helper helper;

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

        name = findViewById(R.id.Name);
        surname = findViewById(R.id.Surname);
        marks = findViewById(R.id.Marks);
        data = findViewById(R.id.Data);
        helper = new Helper(this );
    }

    public void add_data(View view)
    {
        boolean isInserted =
helper.insert_data(name.getText().toString(),
                surname.getText().toString(),
                Integer.parseInt(marks.getText().toString()));

        if(isInserted == true)
        {
            Toast.makeText(this, "Data inserted successfully",
Toast.LENGTH_LONG).show();
        }
        else
        {
            Toast.makeText(this, "Data not inserted ",
Toast.LENGTH_LONG).show();
        }
    }
```

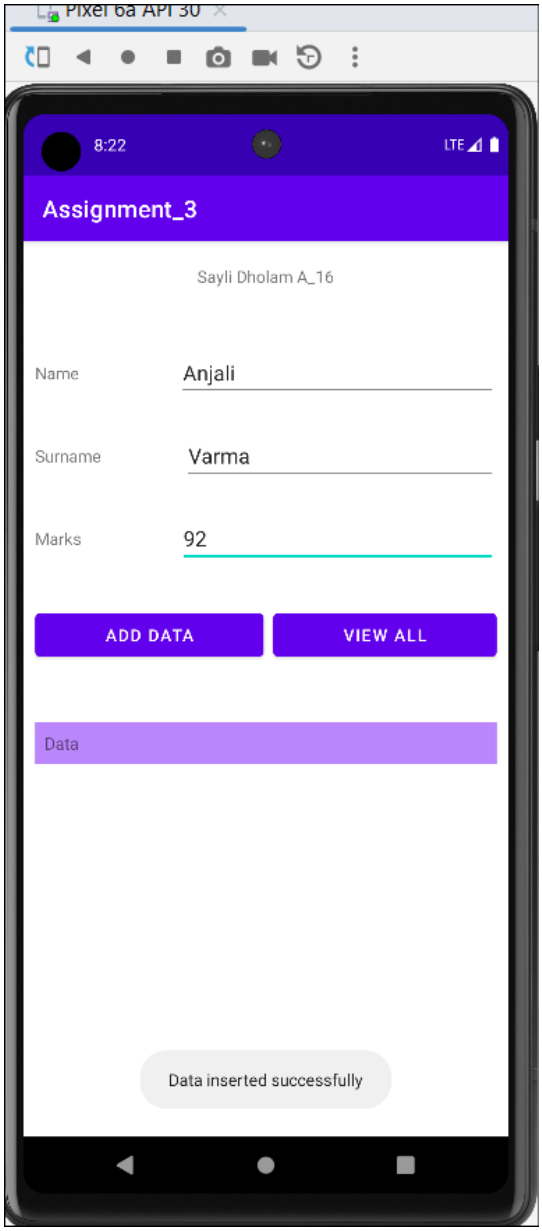
```
    public void view_data(View view)
    {
        Cursor res = helper.getallData();
        if(res.getCount() == 0)
        {
            Toast.makeText(this, "No data found",
Toast.LENGTH_LONG).show();
            return;
        }

        StringBuffer buffer = new StringBuffer();
        while(res.moveToNext())
        {
            buffer.append("Id : " + res.getString(0) + "\n");
            buffer.append("Name : " + res.getString(1) +
"\n");
            buffer.append("Surname : " + res.getString(2) +
"\n");
            buffer.append("Marks : " + res.getString(3) +
"\n");
        }
        data.setText(buffer.toString());
    }
}
```

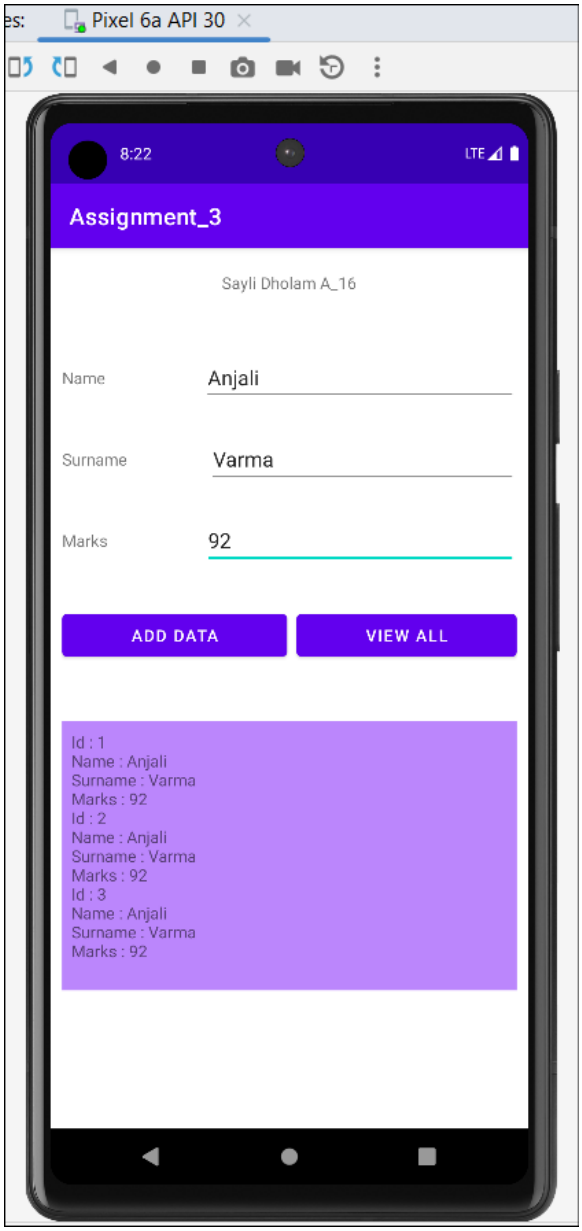
Helper.java

```
package com.example.assignment_3;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import androidx.annotation.Nullable;

public class Helper extends SQLiteOpenHelper {

    public static final String DATABASE_NAME = "mystudent.db";
    public static final String TABLE_NAME = "student_table";

    public static final String col_1 = "ID";
    public static final String col_2 = "NAME";
    public static final String col_3 = "SURNAME";
    public static final String col_4 = "MARKS";

    public Helper(@Nullable Context context) {
        super(context, DATABASE_NAME, null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(" create table " + TABLE_NAME + "(ID
INTEGER PRIMARY KEY AUTOINCREMENT, NAME TEXT, SURNAME TEXT,
MARKS INTEGER)");
    }

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

    public boolean insert_data(String Name, String Surname,
int Marks)
    {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentvalues = new ContentValues();
        contentvalues.put(col_2,Name);
        contentvalues.put(col_3,Surname);
        contentvalues.put(col_4,Marks);

        long result = db.insert(TABLE_NAME, null,
contentvalues);
        if(result == -1)
        {
```

```
            return false;
        }
        else
        {
            return true;
        }
    }

    public Cursor getallData()
    {
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor res = db.rawQuery(" select * from " +
TABLE_NAME, null);
        return res;
    }
}
```

Output:


Pixel 6a API 30
8:22
LTE
Assignment_3
Sayli Dholam A_16
Name
Anjali
Surname
Varma
Marks
92
ADD DATA
VIEW ALL
Data
Data inserted successfully


Pixel 6a API 30
8:22
LTE
Assignment_3
Sayli Dholam A_16
Name
Anjali
Surname
Varma
Marks
92
ADD DATA
VIEW ALL
Id : 1
Name : Anjali
Surname : Varma
Marks : 92
Id : 2
Name : Anjali
Surname : Varma
Marks : 92
Id : 3
Name : Anjali
Surname : Varma
Marks : 92

Q8. Write a program to perform all CRUD operations using SQLite database.

Aim: To develop an Android application that performs all CRUD (Create, Read, Update, Delete) operations using an SQLite database.

Objective: To learn how to interact with an SQLite database in Android by performing basic CRUD operations:

1. Create - Insert data into the database.
2. Read - Retrieve data from the database.
3. Update - Modify existing data in the database.
4. Delete - Remove data from the database.

Theory: SQLite is a lightweight, self-contained, serverless database engine. It is a popular choice for storing structured data in mobile applications. Android provides native support for SQLite databases. The SQLiteOpenHelper class is used to manage database creation and version management. In this program, we will use an SQLite database to store user details and perform various CRUD operations.

SQLiteOpenHelper class - This class helps create the database, upgrade it, and manage its version.

SQLiteDatabase class - This class provides methods to perform CRUD operations on the database.

Cursor res = helper.getallData():

Calls the getallData() method of the Helper class, which returns a Cursor object containing the data from the database.

helper.getallData(): Retrieves all data from the SQLite database.

Helper class : The Helper class in your code is an extension of the SQLiteOpenHelper class, which is used to manage database creation and version management. This class includes methods for creating the database, upgrading it, and performing operations like inserting and retrieving data.

SQLiteDatabase db = this.getReadableDatabase():

Opens the database for reading.

Helper(Context context):

This constructor initializes the SQLiteOpenHelper with the provided context, database name, and version. The null passed as the second argument indicates that a default CursorFactory will be used.

onCreate(SQLiteDatabase db):

This method is called when the database is created for the first time. It’s where the creation of tables and initial data insertion should happen.

SQLiteDatabase db = this.getWritableDatabase():

Opens the database for writing.

ContentValues contentvalues = new ContentValues():

A ContentValues object is used to store the data you want to insert into the table. It's essentially a key-value pair where the key is the column name, and the value is the data to be inserted.

res.moveToNext() does:

res is the Cursor object that holds the results of a query from the SQLite database.The moveToNext() method advances the cursor to the next row in the result set. It returns a boolean value:

true if the cursor is moved to the next row.false if there are no more rows

Code:

MainActivity.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:layout_height="match_parent"
    android:padding = "10dp"
    android:orientation = "vertical"
    tools:context=".MainActivity_crud">

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:layout_gravity="center"
        android:text="Sayli Dholam A_16" />

    <!-- ID TextView and EditText -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="40dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="ID"
            android:layout_gravity="center_vertical"/>


        <EditText
            android:id="@+id/Id"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="2"
            android:hint="Enter Id"/>
    </LinearLayout>

    <!-- Name TextView and EditText -->
    <LinearLayout
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="30dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Name"
            android:layout_gravity="center_vertical"/>

        <EditText
            android:id="@+id/Name"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="2"
            android:hint="Enter Name"/>
    </LinearLayout>

    <!-- Salary TextView and EditText -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="30dp">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Salary"
            android:layout_gravity="center_vertical"/>

        <EditText
            android:id="@+id/Salary"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="2"
            android:hint="Enter Salary"/>
    </LinearLayout>

    <!-- Buttons Row 1 -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center"
        android:layout_marginTop="20dp">

        <Button
            android:id="@+id/add"
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="ADD DATA"
            android:onClick="add_data"
            android:layout_marginEnd="8dp" />

        <Button
            android:id="@+id/view"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="VIEW DATA"
            android:onClick="view_data"/>
    </LinearLayout>

    <!-- Buttons Row 2 -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center"
        android:layout_marginTop="10dp">

        <Button
            android:id="@+id/update"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="UPDATE DATA"
            android:onClick="update_data"
            android:layout_marginEnd="8dp" />

        <Button
            android:id="@+id/delete"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="DELETE DATA"
            android:onClick="delete_data"/>
    </LinearLayout>

    <!-- Buttons Row 3 -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center"
        android:layout_marginTop="10dp">

        <Button
```

```
            android:id="@+id/search"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="SEARCH DATA"
            android:onClick="search_data"
            android:layout_marginEnd="8dp" />

        <Button
            android:id="@+id/clear"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:onClick="clear_data"
            android:text="CLEAR DATA" />
    </LinearLayout>

    <!-- Data TextView -->
    <TextView
        android:id="@+id/DataTextView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Data"
        android:layout_marginTop="50dp"
        android:padding="8dp"
        android:background="@color/purple_200" />

</LinearLayout>
```

MainActivity.java

```
package com.example.assignment_3;

import androidx.appcompat.app.AppCompatActivity;
import android.database.Cursor;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity_crud extends AppCompatActivity {

    Helper_2 mDb;
    EditText editID, editName, editSalary;
    TextView tv;

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

        editID = findViewById(R.id.Id);
        editName = findViewById(R.id.Name);
        editSalary = findViewById(R.id.Salary);
        tv = findViewById(R.id.DataTextView);
        mDb = new Helper_2(this );
    }

    public void add_data(View view)
    {
        boolean isInserted =
mDb.insert_data(editName.getText().toString(),
Integer.parseInt(editSalary.getText().toString()));


        if(isInserted == true)
        {
            Toast.makeText(this, "Data inserted successfully",
Toast.LENGTH_LONG).show();
        }
        else
        {
            Toast.makeText(this, "Data not inserted ",
Toast.LENGTH_LONG).show();
        }
    }
```

```
    public void view_data(View view)
    {
        Cursor res = mDb.getallData();
        if(res.getCount() == 0)
        {
            Toast.makeText(this, "No data found",
Toast.LENGTH_LONG).show();
            return;
        }

        StringBuffer buffer = new StringBuffer();
        while(res.moveToNext())
        {
            buffer.append("Id : " + res.getString(0) + "\n");
            buffer.append("Name : " + res.getString(1) +
"\n");
            buffer.append("Salary : " + res.getString(2) +
"\n");
        }
        tv.setText(buffer.toString());
    }

    public void update_data(View view)
    {
        boolean isUpdated =
mDb.update_data(editID.getText().toString(),
editName.getText().toString(),
editSalary.getText().toString());


        if(isUpdated == true)
        {
            Toast.makeText(this, "Data updated successfully",
Toast.LENGTH_LONG).show();
        }
        else
        {
            Toast.makeText(this, "Data not updated ",
Toast.LENGTH_LONG).show();
        }
    }

    public void delete_data(View view)
    {
        Integer deleted_rows =
mDb.delete_data(editID.getText().toString());
        if(deleted_rows > 0)
        {
            Toast.makeText(this, "Data deleted successfully",
Toast.LENGTH_LONG).show();
```

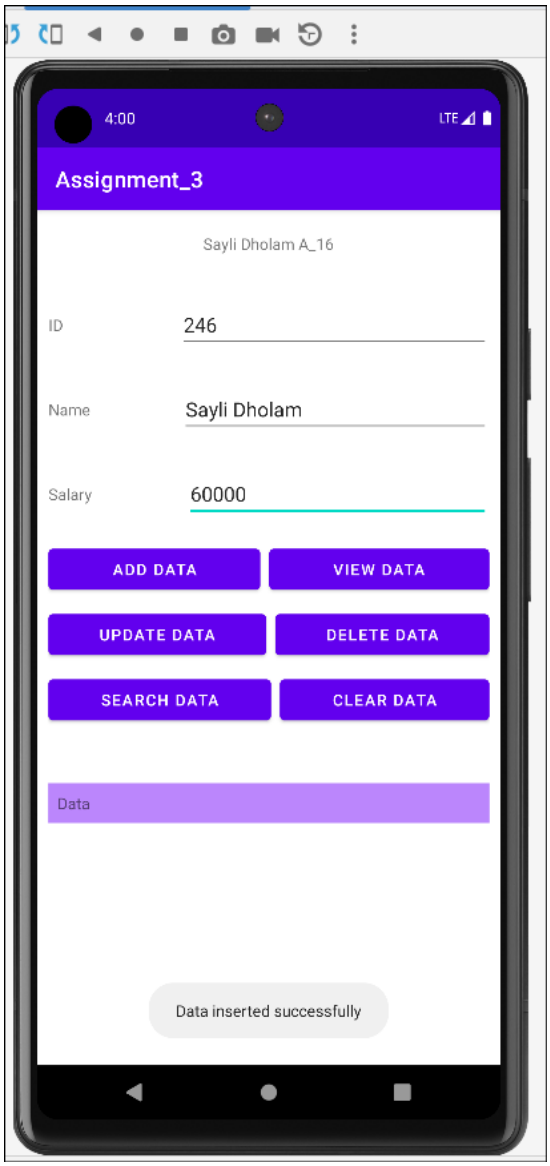
```
        }
        else
        {
            Toast.makeText(this, "Data not deleted ",
Toast.LENGTH_LONG).show();
        }
    }

    public void search_data(View view)
    {
        Cursor res =
mDb.search_data(editID.getText().toString());
        if(res.getCount() == 0)
        {
            Toast.makeText(this, "No data found",
Toast.LENGTH_LONG).show();
        }
        else
        {
            while(res.moveToNext())
            {
                editID.setText(res.getString(0));
                editName.setText(res.getString(1));
                editSalary.setText(res.getString(2));
            }
        }
    }

    public void clear_data(View view)
    {
        editID.setText("");
        editName.setText("");
        editSalary.setText("");
    }
}
```

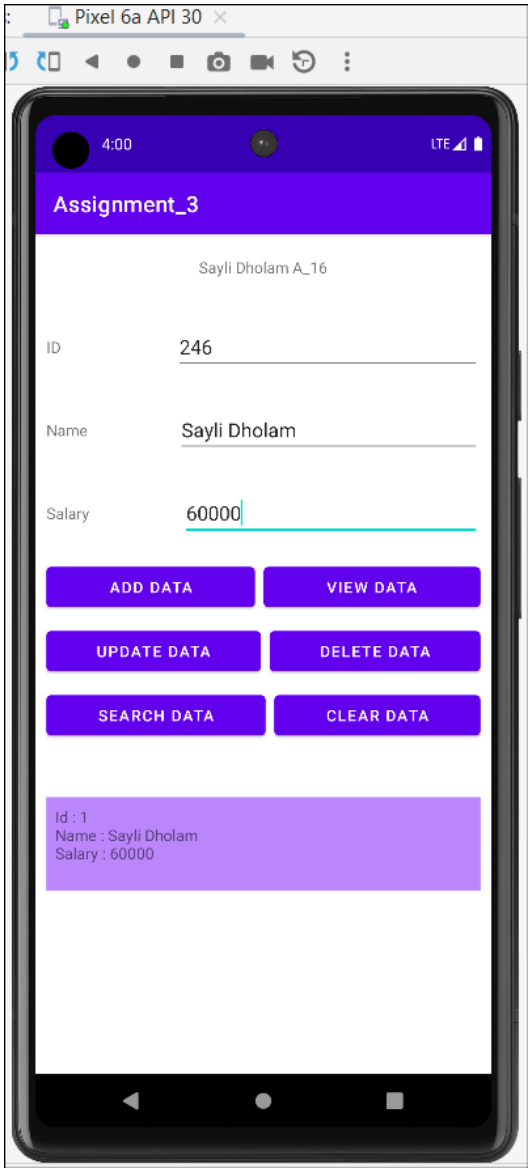
Helper.java

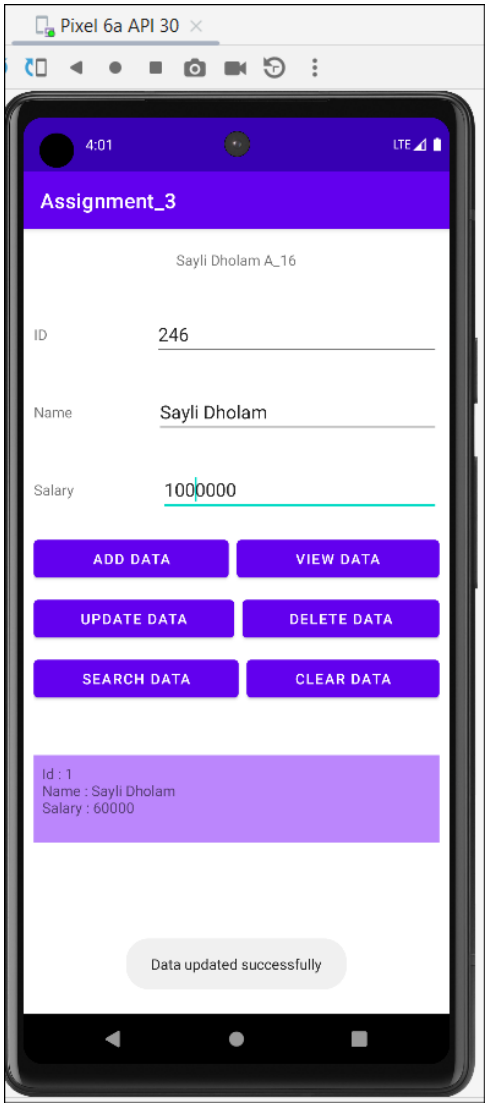
```
package com.example.assignment_3;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import androidx.annotation.Nullable;

public class Helper_2 extends SQLiteOpenHelper {
    public static final String DATABASE_NAME = "employee.db";
    public static final String TABLE_NAME = "employee_table";
    public static final String col_1 = "ID";
    public static final String col_2 = "NAME";
    public static final String col_3 = "SALARY";

    public Helper_2(@Nullable Context context) {
        super(context,DATABASE_NAME, null, 1);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(" create table " + TABLE_NAME + "(ID
INTEGER PRIMARY KEY AUTOINCREMENT, NAME TEXT, SALARY
INTEGER)");
    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int
i, int i1) {
    }

    public boolean insert_data(String name, int salary)
    {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentvalues = new ContentValues();
        contentvalues.put(col_2,name);
        contentvalues.put(col_3,salary);


        long result = db.insert(TABLE_NAME, null,
contentvalues);
        if(result == -1)
        {
            return false;
        }
        else
```

```
        {
            return true;
        }
    }

    public Cursor getallData()
    {
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor res = db.rawQuery(" select * from " +
TABLE_NAME, null);
        return res;
    }

    public Cursor search_data(String name)
    {
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor res = db.rawQuery(" select * from " +
TABLE_NAME + "where" +  col_2 + " like '%" + name + "%'",
null); ;
        return res;
    }

    public boolean update_data(String id, String name, String
salary)
    {
        SQLiteDatabase db = this.getReadableDatabase();
        ContentValues contentvalues = new ContentValues();
        contentvalues.put(col_1,id);
        contentvalues.put(col_2,name);
        contentvalues.put(col_3,salary);
        db.update(TABLE_NAME, contentvalues, "ID = ?", new
String[]{id});
        return true;
    }

    public Integer delete_data(String id)
    {
        SQLiteDatabase db = this.getReadableDatabase();
        ContentValues contentvalues = new ContentValues();
        Integer res = db.delete(TABLE_NAME, "ID = ?", new
String[]{id});
        return res;
    }
}
```

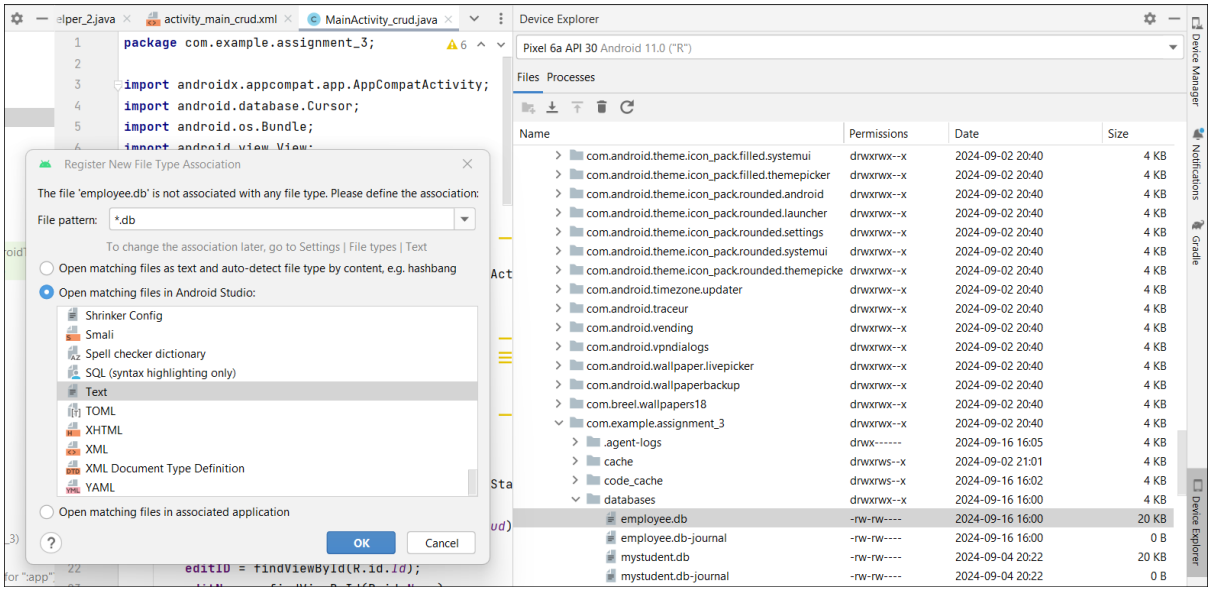
Output:

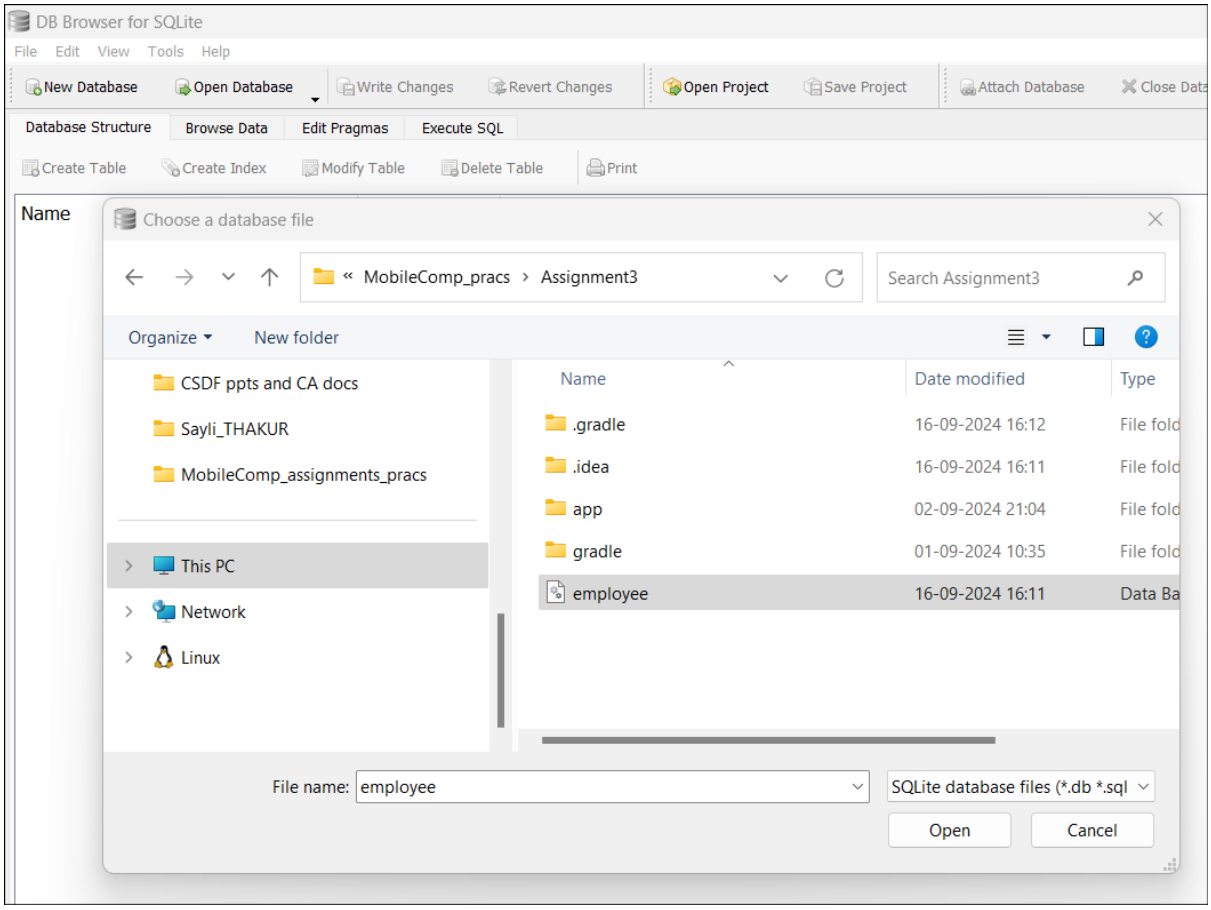

4:00
LTE
Assignment_3
Sayli Dholam A_16
ID
246
Name
Sayli Dholam
Salary
60000
ADD DATA
VIEW DATA
UPDATE DATA
DELETE DATA
SEARCH DATA
CLEAR DATA
Data
Data inserted successfully

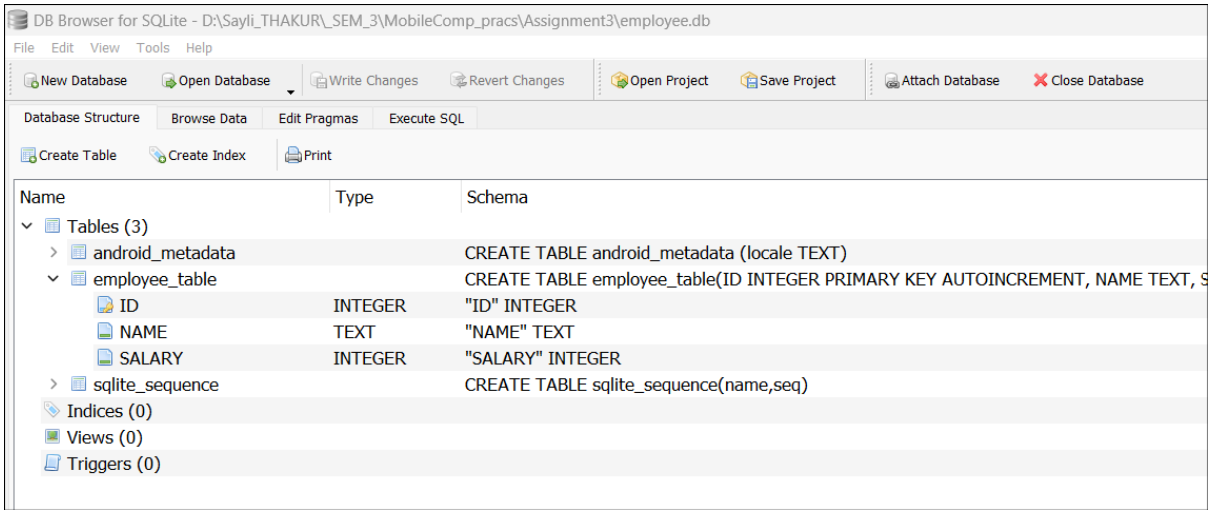

Pixel 6a API 30
4:00
LTE
Assignment_3
Sayli Dholam A_16
ID
246
Name
Sayli Dholam
Salary
60000
ADD DATA
VIEW DATA
UPDATE DATA
DELETE DATA
SEARCH DATA
CLEAR DATA
Id : 1
Name : Sayli Dholam
Salary : 60000

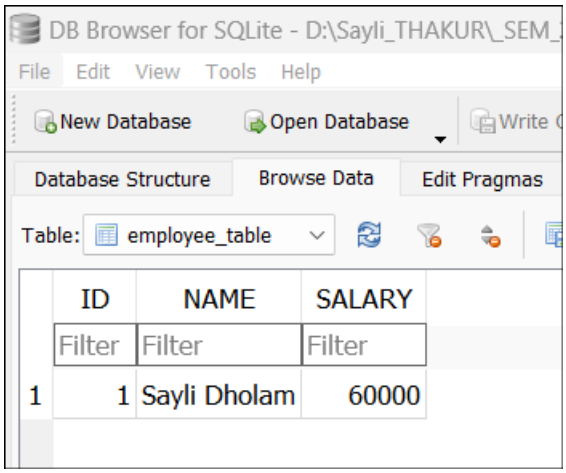

Pixel 6a API 30
4:01
LTE
Assignment_3
Sayli Dholam A_16
ID
246
Name
Sayli Dholam
Salary
1000000
ADD DATA
VIEW DATA
UPDATE DATA
DELETE DATA
SEARCH DATA
CLEAR DATA
Id : 1
Name : Sayli Dholam
Salary : 60000
Data updated successfully


data
drwxrwx--x
2024-09-
app
drwxrwx--x
2024-09-
data
drwxrwx--x
2024-09-
android
drwxrwx--x
2024-09-


MainActivity_crud.java
Device Explorer
Pixel 6a API 30 Android 11.0 ("R")
Register New File Type Association
The file 'employee.db' is not associated with any file type. Please define the association:
File pattern: *.db
Open matching files in Android Studio:
Shrinker Config
Smali
Spell checker dictionary
SQL (syntax highlighting only)
Text
TOML
XHTML
XML
XML Document Type Definition
YAML
OK
Cancel
employee.db
employee.db-journal
mystudent.db
mystudent.db-journal


DB Browser for SQLite
New Database
Open Database
Write Changes
Revert Changes
Open Project
Save Project
Database Structure
Browse Data
Edit Pragmas
Execute SQL
Create Table
Create Index
Modify Table
Delete Table
Print
Choose a database file
MobileComp_pracs › Assignment3
Search Assignment3
Organize
New folder
CSDF ppts and CA docs
Sayli_THAKUR
MobileComp_assignments_pracs
This PC
Network
Linux
Name
Date modified
Type
.gradle
.idea
app
gradle
employee
File name: employee
SQLite database files (*.db *.sql
Open
Cancel


DB Browser for SQLite - D:\Sayli_THAKUR_SEM_3\MobileComp_pracs\Assignment3\employee.db
File Edit View Tools Help
New Database
Open Database
Write Changes
Revert Changes
Open Project
Save Project
Attach Database
Close Database
Database Structure
Browse Data
Edit Pragmas
Execute SQL
Create Table
Create Index
Print
Name
Type
Schema
Tables (3)
android_metadata
CREATE TABLE android_metadata (locale TEXT)
employee_table
CREATE TABLE employee_table(ID INTEGER PRIMARY KEY AUTOINCREMENT, NAME TEXT,
ID
INTEGER
"ID" INTEGER
NAME
TEXT
"NAME" TEXT
SALARY
INTEGER
"SALARY" INTEGER
sqlite_sequence
CREATE TABLE sqlite_sequence(name,seq)
Indices (0)
Views (0)
Triggers (0)


DB Browser for SQLite - D:\Sayli_THAKUR_SEM_
File Edit View Tools Help
New Database
Open Database
Database Structure
Browse Data
Edit Pragmas
Table:
employee_table
ID
NAME
SALARY
Filter
Filter
Filter
1
1
Sayli Dholam
60000

Q9. Write a program to read all contacts using a content provider.

Aim: To create an Android application that reads all contacts from the device using a Content Provider.

Objective: The objective is to demonstrate the use of Android's Content Providers to fetch data from a phone's contact list and display it in the app using a Cursor and ListView.

Theory: Android provides Content Providers as a way to manage access to a structured set of data. One of the default Content Providers is the Contacts Content Provider, which gives access to the device's contact data. The ContentResolver class communicates with the Content Provider to query or retrieve data. The result of a query is returned as a Cursor, which is a pointer to a database-like table of results. With appropriate permissions, we can read all contacts and display them in the UI.

The app requires the READ_CONTACTS permission to access the user's contacts. A Button is used to trigger the action of loading contacts, and a ListView displays the list of contacts. The ContentResolver is used to query the ContactsContract Content Provider. A Cursor is returned, which is used to iterate over the contact entries. Each contact's name and phone number are retrieved using getColumnIndex() with constants from ContactsContract. The contacts are stored in an ArrayList and displayed using an ArrayAdapter in a ListView. The cursor.moveToNext() method is used to iterate through each row of the contact results

Code:

AndroidManifest.xml

```
<uses-permission
android:name="android.permission.READ_CONTACTS" />
```

MainActivity1.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:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:padding="16dp"
    tools:context=".MainActivity_contacts1">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/getData"
        android:text="Get Contacts"
        android:onClick="getContact"
        android:layout_gravity="center_horizontal" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="15dp"
        android:id="@+id/showtext"
        android:gravity="center"
        android:textAlignment="center"
        android:scrollbars="vertical" />

</LinearLayout>
```

MainActivity1.java

```
package com.example.assignment_3;

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

import android.annotation.SuppressLint;
import android.content.ContentResolver;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.text.method.ScrollingMovementMethod;
import android.view.View;
import android.widget.TextView;

public class MainActivity_contacts1 extends AppCompatActivity
{
    TextView tv;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_contacts1);
        tv=findViewById(R.id.showtext);
    }

    public void getContact(View view) {
        if(ContextCompat.checkSelfPermission(this,
android.Manifest.permission.READ_CONTACTS) !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new
String[]{android.Manifest.permission.READ_CONTACTS}, 10);
        }

        StringBuilder contact = new StringBuilder();
        ContentResolver contentResolver =
getContentResolver();
        Uri uri =
ContactsContract.CommonDataKinds.Phone.CONTENT_URI;
        Cursor cursor = contentResolver.query(uri, null, null,
null);
        if(cursor.getCount() > 0){
            while(cursor.moveToNext()){
                @SuppressLint("Range") String contactName =
```

```
cursor.getString(cursor.getColumnIndex(ContactsContract.Common
DataKinds.Phone.DISPLAY_NAME));
                @SuppressLint("Range") String contactNumber =
cursor.getString(cursor.getColumnIndex(ContactsContract.Common
DataKinds.Phone.NUMBER));
                tv.append("Name: " + contactName + " Number: "
+ contactNumber + "\n");
                contact.append("Name: " + contactName + "
Number: " + contactNumber + "\n");
            }
            tv.setMovementMethod(new
ScrollingMovementMethod());
        }
        Intent intent = new Intent(this,
MainActivity_contacts2.class);
        intent.putExtra("contact", contact.toString());
        startActivity(intent);
    }


}
```

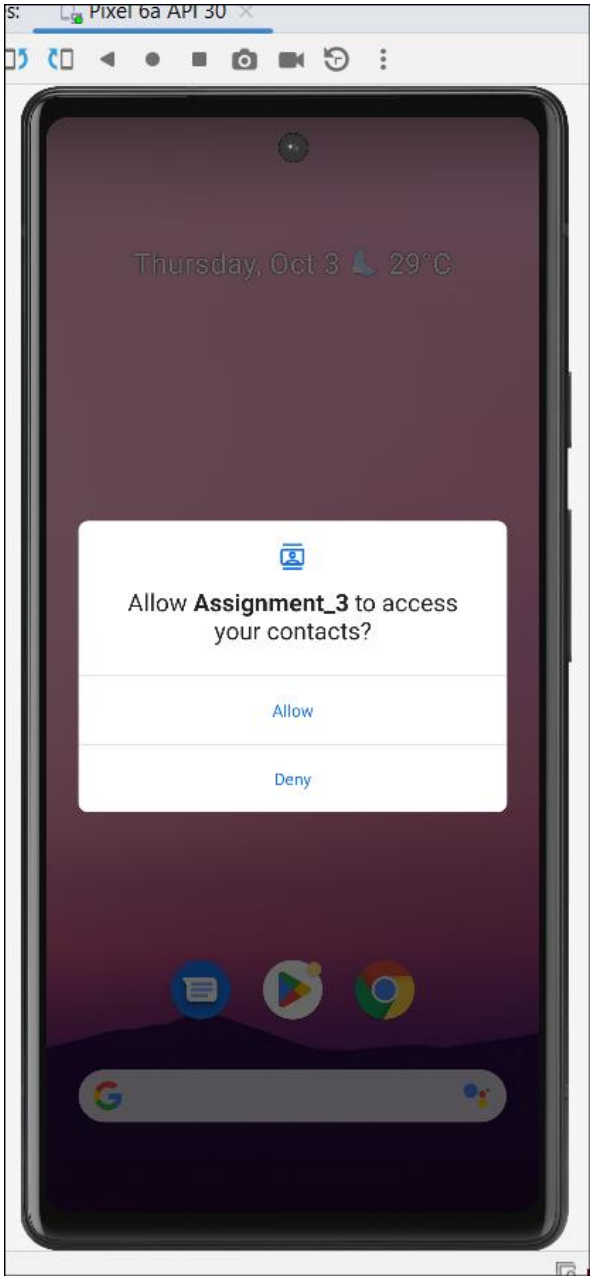
MainActivity2.xml

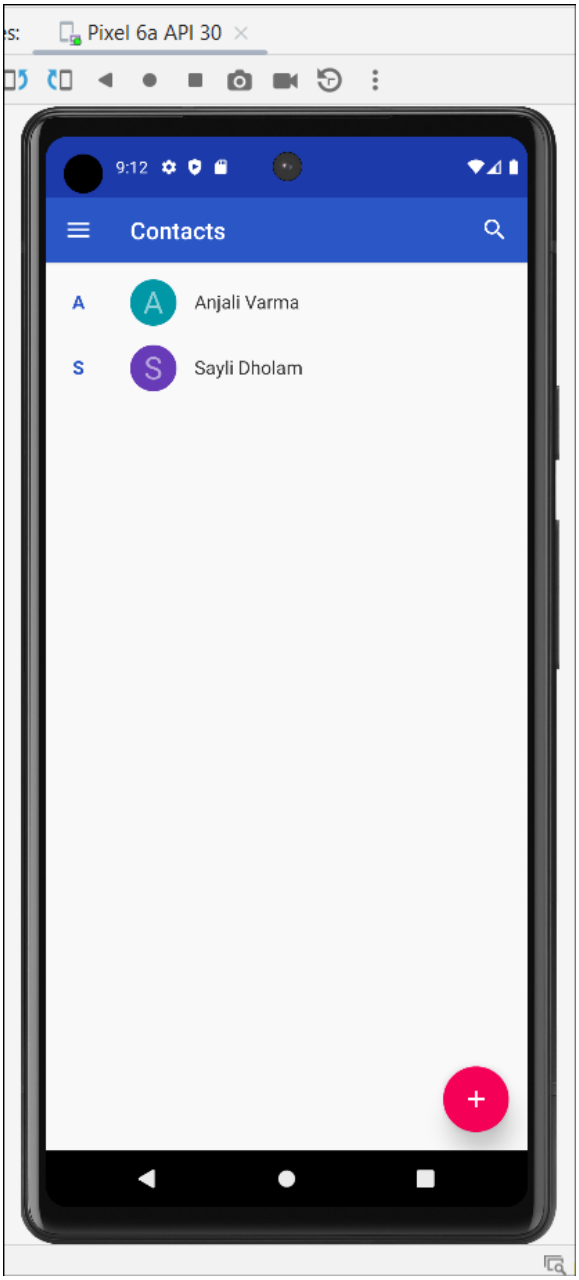
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity_contacts2">

    <TextView
        android:id="@+id/textContact"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:gravity="center"
        android:textAlignment="center"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />


</androidx.constraintlayout.widget.ConstraintLay
```

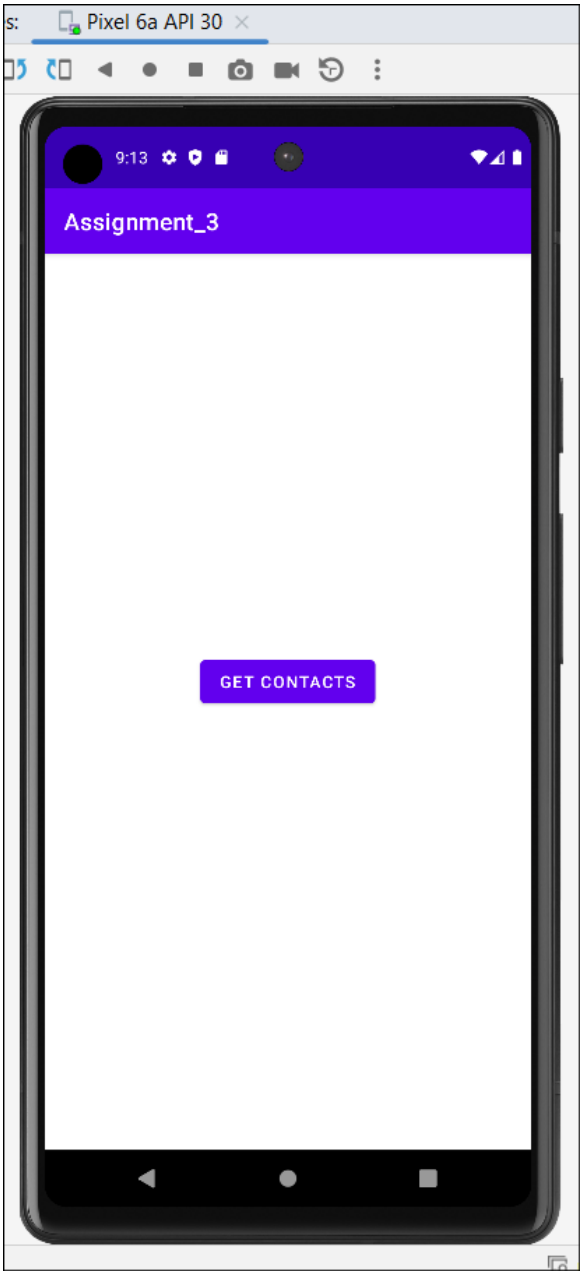
MainActivity2.java

```
package com.example.assignment_3;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.TextView;

public class MainActivity_contacts2 extends AppCompatActivity
{

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main_contacts2);
        String str = getIntent().getStringExtra("contact");
        TextView textContact = findViewById(R.id.textContact);
        textContact.setText(str);
    }
}
```

Output:


Pixel 6a API 30
Thursday, Oct 3 29°C
Allow Assignment_3 to access your contacts?
Allow
Deny


Pixel 6a API 30
9:12
Contacts
A
Anjali Varma
S
Sayli Dholam


Pixel 6a API 30
9:13
Assignment_3
GET CONTACTS


Pixel 6a API 30
9:13
Assignment_3
Name: Anjali Varma Number: (984) 627-3654
Name: Sayli Dholam Number: (996) 952-1693

Q10. Write a program to demonstrate JSON data parsing using HTTPUrlConnection (you can use https://api.github.com/users Json data).

Aim: To create an Android application that demonstrates how to fetch and parse JSON data from a server using HttpURLConnection and display the data in the application.

Objective: The main objective of this application is to understand how to:

1. Connect to a web server using HttpURLConnection.
2. Fetch JSON data from a web API.
3. Parse the JSON response.
4. Display the parsed data in the app.

Theory: AsyncTask: An abstract class that allows performing background operations and publishing results on the UI thread without handling threads directly.

doInBackground(): The method that runs the network operations in the background on a separate thread. It fetches data from the provided URL.

onPostExecute(): This method receives the result from doInBackground() and runs on the UI thread to update the user interface with the fetched data.

HttpURLConnection: A class that represents a connection to an HTTP server and allows you to send HTTP requests and receive responses.

InputStream: Reads the response from the server in a stream format.

BufferedReader: Used to read text from the InputStream efficiently.

URL: Represents the URL to which the HttpURLConnection connects.

Log.d(): Logs debugging messages to the console for monitoring the execution flow.

setMovementMethod(): Allows the TextView (tv) to scroll if the content exceeds the view height. ScrollingMovementMethod provides scrolling capability.

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.BLUETOOTH_CONNECT"/>
    <uses-permission
android:name="android.permission.BLUETOOTH"/>
    <uses-permission
android:name="android.permission.BLUETOOTH_ADMIN"/>
    <uses-permission
android:name="android.permission.INTERNET"/>

    <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.Assignment_3"
        tools:targetApi="31">

        <activity
            android:name=".MainActivity_JSONhttpurl"
            android:exported="true">
            <intent-filter>
                <action
android:name="android.intent.action.MAIN" />

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

</manifest>
```

MainActivity.xml

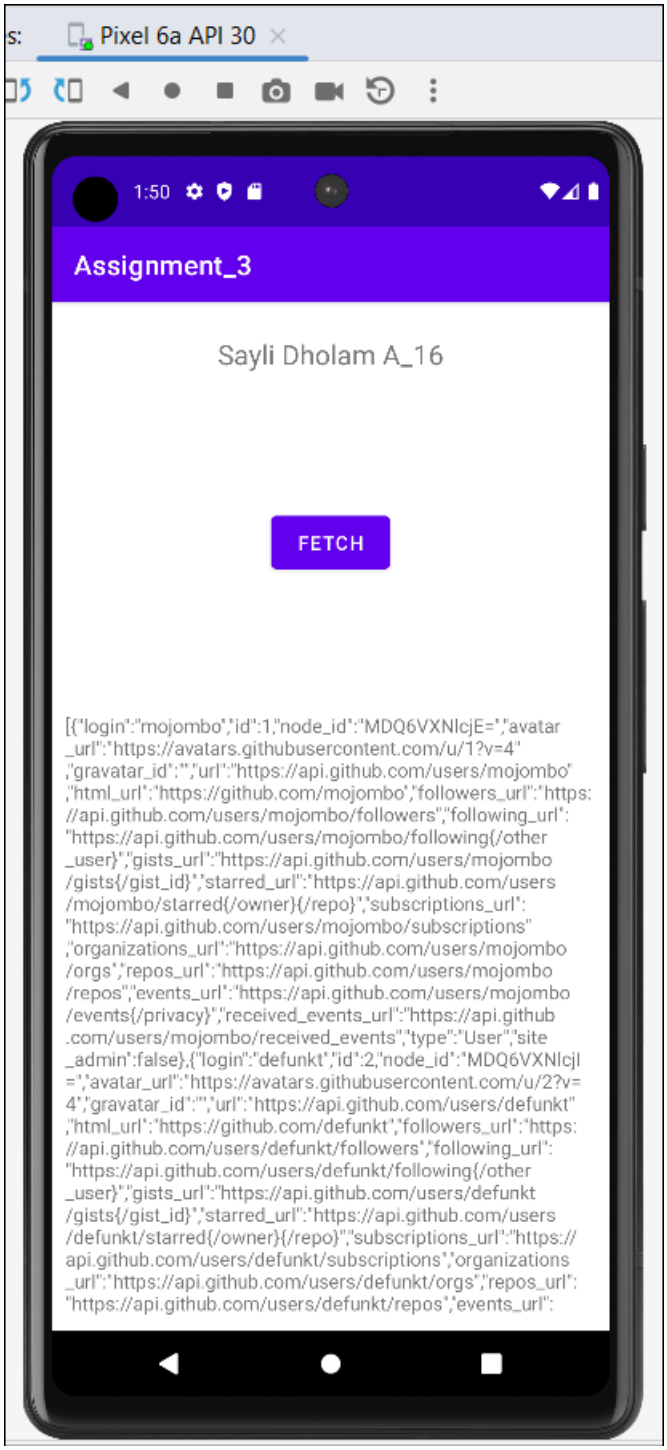
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity_JSONhttpurl">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sayli Dholam A_16"
        android:textSize="20dp"
        android:padding="10dp"
        android:layout_marginTop="15dp"
        android:layout_gravity="center"
        />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="FETCH"
        android:layout_marginTop="90dp"
        android:padding="10dp"
        android:onClick="FetchData"
        android:layout_gravity="center"
        />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TextView"
        android:layout_gravity="center"
        android:layout_marginTop="90dp"
        android:padding="10dp"
        />

</LinearLayout>
```

MainActivity.java

```
package com.example.assignment_3;

import androidx.appcompat.app.AppCompatActivity;

import android.os.AsyncTask;
import android.os.Bundle;
import android.text.method.ScrollingMovementMethod;
import android.util.Log;
import android.view.View;
import android.widget.TextView;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;

public class MainActivity_JSONhttpurl extends
AppCompatActivity {
    TextView tv;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_jsonhttpurl);
        tv = findViewById(R.id.textView2);
    }

    public void FetchData(View view) {
        RequestData req = new RequestData();
        req.execute();
    }

    class RequestData extends AsyncTask {
        @Override
        protected Object doInBackground(Object[] objects) {
            HttpURLConnection connection = null;
            BufferedReader reader = null;

            try {
                URL url = new
URL("https://api.github.com/users");
                connection = (HttpURLConnection)
url.openConnection();
                connection.connect();
```

```
                Log.d("request!", "starting");
                InputStream stream =
connection.getInputStream();
                reader = new BufferedReader(new
InputStreamReader(stream));

                StringBuffer buffer = new StringBuffer();
                String line = "";

                while ((line = reader.readLine()) != null) {
                    buffer.append(line);
                }
                return buffer.toString();
            } catch (MalformedURLException e) {
                throw new RuntimeException(e);
            } catch (IOException ee) {
                throw new RuntimeException(ee);
            }
        }

        @Override
        public void onPostExecute(Object o) {
            super.onPostExecute(o);
            tv.setText(o.toString());
            tv.setMovementMethod(new
ScrollingMovementMethod());

        }
    }
}
```

Output:


Pixel 6a API 30
1:50
Assignment_3
Sayli Dholam A_16
FETCH
[{"login":"mojombo","id":1,"node_id":"MDQ6VXNlcjE=","avatar
_url":"https://avatars.githubusercontent.com/u/1?v=4"
,"gravatar_id":"","url":"https://api.github.com/users/mojombo"
,"html_url":"https://github.com/mojombo","followers_url":"https:
//api.github.com/users/mojombo/followers","following_url":
"https://api.github.com/users/mojombo/following{/other
_user}","gists_url":"https://api.github.com/users/mojombo
/gists{/gist_id}","starred_url":"https://api.github.com/users
/mojombo/starred{/owner}{/repo}","subscriptions_url":
"https://api.github.com/users/mojombo/subscriptions"
,"organizations_url":"https://api.github.com/users/mojombo
/orgs","repos_url":"https://api.github.com/users/mojombo
/repos","events_url":"https://api.github.com/users/mojombo
/events{/privacy}","received_events_url":"https://api.github
.com/users/mojombo/received_events","type":"User","site
_admin":false},{"login":"defunkt","id":2,"node_id":"MDQ6VXNlcjI
=","avatar_url":"https://avatars.githubusercontent.com/u/2?v=
4","gravatar_id":"","url":"https://api.github.com/users/defunkt"
,"html_url":"https://github.com/defunkt","followers_url":"https:
//api.github.com/users/defunkt/followers","following_url":
"https://api.github.com/users/defunkt/following{/other
_user}","gists_url":"https://api.github.com/users/defunkt
/gists{/gist_id}","starred_url":"https://api.github.com/users
/defunkt/starred{/owner}{/repo}","subscriptions_url":"https://
api.github.com/users/defunkt/subscriptions","organizations
_url":"https://api.github.com/users/defunkt/orgs","repos_url":
"https://api.github.com/users/defunkt/repos","events_url":

Q11. Write a program to demonstrate JSON data parsing using OkHttp (you can use https://api.github.com/users Json data).

Aim: To demonstrate how to fetch and parse JSON data using OkHttp in an Android application

Objective: Learn how to integrate the OkHttp library in an Android project.

Fetch JSON data from a remote server.

Parse the JSON response and display it in the user interface.

Theory: OkHttp is an efficient HTTP client for Android that simplifies sending HTTP requests and receiving responses. JSON data is commonly used for communication between servers and clients. OkHttp helps us retrieve JSON data from the server, and then we can parse the data to extract useful information.

In your build.gradle file (Module: app), add the following dependency:

implementation ('com.squareup.okhttp3:okhttp:4.9.3')

Request: This class from OkHttp represents an HTTP request. The Request.Builder().url(url).build() builds the request to be sent to the server.

OkHttpClient: This is the main client class for making network requests in OkHttp. The newCall(request) method initiates a call with the given request.

enqueue(): This method is used to send the request asynchronously. It takes a Callback object to handle success or failure.

Callback: An interface that handles the asynchronous response of the request. It has two methods:

- onFailure(): Called when the request fails. The IOException e provides details about the failure.

- onResponse(): Called when the request succeeds. The Response object contains the server's response.

Response.body().string(): Extracts the response body as a string from the Response object.

runOnUiThread(): Ensures that UI updates (like setting the text in a TextView) happen on the main thread since network operations happen in the background.

TextView.setText(): Updates the text in the TextView with the result from the HTTP request.

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.INTERNET"/>

    <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.Assignment_3"
        tools:targetApi="31">

        <activity
            android:name=".MainActivity_JSONokhttp"
            android:exported="true">
            <intent-filter>
                <action
android:name="android.intent.action.MAIN" />

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

</manifest>
```

MainActivity.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:layout_height="match_parent"
    android:padding = "10dp"
    android:orientation = "vertical"
    tools:context=".MainActivity_JSONokhttp">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:layout_gravity="center"
        android:text="Sayli Dholam A_16" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="100dp"
        android:textSize="25dp"
        android:layout_gravity="center"
        android:text="Tevtview" />

</LinearLayout>
```

MainActivity.java

```
package com.example.assignment_3;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;

import androidx.annotation.NonNull;
import android.widget.TextView;
import java.io.IOException;
import okhttp3.Call;
import okhttp3.Callback;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.Response;

public class MainActivity_JSONokhttp extends AppCompatActivity
{
    TextView tv;

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

        tv = findViewById(R.id.textView2);

        OkHttpClient client = new OkHttpClient();
        String url = "https://api.github.com/users";
        Request request = new
Request.Builder().url(url).build();

        client.newCall(request).enqueue(new Callback() {
            @Override
            public void onFailure(@NonNull Call call, @NonNull
IOException e) {
                e.printStackTrace();
            }

            @Override
            public void onResponse(@NonNull Call call,
@NonNull Response response) throws IOException {
                final String myResponse =
response.body().string();
                MainActivity_JSONokhttp.this.runOnUiThread(new
Runnable() {
                    @Override
                    public void run() {
                        tv.setText(myResponse);
                    }
                });
```

```
            }
        });
    }
}
```

Build.gradle.kts(:app)

```
dependencies {

    implementation("androidx.appcompat:appcompat:1.7.0")

implementation("com.google.android.material:material:1.12.0")

implementation("androidx.constraintlayout:constraintlayout:2.1
.4")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.2.1")

androidTestImplementation("androidx.test.espresso:espresso-
core:3.6.1")

    implementation("com.squareup.okhttp3:okhttp:4.9.0")
}
```

Output:


Pixel 6a API 30
2:05
Assignment_3
Sayli Dholam A_16
[{"login":"mojombo","id":1,"node
_id":"MDQ6VXNlcjE=","avatar_url":
"https://avatars.githubusercontent
.com/u/1?v=4","gravatar_id":"","url":
"https://api.github.com/users
/mojombo","html_url":"https://
github.com/mojombo","followers
_url":"https://api.github.com/users
/mojombo/followers","following
_url":"https://api.github.com/users
/mojombo/following{/other_user}"
,"gists_url":"https://api.github.com
/users/mojombo/gists{/gist_id}"
,"starred_url":"https://api.github
.com/users/mojombo/starred{
/owner}{/repo}","subscriptions
_url":"https://api.github.com
/users/mojombo/subscriptions"
,"organizations_url":"https://api
.github.com/users/mojombo/orgs"

Q12. Write a program to demonstrate JSON data parsing using Volley (you can use https://api.github.com/users/mojombo Json data).

Aim: To demonstrate how to fetch data from a remote server using Volley and display an image from a URL using Picasso.

Objective: Fetch JSON data from a remote server using the Volley library.

Parse the JSON data.

Display text data from the JSON response.

Display an image using Picasso from a URL in the JSON response.

Theory: Volley is an HTTP library provided by Android that allows you to easily make asynchronous network requests. It is particularly useful for networking in Android apps. Picasso is a powerful image downloading and caching library for Android. It simplifies the process of loading images from external URLs.

add the necessary dependencies for Volley and Picasso in your build.gradle file:

```
dependencies {
    implementation 'com.android.volley:volley:1.2.1'
    implementation 'com.squareup.picasso:picasso:2.71828'
}
```

JsonObjectRequest: Sends an HTTP request to fetch a JSON object and processes the response.

Request.Method.GET: Specifies that the request method is GET for retrieving data.

JSONObject response: Represents the JSON data returned by the server.

getString("key"): Retrieves the value of the specified key from the JSON object as a string.

Picasso.get().load(img).into(iv): Loads the image from the provided URL and sets it into the specified ImageView.

Response.Listener<JSONObject>(): Defines the callback for handling the response when the JSON data is successfully fetched.

Response.ErrorListener(): Defines the callback for handling errors in the HTTP request.

mQueue.add(jReq): Adds the request to the request queue to be executed.

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.INTERNET" />

    <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.Assignment_3"
        tools:targetApi="31">

        <activity
            android:name=".MainActivity_JSONvolley"
            android:exported="true">
            <intent-filter>
                <action
android:name="android.intent.action.MAIN" />

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

</manifest>
```

MainActivity.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:layout_height="match_parent"
    android:padding = "10dp"
    android:orientation = "vertical"
    tools:context=".MainActivity_JSONvolley">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:layout_gravity="center"
        android:text="Sayli Dholam A_16" />

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="80dp"
        tools:srcCompat="@tools:sample/avatars" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="60dp"
        android:textSize="22dp"
        android:text="loading content..." />


    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="50dp"
        android:text="Load Data"
        android:onClick="Click_btn"/>

</LinearLayout>
```

MainActivity.java

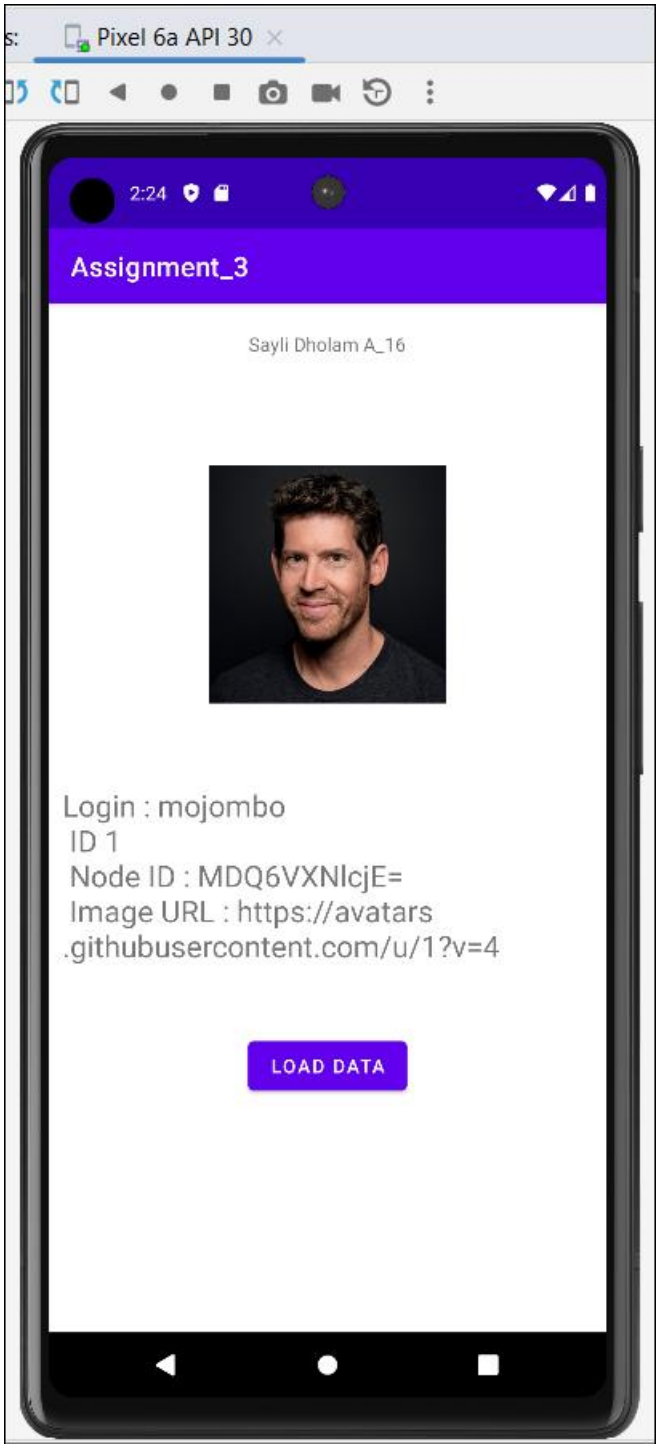
```
package com.example.assignment_3;

import androidx.appcompat.app.AppCompatActivity;

import android.app.DownloadManager;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.ImageView;

import com.android.volley.Request;
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.JsonObjectRequest;
import com.android.volley.toolbox.Volley;
import com.squareup.picasso.Picasso;

import org.json.JSONException;
import org.json.JSONObject;

public class MainActivity_JSONvolley extends AppCompatActivity
{

    TextView tv;
    ImageView iv;
    Button btn_load;
    private RequestQueue mQueue;

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

        tv = findViewById(R.id.textView2);
        iv = findViewById(R.id.imageView);
        btn_load = findViewById(R.id.button);
        mQueue = Volley.newRequestQueue(this);
    }

    public void Click_btn(View view){
        JsonParse();
    }

    private void JsonParse()
    {
```

```
        String url = "https://api.github.com/users/mojombo";
        JsonObjectRequest jReq = new
JsonObjectRequest(Request.Method.GET, url, null, new
Response.Listener<JSONObject>() {
            @Override
            public void onResponse(JSONObject response) {


                try {
                    String login =
response.getString("login");
                    String id = response.getString("id");
                    String node =
response.getString("node_id");
                    String img =
response.getString("avatar_url");
                    tv.setText("Login : " + login + "\n ID " +
id + "\n Node ID : " + node + "\n Image URL : " + img);
                    Picasso.get().load(img).into(iv);
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        }, new Response.ErrorListener()
        {
            @Override
            public void onErrorResponse(VolleyError error) {
                error.printStackTrace();
            }
        }
        );
        mQueue.add(jReq);
    }
}
```

Build.gradle.kts(:app)

```
dependencies {

    implementation("androidx.appcompat:appcompat:1.7.0")

implementation("com.google.android.material:material:1.12.0")

implementation("androidx.constraintlayout:constraintlayout:2.1
.4")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.2.1")

androidTestImplementation("androidx.test.espresso:espresso-
core:3.6.1")

    implementation("com.squareup.okhttp3:okhttp:4.9.0")

    implementation("com.android.volley:volley:1.2.1")
    implementation("com.squareup.picasso:picasso:2.71828")
}
```

Output:


Pixel 6a API 30
2:24
Assignment_3
Sayli Dholam A_16
Login : mojombo
ID 1
Node ID : MDQ6VXNlcjE=
Image URL : https://avatars
.githubusercontent.com/u/1?v=4
LOAD DATA

Q13. Write a program to demonstrate JSON data parsing using Retrofit (you can use https://api.github.com/users Json data).

Aim: To create an Android application that demonstrates JSON data parsing using the Retrofit library with an API endpoint (https://api.github.com/users).

Objective: To demonstrate how to use Retrofit for network calls and parsing JSON data

Theory: Retrofit is a type-safe HTTP client for Android and Java, developed by Square, used to simplify the network call process. It handles network requests and responses, makes asynchronous or synchronous requests, and simplifies JSON parsing by integrating with converters like Gson. By using Retrofit, we can easily fetch and handle JSON data from RESTful APIs.

Steps:

1. Add necessary dependencies to your project.
2. Create a model class for the JSON structure.
3. Set up Retrofit for making HTTP calls.
4. Parse the JSON data and display it in the UI.

Retrofit: Used for making HTTP requests. It's configured using Retrofit.Builder and connected to the base URL of the API.

GsonConverterFactory: This is used to convert the JSON response from the API into Java objects.

Call.enqueue(): This is an asynchronous method in Retrofit to make the network request and handle responses or failures.

Call<List<MyClass>>: Represents a request to retrieve a list of MyClass objects from the server.

onResponse(): Processes the received data and appends the login and node_id of each user to the TextView.

API interface: Defines the API endpoints with a method getRecords() to fetch data from the GitHub users API (/users).

Code:

MainActivity.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:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_marginTop="20dp"
    tools:context=".MainActivity_JSONretrofit">

    <TextView
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="50dp"
        android:text="Sayli Dholam A_16"
        android:textAlignment="center"/>

    <TextView
        android:id="@+id/retroView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

</LinearLayout>
```

API.java

```
package com.example.assignment_3;

import java.util.List;
import retrofit2.Call;
import retrofit2.http.GET;

public interface API {
    String BASE_URL="https://api.github.com/";
    @GET("users")
    Call<List<MyClass>> getRecords();

}
```

AndroidManifest.xml

```
<uses-permission android:name="android.permission.INTERNET" />
```

MainActivity.java

```
package com.example.assignment_3;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.TextView;
import android.widget.Toast;

import java.util.List;

import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;

public class MainActivity_JSONretrofit extends
AppCompatActivity {

    TextView tv;

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

        tv=findViewById(R.id.retroView);
        getRecords();
    }
    private void getRecords() {
        Retrofit retrofit=new
Retrofit.Builder().baseUrl(API.BASE_URL).addConverterFactory(G
sonConverterFactory.create()).build();
        API api =retrofit.create(API.class);
        Call<List<MyClass>> call=api.getRecords();
        call.enqueue(new Callback<List<MyClass>>()
        {
            @Override
            public void onResponse(Call<List<MyClass>> call,
Response<List<MyClass>> response) {
                List<MyClass> list = response.body();
                for (int i = 0; i < list.size(); i++) {
                    tv.append(list.get(i).getLogin() + ", " +
list.get(i).getNode_id());
                }
            }

            @Override
            public void onFailure(Call<List<MyClass>> call,
```

```
Throwable t) {
                Toast.makeText(MainActivity_JSONretrofit.this,
"No input", Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

MyClass.java

```
package com.example.assignment_3;

public class MyClass {
    String login;
    String node_id;
    public MyClass(String login, String node_id){
        this.login=login;
        this.node_id=node_id;
    }
    public String getLogin() {
        return login;
    }

    public void setLogin(String login) {
        this.login = login;
    }

    public String getNode_id() {
        return node_id;
    }

    public void setNode_id(String node_id) {
        this.node_id = node_id;
    }
}
```

Output:


Pixel 6a API 30
8:34
Assignment_3
Sayli Dholam A_16
mojombo, MDQ6VXNlcjE=defunkt, MDQ6VXNlcjI=pjhyett,
MDQ6VXNlcjM=wycats, MDQ6VXNlcjQ=ezmobius,
MDQ6VXNlcjU=ivey, MDQ6VXNlcjY=evanphx,
MDQ6VXNlcjc=vanpelt, MDQ6VXNlcjE3wayneeseguin,
MDQ6VXNlcjE4brynary, MDQ6VXNlcjE5kevinclark,
MDQ6VXNlcjIwtechnoweenie, MDQ6VXNlcjIxmacournoyer,
MDQ6VXNlcjIytakeo, MDQ6VXNlcjIzcaged,
MDQ6VXNlcjI1topfunky, MDQ6VXNlcjI2anotherjesse,
MDQ6VXNlcjI3roland, MDQ6VXNlcjI4lukas,
MDQ6VXNlcjI5fanvsfan, MDQ6VXNlcjMwtomtt,
MDQ6VXNlcjMxrailsjitsu, MDQ6VXNlcjMynitay,
MDQ6VXNlcjM0kevwil, MDQ6VXNlcjM1KirinDave,
MDQ6VXNlcjM2jamesgolick, MDQ6VXNlcjM3atmos,
MDQ6VXNlcjM4errfree, MDEyOk9yZ2FuaXphdGlvbjQ0mojodna,
MDQ6VXNlcjQ1bmizerany, MDQ6VXNlcjQ2

Q14. Write a program to demonstrate camera access in android studio.

Aim: To develop an Android application that demonstrates how to access the camera and capture an image using an Intent.

Objective: To understand how to use implicit intents to access the camera. To capture and display the image taken from the camera in an ImageView.

Theory: In Android, the camera is an essential part of mobile hardware that allows users to capture images or record videos. To interact with the camera, Android provides two primary ways: using the Camera API directly or by using intents to invoke the device's native camera app.

When using intents, the app does not need to manage the camera directly. Instead, it can request the default camera application to capture images or videos. Once the image is captured, the result (image) is returned to the activity, which can then display it or save it. The necessary permissions and handling of the camera are essential for Android apps, which requires including permissions for accessing the camera in the manifest file.

Implicit Intent: The intent MediaStore.ACTION_IMAGE_CAPTURE is used to launch the camera app and capture an image.

startActivityForResult(): This method is used to start the camera activity and return the captured image as a result.

onActivityResult(): This callback method handles the result from the camera app, where the captured image is returned as a Bitmap and displayed in the ImageView.

Permissions: The camera permission is requested in the AndroidManifest.xml file using <uses-permission android:name="android.permission.CAMERA" />.

Code:

AndroidManifest.xml

```
<uses-feature
    android:name="android.hardware.camera"
    android:required="false" />
<uses-permission android:name="android.permission.CAMERA"/>
```

MainActivity.xml

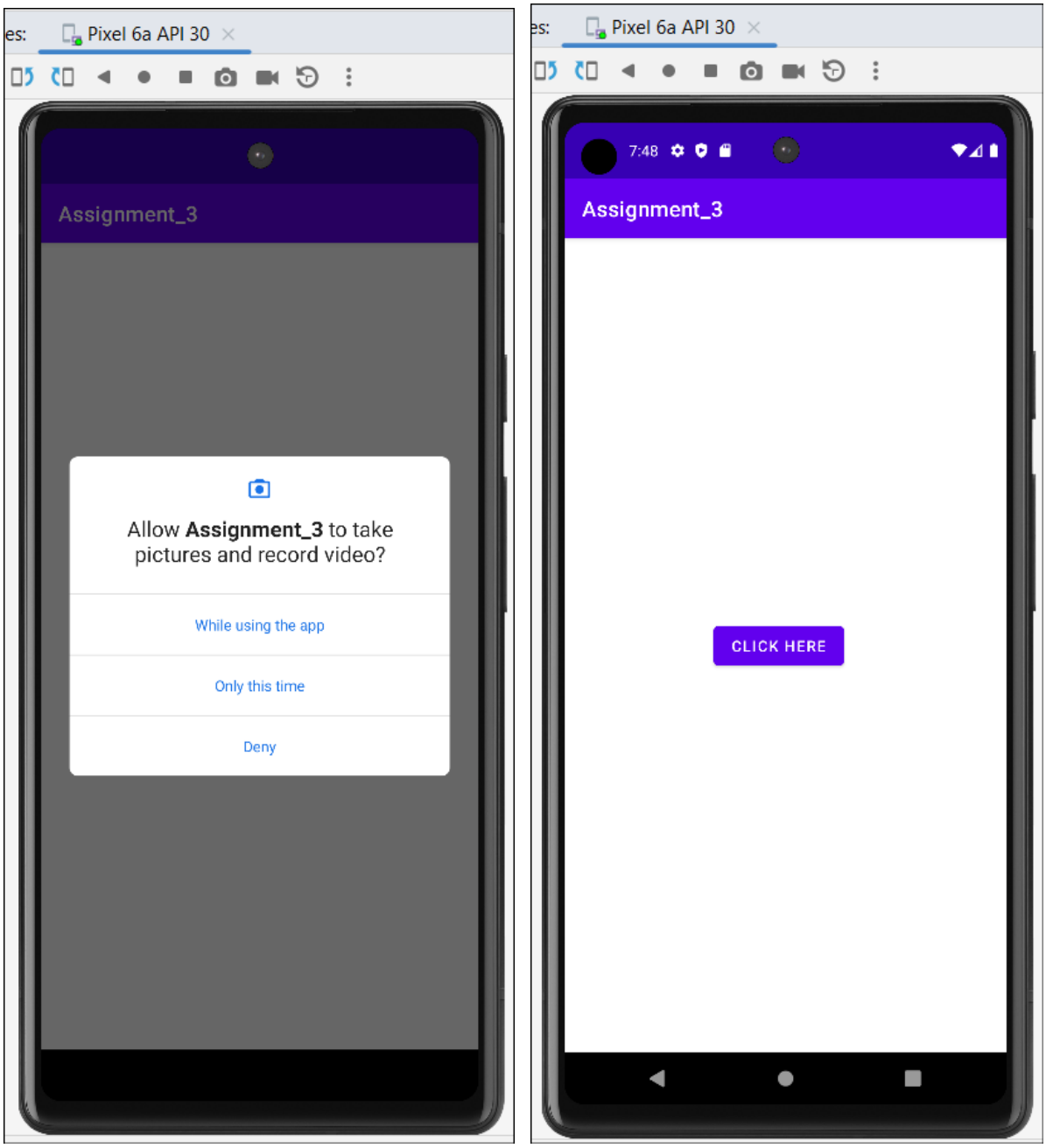
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity_camera">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="262dp"
        android:layout_height="222dp"
        app:layout_constraintBaseline_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias=".578"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:srcCompat="@tools:sample/avatars" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="click here"
        android:onClick="click"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.478"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:flow_firstVerticalBias=".74"
        />

</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

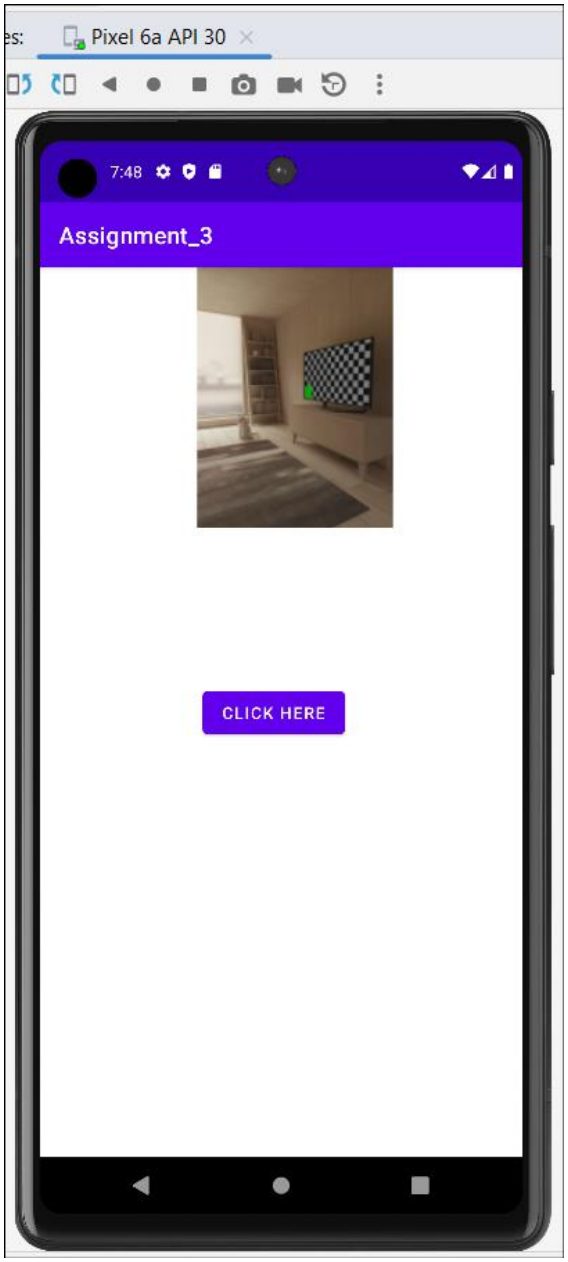
```
package com.example.assignment_3;

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

import android.app.Activity;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Bitmap;
import android.os.Bundle;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;

import org.jetbrains.annotations.NotNull;

public class MainActivity_camera extends AppCompatActivity {
    ImageView img;
    Button btn;
    View view;

    private static final  int CAMERA_REQUEST = 150;

    private static final int MY_CAMERA_PERMISSION_CODE = 100;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_camera);
        img = findViewById(R.id.imageView);
        btn = findViewById(R.id.button);


if(ContextCompat.checkSelfPermission(this,android.Manifest.per
mission.CAMERA) != PackageManager.PERMISSION_GRANTED) {

            ActivityCompat.requestPermissions(this, new
String[] {android.Manifest.permission.CAMERA},
MY_CAMERA_PERMISSION_CODE);

        }
    }

    public void onRequestPermissionResult(int requestCode,
```

```
@NotNull String[] permissions, @NotNull int[] grantResults){
        super.onRequestPermissionsResult(requestCode,
permissions,grantResults);

        if(requestCode == MY_CAMERA_PERMISSION_CODE){
            Toast.makeText(this, "Permission is Granted",
Toast.LENGTH_LONG).show();
        }else{
            Toast.makeText(this, "Permission Denied",
Toast.LENGTH_LONG).show();
        }
    }

    public void click(View view){
        Intent intent = new
Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        startActivityForResult(intent,CAMERA_REQUEST);
    }

    protected void onActivityResult(int requestCode, int
resultCode , @NotNull Intent data){

        super.onActivityResult(requestCode,resultCode , data);
        if(requestCode == CAMERA_REQUEST && resultCode ==
Activity.RESULT_OK){
            Bitmap bmap = (Bitmap)
data.getExtras().get("data");
            img.setImageBitmap(bmap);
        }
    }
}
```

Output:


Pixel 6a API 30
Assignment_3
Allow Assignment_3 to take pictures and record video?
While using the app
Only this time
Deny
Pixel 6a API 30
7:48
Assignment_3
CLICK HERE


Pixel 6a API 30
control camera


Pixel 6a API 30
7:48
Assignment_3
CLICK HERE

Q15. Write a program to demonstrate audio recording in android studio.

Aim: To create an Android application that demonstrates how to record and play back audio using the Android MediaRecorder and MediaPlayer classes.

Objective: To understand how to use the MediaRecorder class for recording audio. To use the MediaPlayer class to play back recorded audio. To handle runtime permissions for recording audio and accessing storage.

Theory: MediaRecorder: This class is used for audio and video capture. It provides a simple API to start, stop, and manage the recording process. The audio can be saved in various formats, such as .3gp.

MediaPlayer: This class is used to control the playback of audio and video files. You can play, pause, and stop media, and even stream audio from a URL.

Permissions: Android requires applications to request certain permissions at runtime, especially for accessing sensitive features like the microphone and external storage.

File Storage: Depending on the Android version, recorded audio is saved in either internal or external storage, typically within the app-specific directory.

Code:

AndroidManifest.xml

```
<uses-permission
android:name="android.permission.RECORD_AUDIO"/>
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

MainActivity.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:layout_height="match_parent"
    android:padding = "10dp"
    android:orientation = "vertical"
    tools:context=".MainActivity_audiorecord">

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="50dp"
        android:text="Start recording"
        android:onClick="Start_record"/>

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="50dp"
        android:text="Stop recording"
        android:onClick="Stop_record"/>

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="50dp"
        android:text="Play last recording"
        android:onClick="PlayLast_record"/>

    <Button
        android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="50dp"
        android:text="stop playing recording"
        android:onClick="PlayStop_record"/>

</LinearLayout>
```

MainActivity.java

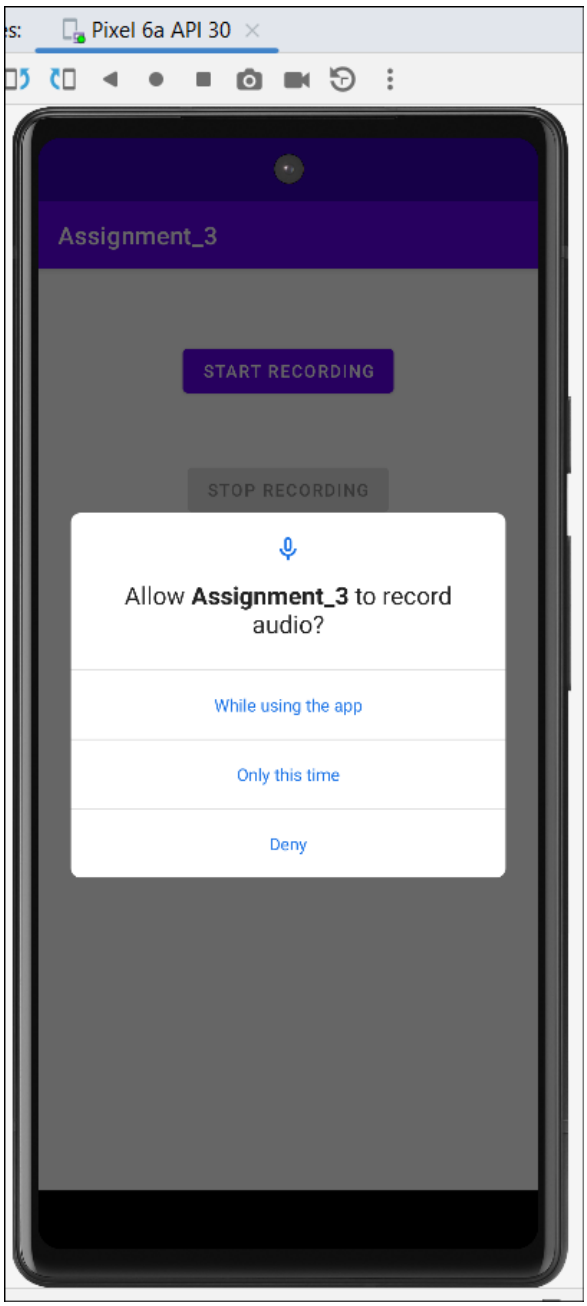
```
package com.example.assignment_3;

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.media.MediaRecorder;
import android.os.Bundle;
import android.os.Environment;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import org.jetbrains.annotations.NotNull;

import java.io.IOException;
import java.util.UUID;


public class MainActivity_audiorecord extends
AppCompatActivity {
    Button btnStart;
    Button btnStop;
    Button btnPlaylastRecordAudio;
    Button btnStopLastRecording;
    String AudioSavePathInDevice = null;
    MediaRecorder mediaRecorder;
    MediaPlayer mediaPlayer;
    public static final int RequestPermissionCode = 1;
    boolean permission_granted = false;

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

        // Initialize buttons
        btnStart = findViewById(R.id.button1);
        btnStop = findViewById(R.id.button2);
        btnPlaylastRecordAudio = findViewById(R.id.button3);
        btnStopLastRecording = findViewById(R.id.button4);

        // Set button initial states
        btnStop.setEnabled(false);
        btnPlaylastRecordAudio.setEnabled(false);
```

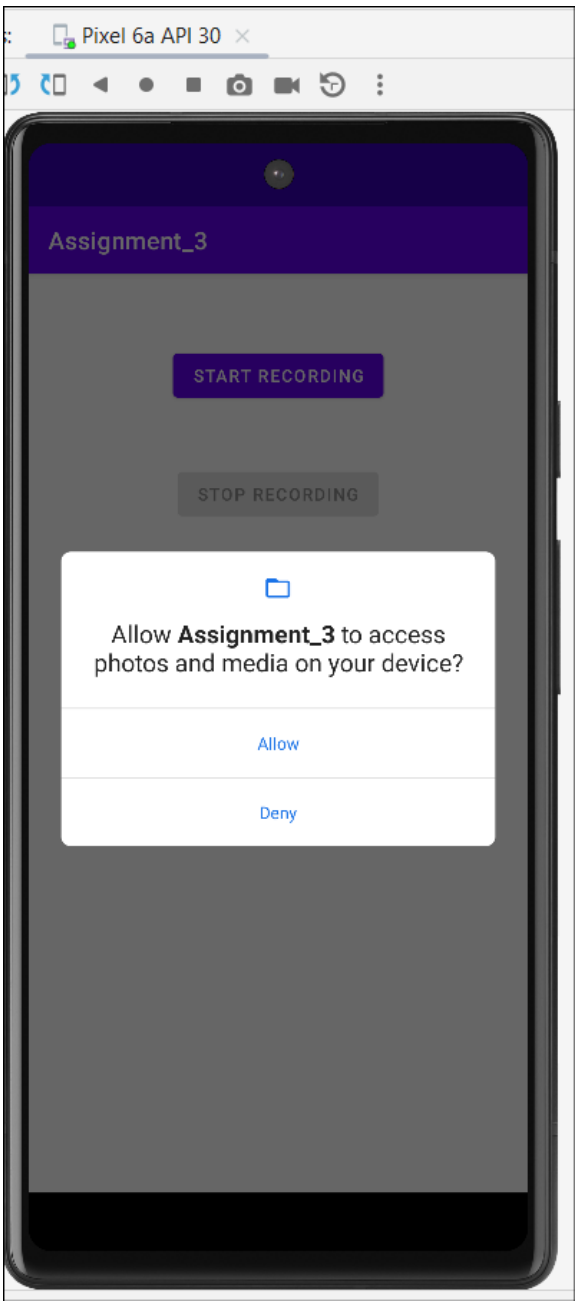
```
        btnStopLastRecording.setEnabled(false);

        // Check permissions and request if not granted
        if (ContextCompat.checkSelfPermission(this,
android.Manifest.permission.RECORD_AUDIO) !=
PackageManager.PERMISSION_GRANTED
                || ContextCompat.checkSelfPermission(this,
android.Manifest.permission.WRITE_EXTERNAL_STORAGE) !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new
String[]{android.Manifest.permission.RECORD_AUDIO,

android.Manifest.permission.WRITE_EXTERNAL_STORAGE}, 100);
        } else {
            permission_granted = true;
            Toast.makeText(this, "Permission already granted",
Toast.LENGTH_LONG).show();
        }
    }

    @Override
    public void onRequestPermissionsResult(int requestCode,
@NotNull String[] permissions, @NotNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode,
permissions, grantResults);
        if (requestCode == 100) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED &&
                    grantResults[1] ==
PackageManager.PERMISSION_GRANTED) {
                permission_granted = true;
                Toast.makeText(this, "All Permissions
granted", Toast.LENGTH_LONG).show();
            } else {
                permission_granted = false;
                Toast.makeText(this, "Please grant all
permissions", Toast.LENGTH_LONG).show();
            }
        }
    }

    // Start recording audio
    public void Start_record(View view) {
        if (permission_granted) {
            // Save audio in app-specific external directory
(for Android 10+ compatibility)
            AudioSavePathInDevice =
getExternalFilesDir(null).getAbsolutePath() + "/" +
UUID.randomUUID() + "_Recording.3gp";
            mediaRecorder = new MediaRecorder();
```

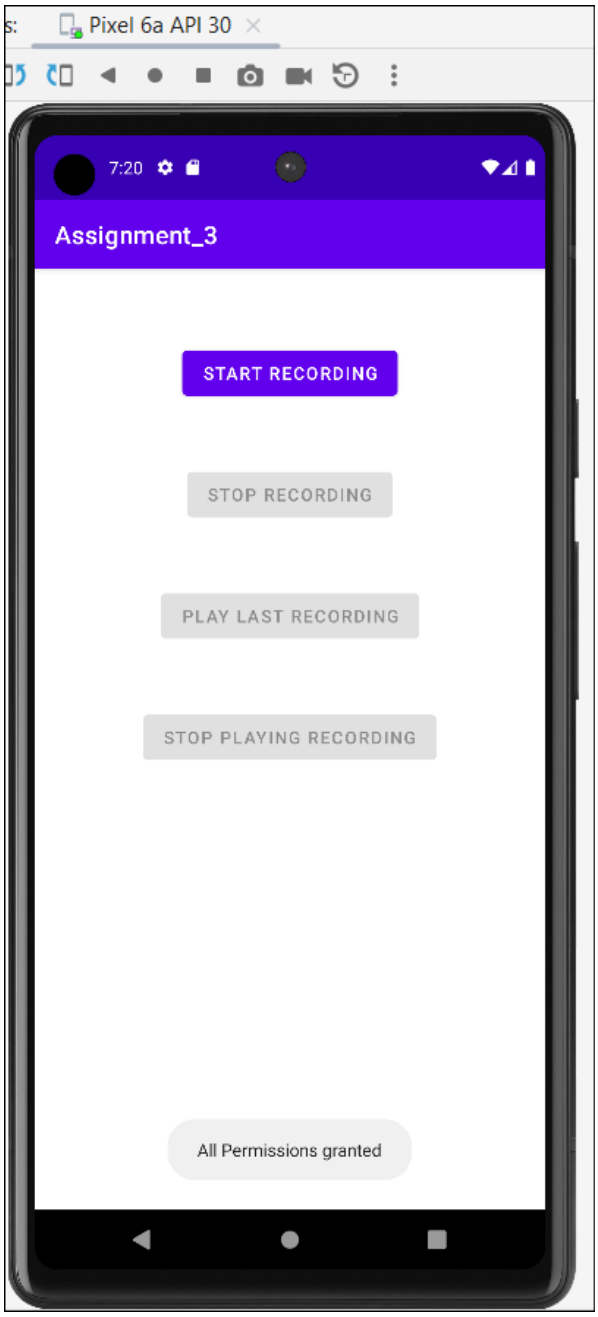
```
mediaRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);

mediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE
_GPP);

mediaRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_N
B);

mediaRecorder.setOutputFile(AudioSavePathInDevice);

            try {
                mediaRecorder.prepare();
                mediaRecorder.start();
                btnStart.setEnabled(false);
                btnStop.setEnabled(true);
                Toast.makeText(this, "Recording Started",
Toast.LENGTH_LONG).show();
            } catch (IOException | IllegalStateException e) {
                e.printStackTrace();
            }
        } else {
            ActivityCompat.requestPermissions(this, new
String[]{android.Manifest.permission.RECORD_AUDIO,

android.Manifest.permission.WRITE_EXTERNAL_STORAGE},
RequestPermissionCode);
        }
    }

    // Stop recording audio
    public void Stop_record(View view) {
        try {
            mediaRecorder.stop();
            mediaRecorder.release();
            mediaRecorder = null;
            btnStart.setEnabled(true);
            btnStop.setEnabled(false);
            btnPlaylastRecordAudio.setEnabled(true);
            btnStopLastRecording.setEnabled(false);
            Toast.makeText(this, "Recording Completed",
Toast.LENGTH_LONG).show();
        } catch (IllegalStateException e) {
            e.printStackTrace();
        }
    }

    // Play the last recorded audio
    public void PlayLast_record(View view) {
        btnStop.setEnabled(false);
        btnStart.setEnabled(false);
        btnPlaylastRecordAudio.setEnabled(false);
```

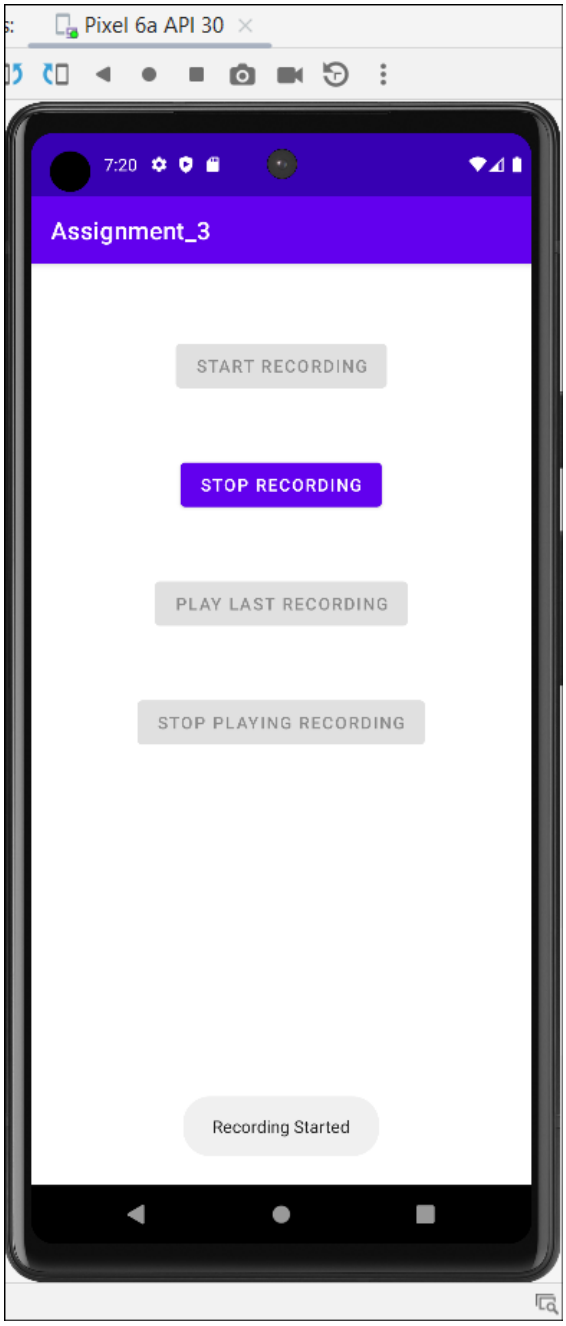
```
            btnStopLastRecording.setEnabled(true);

            mediaPlayer = new MediaPlayer();
            try {
                mediaPlayer.setDataSource(AudioSavePathInDevice);
                mediaPlayer.prepare();
                mediaPlayer.start();
                Toast.makeText(this, "Recorded Audio Playing",
Toast.LENGTH_LONG).show();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }

        // Stop playing the audio
        public void PlayStop_record(View view) {
            btnStop.setEnabled(false);
            btnStart.setEnabled(true);
            btnPlaylastRecordAudio.setEnabled(true);
            btnStopLastRecording.setEnabled(false);

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

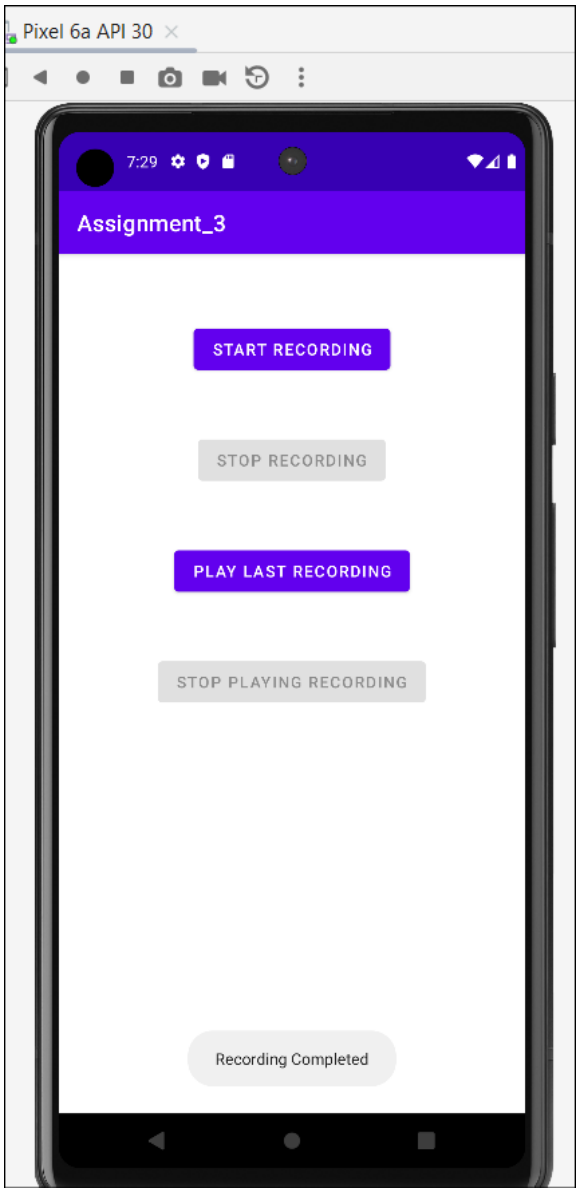
Output:

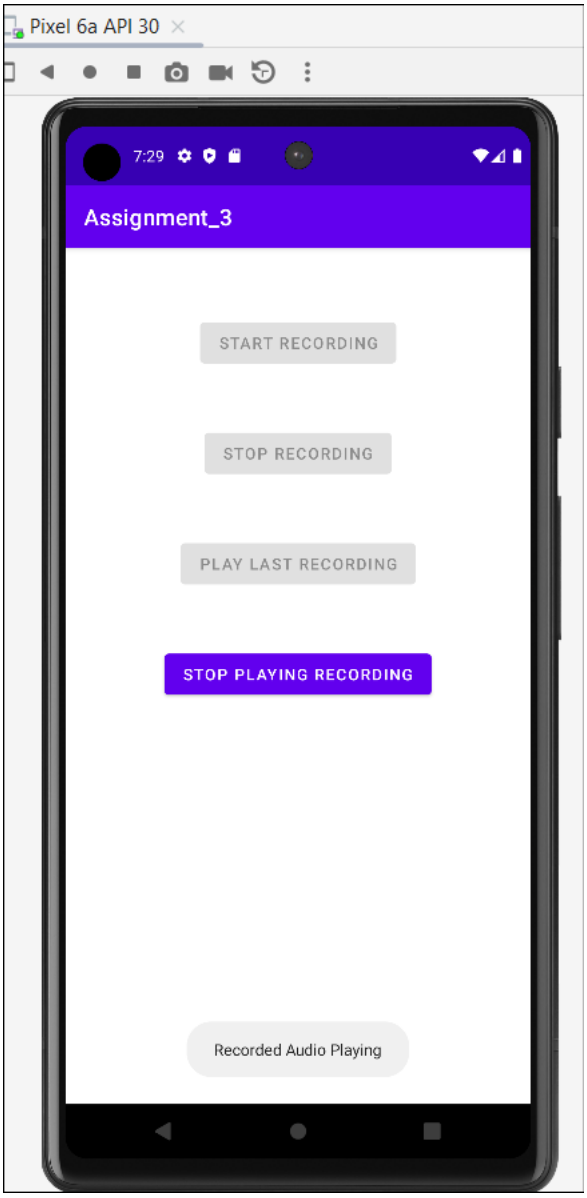

Pixel 6a API 30
Assignment_3
START RECORDING
STOP RECORDING
Allow Assignment_3 to record audio?
While using the app
Only this time
Deny

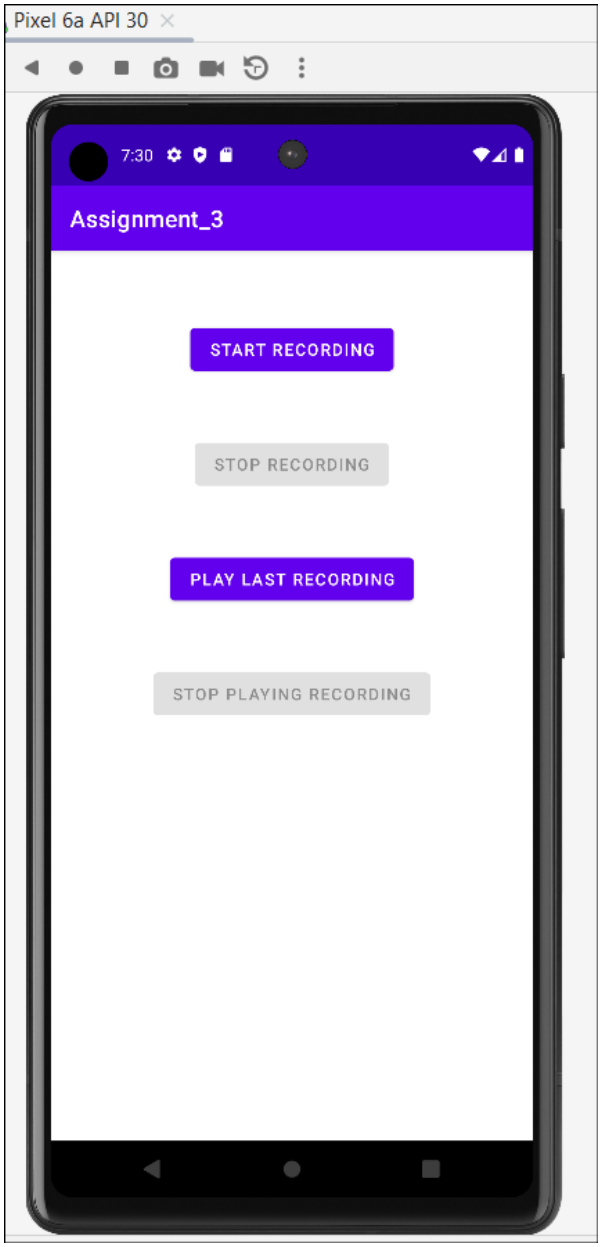

Pixel 6a API 30
Assignment_3
START RECORDING
STOP RECORDING
Allow Assignment_3 to access photos and media on your device?
Allow
Deny

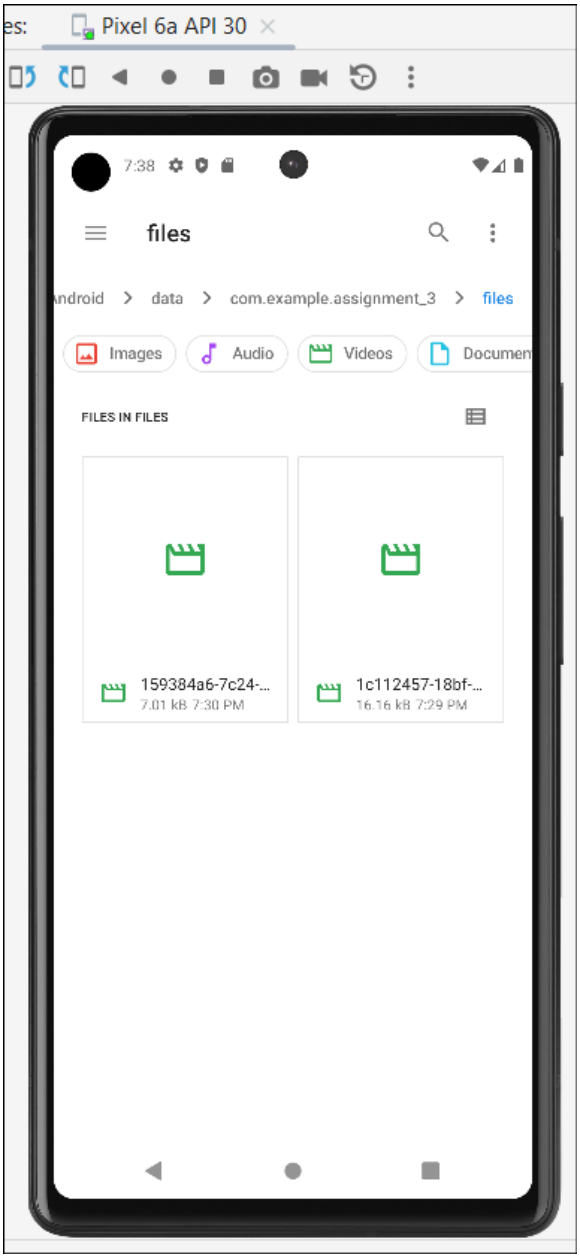

Pixel 6a API 30
7:20
Assignment_3
START RECORDING
STOP RECORDING
PLAY LAST RECORDING
STOP PLAYING RECORDING
All Permissions granted

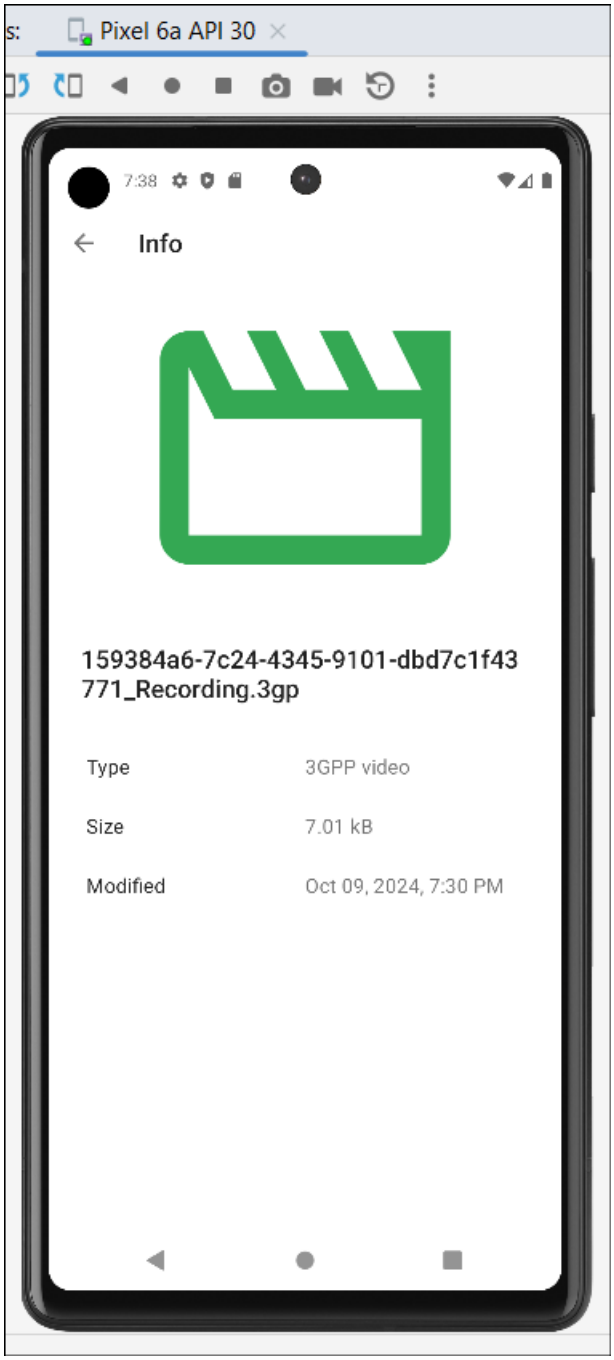

Pixel 6a API 30
7:20
Assignment_3
START RECORDING
STOP RECORDING
PLAY LAST RECORDING
STOP PLAYING RECORDING
Recording Started


Pixel 6a API 30
7:29
Assignment_3
START RECORDING
STOP RECORDING
PLAY LAST RECORDING
STOP PLAYING RECORDING
Recording Completed


Pixel 6a API 30
7:29
Assignment_3
START RECORDING
STOP RECORDING
PLAY LAST RECORDING
STOP PLAYING RECORDING
Recorded Audio Playing


Pixel 6a API 30
7:30
Assignment_3
START RECORDING
STOP RECORDING
PLAY LAST RECORDING
STOP PLAYING RECORDING


Pixel 6a API 30
7:38
files
ndroid > data > com.example.assignment_3 > files
Images
Audio
Videos
Documen
FILES IN FILES
159384a6-7c24-...
7.01 kB 7:30 PM
1c112457-18bf-...
16.16 kB 7:29 PM



All apps > files > menu > Sdk_gphone_x68 > android > data > com.example.assignment_3 > files > click open > hold select > 3 dots > get info

****