Example – External Storage

Creates a file in external memory (SD Card) to store the details

1. activity_main.xml

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


    <Button
        android:id="@+id/saveButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editText_data"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_marginStart="48dp"
        android:layout_marginLeft="48dp"
        android:layout_marginTop="8dp"
        android:padding="8dp"
        android:text="Save"
        android:textAllCaps="false"/>

    <Button
        android:id="@+id/viewButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editText_data"
        android:layout_centerHorizontal="false"
        android:layout_marginStart="50dp"
        android:layout_marginLeft="50dp"
        android:layout_marginTop="8dp"
        android:layout_toEndOf="@id/saveButton"
        android:layout_toRightOf="@id/saveButton"
        android:padding="8dp"
        android:text="View"
        android:textAllCaps="false" />

    <EditText
        android:id="@+id/editText_data"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textView_data"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="8dp"
        android:autofillHints=""
        android:hint="Enter text data"
        android:inputType="textMultiLine" />

    <TextView
        android:id="@+id/textView_data"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="8dp"
        android:text="External Storage"
        android:textAllCaps="true"
        android:textStyle="bold" />

</RelativeLayout>
```

2. AndroidManifest.xml

//Permissions to read and write to external storage are to be set.

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

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.ExampleExternalStorage">

        <activity
android:name=".ViewInformationActivity"></activity>
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

3. MainActivity.java

```
package com.example.exampleexternalstorage1;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
```

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

public class MainActivity extends AppCompatActivity {
    EditText editText_data;
    Button savebutton, viewbutton;

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

        editText_data = findViewById(R.id.editText_data);
        savebutton = findViewById(R.id.saveButton);
        viewbutton = findViewById(R.id.viewButton);


        savebutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String datasaved =
editText_data.getText().toString();
                File folder = getExternalFilesDir("PracticalData");
                File file = new File(folder, "f2.txt");
                writeTextData(file, datasaved);
                editText_data.setText("");
            }
        });


        viewbutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

            // PracticalData represents the folder name to access
privately saved data
                File folder = getExternalFilesDir("PracticalData");

                // f2.txt is the file that is saved privately
                File file = new File(folder, "f2.txt");
                String data = getdata(file);
                if (data != null) {
                    editText_data.setText(data);

Toast.makeText(getApplicationContext(),data,Toast.LENGTH_LONG).show(
);
                } else {
                    editText_data.setText("No data available");
                 //  Toast.makeText(getApplicationContext(),"No data
available",Toast.LENGTH_LONG).show();
                }
            }
        });
    }

// writeTextData() method to save the data into the file in byte
format
  // It also toast a message “Done/filepath_where_the_file_is_saved"
```

```
    private void writeTextData(File file, String data) {
        FileOutputStream fileOutputStream = null;
        try {
            fileOutputStream = new FileOutputStream(file);
            fileOutputStream.write(data.getBytes());
            Toast.makeText(this, "Done" + file.getAbsolutePath(),
Toast.LENGTH_LONG).show();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (fileOutputStream != null) {
                try {
                    fileOutputStream.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }

// getdata() is the method which reads the data
    // the data that is saved in byte format in the file

    private String getdata(File myfile) {
        FileInputStream fileInputStream = null;
        try {
            fileInputStream = new FileInputStream(myfile);
            int i = -1;
            StringBuffer buffer = new StringBuffer();
            while ((i = fileInputStream.read()) != -1) {
                buffer.append((char) i);
            }
            return buffer.toString();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (fileInputStream != null) {
                try {
                    fileInputStream.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
        return null;
    }
}
```