

Practical No. 6

Title: Android program using Shared Preferences, Internal and External Storage

Aim: Create an application to demonstrate Shared Preferences, Internal and External Storage

Introduction

Android provides many ways of storing data of an application. One of this way is called Shared Preferences. Shared Preferences allow you to save and retrieve data in the form of key, value pair.

In order to use shared preferences, you have to call a method `getSharedPreferences()` that returns a `SharedPreferences` instance pointing to the file that contains the values of preferences.

```
SharedPreferences sharedPreferences = getSharedPreferences(MyPREFERENCES, Context.MODE_PRIVATE);
```

The first parameter is the key and the second parameter is the MODE. Apart from private there are other modes available that are listed below –

Sr.No	Mode & description
1	MODE_APPEND This will append the new preferences with the already existing preferences
2	MODE_ENABLE_WRITE_AHEAD_LOGGING Database open flag. When it is set, it would enable write ahead logging by default
3	MODE_MULTI_PROCESS

	This method will check for modification of preferences even if the sharedpreference instance has already been loaded
4	MODE_PRIVATE By setting this mode, the file can only be accessed using calling application
5	MODE_WORLD_READABLE This mode allows other application to read the preferences
6	MODE_WORLD_WRITEABLE This mode allows other application to write the preferences

You can save something in the sharedpreferences by using SharedPreferences.Editor class. You will call the edit method of SharedPreferences instance and will receive it in an editor object.

Its syntax is –

```
Editor editor = sharedPreferences.edit();  
editor.putString("key", "value");  
editor.commit();
```

Apart from the putString method, there are methods available in the editor class that allows manipulation of data inside shared preferences. They are listed as follows –

Sr. NO	Mode & description
1	apply() It is an abstract method. It will commit your changes back from editor to the sharedPreferences object you are calling
2	clear()

	It will remove all values from the editor
3	remove(String key) It will remove the value whose key has been passed as a parameter
4	putLong(String key, long value) It will save a long value in a preference editor
5	putInt(String key, int value) It will save an integer value in a preference editor
6	putFloat(String key, float value) It will save a float value in a preference editor

Exercise - Create an application to save data in internal or external storage and also demonstrate implementation of shared preferences

Implementation:

Program:

activity_main.xml

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

```
<TextView
    android:id="@+id/txtVwUN"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/user_name"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.24"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.214" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/password"
    android:id="@+id/txtVwPass"
    android:textSize="18sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.245"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.294" />
```

```
<EditText
    android:autofillHints="name"
    android:inputType="text"
    android:id="@+id/edVwUser"
    android:layout_width="177dp"
    android:layout_height="48dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.799"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.203"
    android:hint="@string/enter_user_name"
```

```
/>
```

```
<EditText
```

```
    android:autofillHints="password"
    android:inputType="textPassword"
    android:id="@+id/edVwPass"
    android:layout_width="177dp"
    android:layout_height="48dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.799"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.282"
    android:hint="@string/enter_password"/>
```

```
<CheckBox
```

```
    android:id="@+id/chkMe"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/remember_me"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.468"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.38" />
```

```
<Button
```

```
    android:id="@+id/saveIt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/show_me"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.47"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.468"
    />
```

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

activity_file_handling.xml

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

<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context=".FileHandling">

    <TextView
        android:id="@+id/DataVw"
        android:layout_width="225dp"
        android:layout_height="64dp"
        android:fontFamily="sans-serif"
        android:text="@string/click_on_the_button_to_cleared_the_saved_data"
        android:textAlignment="center"
        android:textSize="18sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.322" />

    <TextView
        android:id="@+id/clrDataVw"
        android:layout_width="225dp"
```

```
        android:layout_height="64dp"
        android:fontFamily="sans-serif"
        android:text=""
        android:textAlignment="center"
        android:textSize="18sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.631"/>
<Button
    android:id="@+id/clearIt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/clear_data"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.468" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.sharedpredemo;

import android.content.Intent;
import android.content.SharedPreferences;
```

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

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {
    public static final String savePref = "myPref";
    public static final String userNameSP = "UsrNm";
    public static final String userPasswordSP = "UsrPass";
    SharedPreferences sharedPreferences;
    EditText txtUsrNm, txtUsrPass;

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

        txtUsrNm = findViewById(R.id.edVwUser);
        txtUsrPass = findViewById(R.id.edVwPass);
        Button btnSubmit = findViewById(R.id.savelt);
        CheckBox chkBxRMe = findViewById(R.id.chkMe);

        // Initialize SharedPreferences once in onCreate
        sharedPreferences = getSharedPreferences(savePref, MODE_PRIVATE);

        btnSubmit.setOnClickListener(view -> {
            if (chkBxRMe.isChecked()) {
                saveSharedPref();
            }
            // Explicit Intent to navigate to FileHandling activity
            Intent intent = new Intent(MainActivity.this, FileHandling.class);
            startActivity(intent);
        });
    }
}
```



```
// Set edge-to-edge layout for full-screen content
ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) ->
{
    Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
    v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
    return insets;
});
}

public void saveSharedPref() {
    String nm = txtUsrNm.getText().toString();
    String pass = txtUsrPass.getText().toString();
    // Data is stored in the preferences
    SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.putString(userNameSP, nm);
    editor.putString(userPasswordSP, pass);
    editor.apply();
}

@Override
protected void onResume() {
    super.onResume();
    // Retrieve saved values and set them in EditTexts
    String prefString = sharedPreferences.getString(userNameSP, "");
    String prefString1 = sharedPreferences.getString(userPasswordSP, "");
    txtUsrNm.setText(prefString);
    txtUsrPass.setText(prefString1);
}

@Override
protected void onDestroy() {
    // Clear all SharedPreferences data
    SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.clear();
    editor.apply();
    super.onDestroy();
}
}
```

FileHandling.java

```
package com.example.sharedpredemo;

import android.annotation.SuppressLint;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import static com.example.sharedpredemo.MainActivity.savePref;
import static com.example.sharedpredemo.MainActivity.userNameSP;

import androidx.appcompat.app.AppCompatActivity;

public class FileHandling extends AppCompatActivity{
    SharedPreferences sharedPreferences;
    String name;
    @Override
    protected void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_file_handling);

        // Initialize sharedPreferences once
        sharedPreferences = getSharedPreferences(savePref, MODE_PRIVATE);

        // Get saved SharedPreferences data
        getSharedRef();

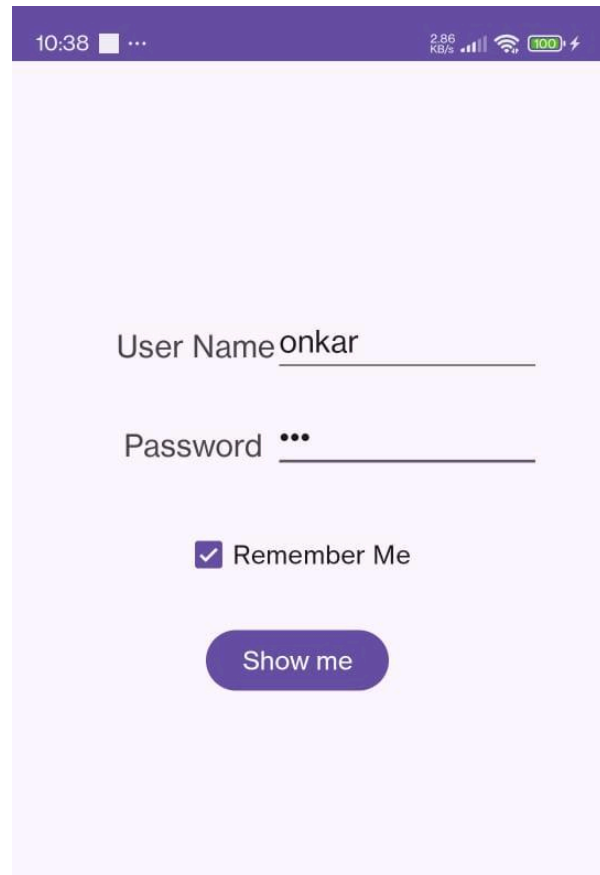
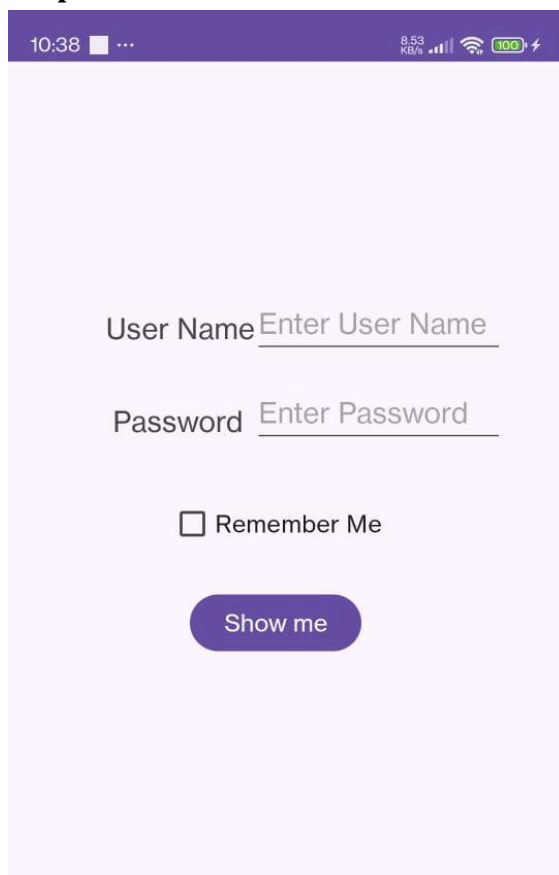
        // Set up the clear button to clear shared preferences data
        Button btnClear = findViewById(R.id.clearIt);
        btnClear.setOnClickListener(view -> clearSharedRef());
    }

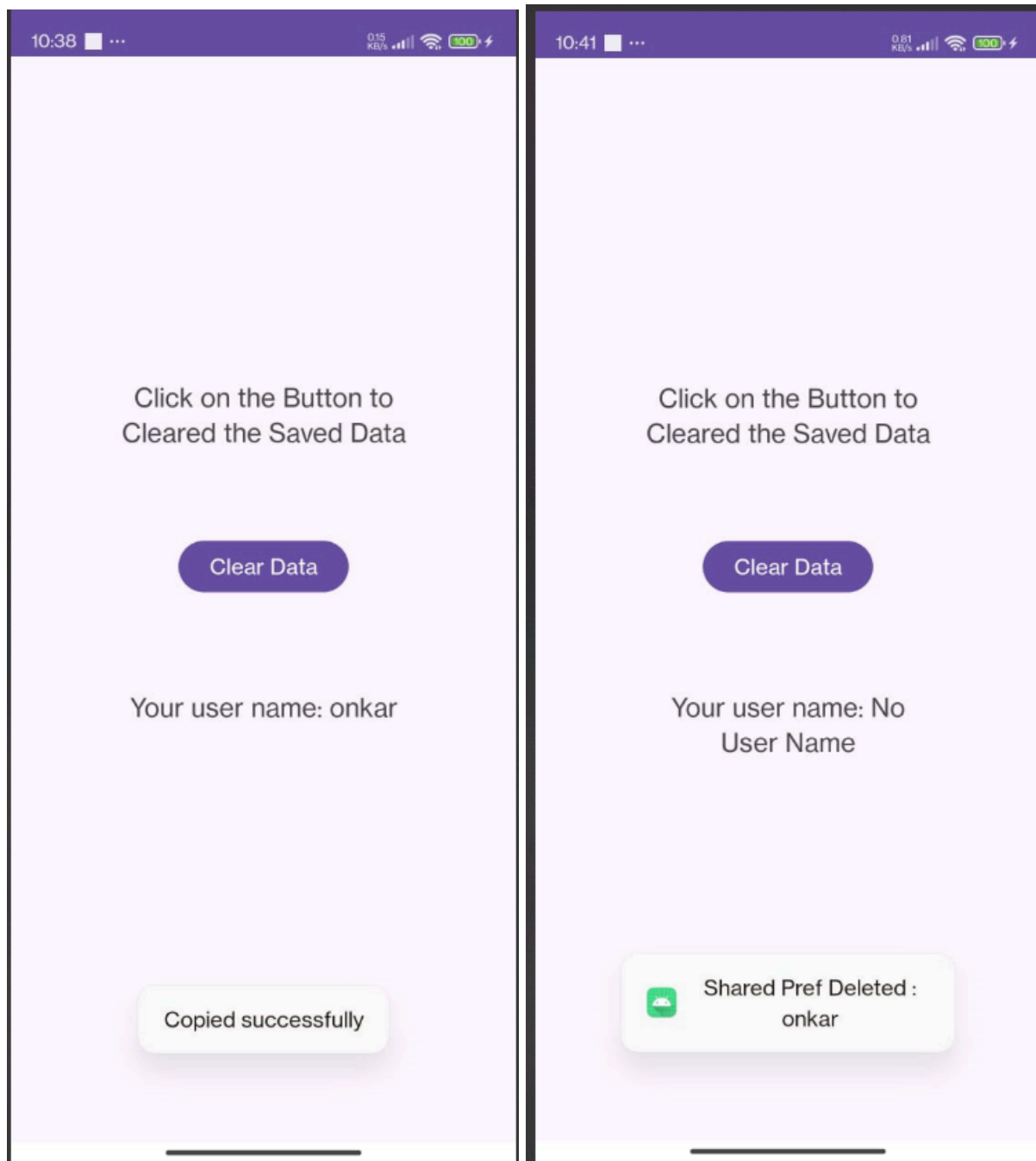
    @SuppressWarnings("SetTextI18n")
    public void getSharedRef(){
        TextView txtVwInfo = findViewById(R.id.clrDataVw);
        // Return No User Name if Remember not Click
        name = sharedPreferences.getString(userNameSP, "No User Name");
        txtVwInfo.setText("Your user name: " + name);
    }
}
```

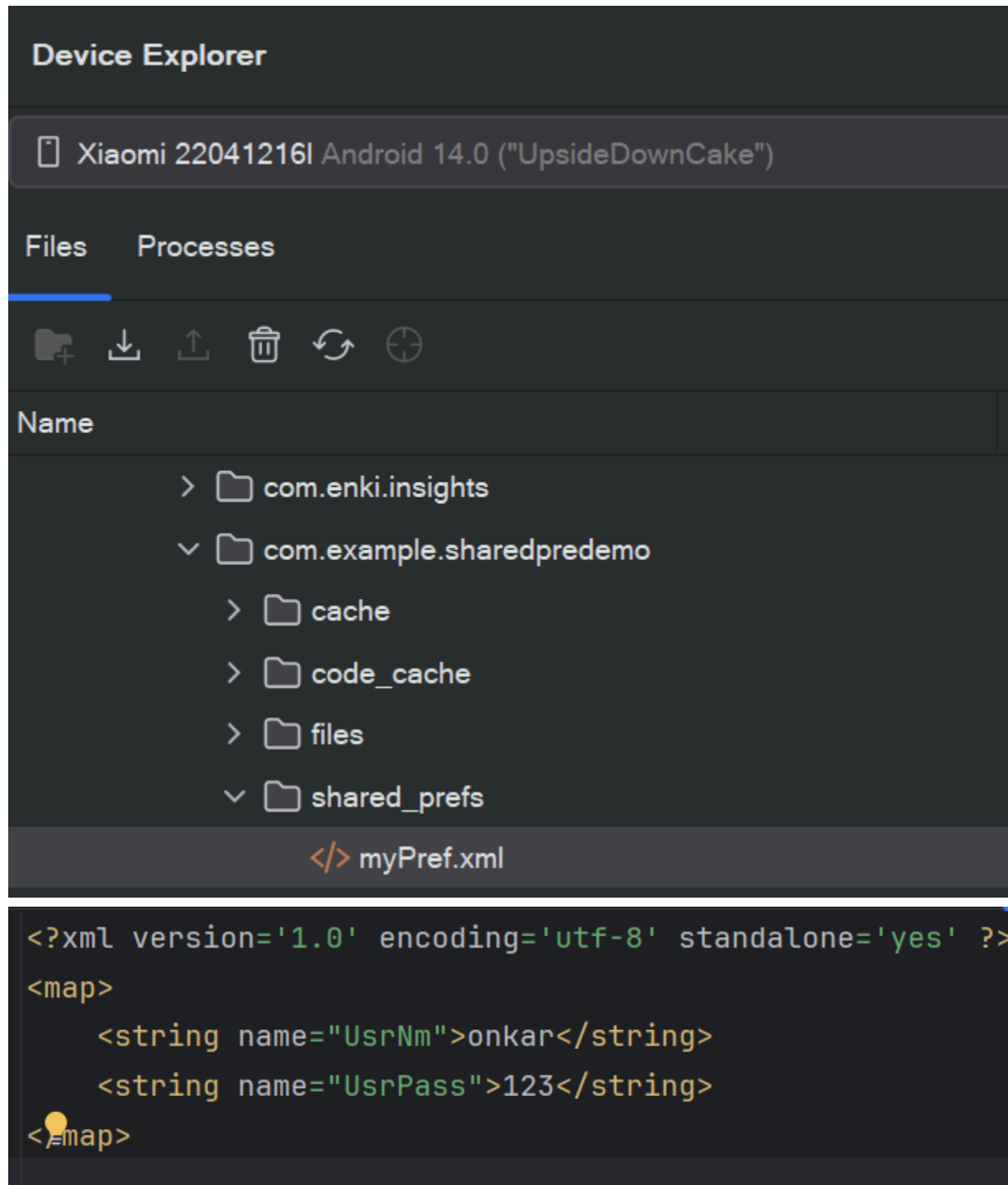
```
public void clearSharedRef(){
    // Clear shared preferences data
    SharedPreferences.Editor editor = sharedPreferences.edit().clear();
    editor.apply(); // Use apply() for asynchronous operation

    // Provide feedback to the user
    Toast.makeText(this, "Shared Pref Deleted : " + name, Toast.LENGTH_SHORT).show();

    // Refresh the displayed data
    getSharedRef();
}
}
```

Output:





Program:**activity_main.xml**

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

<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/txtView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:fontFamily="sans-serif"
        android:text=""
        android:textSize="18sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.734" />
```

```
<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="sans-serif"
    android:text="@string/directory_name"
    android:textSize="18sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.146"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.141" />
```

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="sans-serif"
    android:text="@string/file_name"
    android:textSize="18sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
```

```
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.127"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.253" />
```

<TextView

```
android:id="@+id/textView3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:fontFamily="sans-serif"
android:text="@string/content_of_file"
android:textSize="18sp"
android:textStyle="bold"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.143"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.393" />
```

<Button

```
android:id="@+id/btnRead"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
```



```
android:fontFamily="sans-serif"
android:text="@string/read_file"
android:textSize="18sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.201"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.579" />
```

<Button

```
android:id="@+id/btnDelete"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:fontFamily="sans-serif"
android:text="@string/delete_file"
android:textSize="18sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.792"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.579" />
```

<Button

```
android:id="@+id/btnWrite"
android:layout_width="220dp"
android:layout_height="51dp"
android:fontFamily="sans-serif"
android:text="@string/write_in_file"
android:textSize="18sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.497"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.673" />
```

<EditText

```
android:id="@+id/edTxtDir"
android:layout_width="295dp"
android:layout_height="48dp"
android:layout_marginTop="4dp"
android:autofillHints="address"
android:ems="10"
android:hint="@string/enter_directory_name"
android:inputType="text"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.353"
app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toBottomOf="@+id/textView2" />
```

```
<EditText
```

```
    android:id="@+id/edTxtFname"
    android:layout_width="295dp"
    android:layout_height="48dp"
    android:layout_marginTop="4dp"
    android:autofillHints="name"
    android:ems="10"
    android:hint="@string/enter_file_name"
    android:inputType="text"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.353"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView" />
```

```
<EditText
```

```
    android:id="@+id/edTxtContent"
    android:layout_width="313dp"
    android:layout_height="48dp"
    android:layout_marginTop="4dp"
    android:autofillHints="data"
    android:ems="10"
    android:hint="@string/enter_content_here"
    android:inputType="text"
```

```
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.428"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView3" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.internalexternalfiledemo;

import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.nio.file.Files;

public class MainActivity extends AppCompatActivity {
    EditText txtDir, txtFname, txtCntnt;

    @Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable(this);
    setContentView(R.layout.activity_main);
    txtDir = findViewById(R.id.edTxtDir);
    txtFname = findViewById(R.id.edTxtFname);
    txtCntnt = findViewById(R.id.edTxtContent);
    Button btnRead = findViewById(R.id.btnRead);
    Button btnWrite = findViewById(R.id.btnWrite);
    Button btnDelete = findViewById(R.id.btnDelete);
    TextView txtData = findViewById(R.id.txtView);
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) ->
{
    Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
    v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
    return insets;
});
    btnRead.setOnClickListener(view -> {
        try {
            StringBuilder result = new StringBuilder();
            String line;

            String folder = getApplication().getFilesDir().getAbsolutePath() + File.separator +
txtDir.getText().toString();

            File subFolder = new File(folder);

            if (txtDir.getText().toString().isEmpty() || txtFname.getText().toString().isEmpty()){
```

```
        Toast.makeText(getApplicationContext(), "Please Enter Folder / File Name",
Toast.LENGTH_SHORT).show();
    }
    else {
        BufferedReader bufferedReader = new BufferedReader(
            new FileReader(new File(folder, txtFname.getText().toString()))
        );
        while ((line = bufferedReader.readLine()) != null) {
            result.append(line);
        }
        txtData.setText(result.toString());
    }
} catch (Exception e){
    Toast.makeText(getApplicationContext(), e.getMessage(),
Toast.LENGTH_LONG).show();
    txtData.setText(null);
}
});
btnWrite.setOnClickListener(view -> {
    try{
        if (txtDir.getText().toString().isEmpty() || txtFname.getText().toString().isEmpty()){
            Toast.makeText(getApplicationContext(), "Please Enter Folder / File Name",
Toast.LENGTH_SHORT).show();
        }
        else {
```

```
String folder = getApplication().getFilesDir().getAbsolutePath() + File.separator
+ txtDir.getText().toString();

File subFolder = new File(folder);
if (!subFolder.exists()) {
    subFolder.mkdirs();
}

FileOutputStream outputStream = new FileOutputStream(new File(subFolder,
txtFname.getText().toString()));
outputStream.write(txtCntnt.getText().toString().getBytes());
outputStream.close();

Toast.makeText(getApplicationContext(), "Writing Complete",
Toast.LENGTH_SHORT).show();

txtCntnt.setText(null);
txtData.setText(null);
txtDir.setText(null);
txtFname.setText(null);
}
}catch(Exception e){

    Toast.makeText(getApplicationContext(), e.getMessage(),
Toast.LENGTH_SHORT).show();

    txtData.setText(null);
}
});

btnDelete.setOnClickListener( view -> {
    try {
```

```
String folder = getApplication().getFilesDir().getAbsolutePath() + File.separator +
txtDir.getText().toString();

File subFolder = new File(folder);

File file = new File(folder, txtFname.getText().toString());

if (txtDir.getText().toString().isEmpty() || txtFname.getText().toString().isEmpty()){

    Toast.makeText(getApplicationContext(), "Please Enter Folder / File Name",
Toast.LENGTH_SHORT).show();

}

else {

    if (file.exists()) {

        file.delete();

        Toast.makeText(getApplicationContext(), "Deletion Successful",
Toast.LENGTH_SHORT).show();

        txtData.setText(null);

        txtDir.setText(null);

        txtFname.setText(null);

        txtCntnt.setText(null);

    } else {

        Toast.makeText(getApplicationContext(), "File Not Exists",
Toast.LENGTH_SHORT).show();

        txtData.setText(null);

    }

}

} catch (Exception e){

    Toast.makeText(getApplicationContext(), e.getMessage(),
Toast.LENGTH_SHORT).show();
```



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

Output:

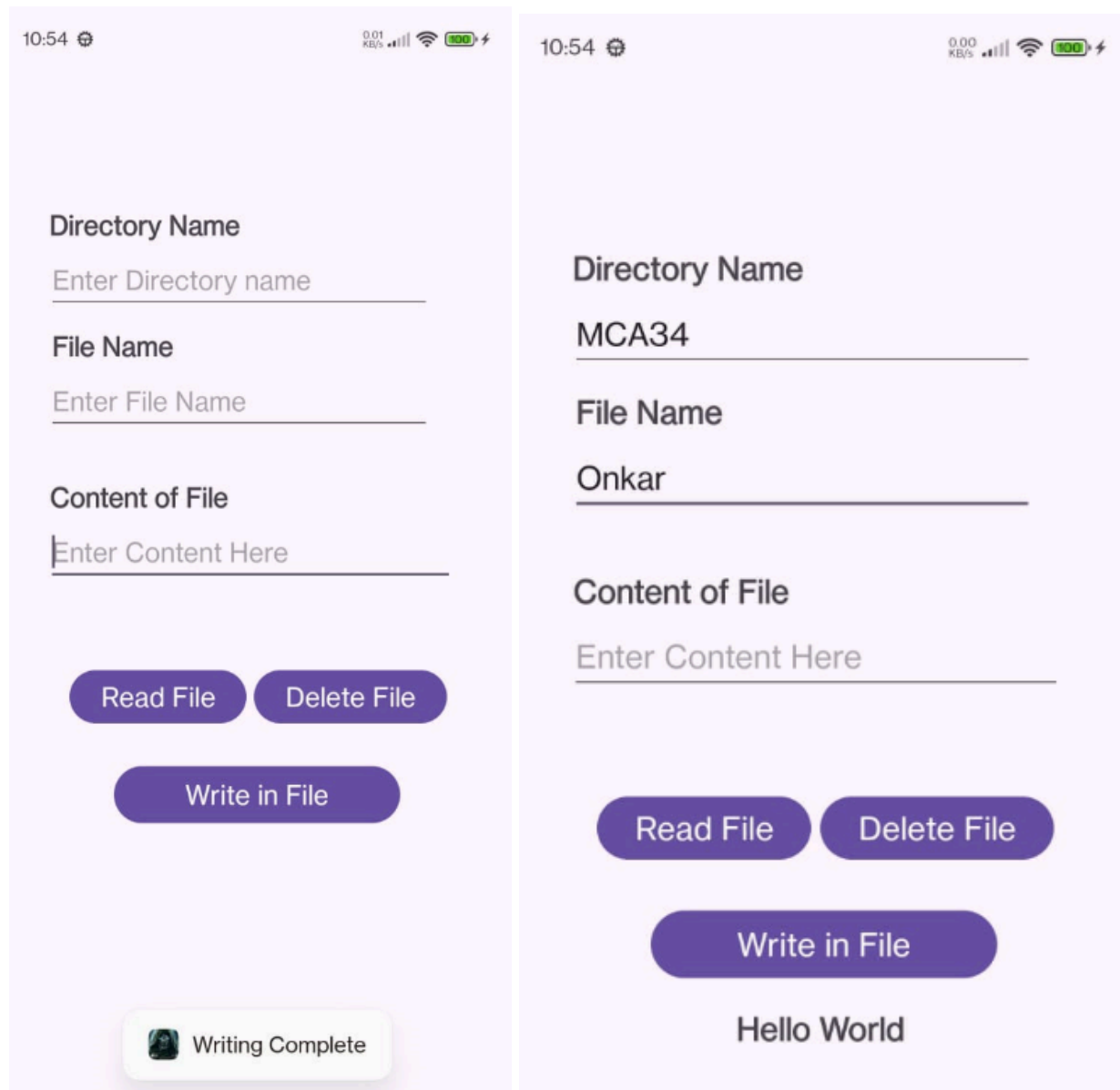
The image displays two side-by-side screenshots of a mobile application interface, likely for file management. Both screens show a light purple background with a status bar at the top indicating time, signal strength, and battery level.

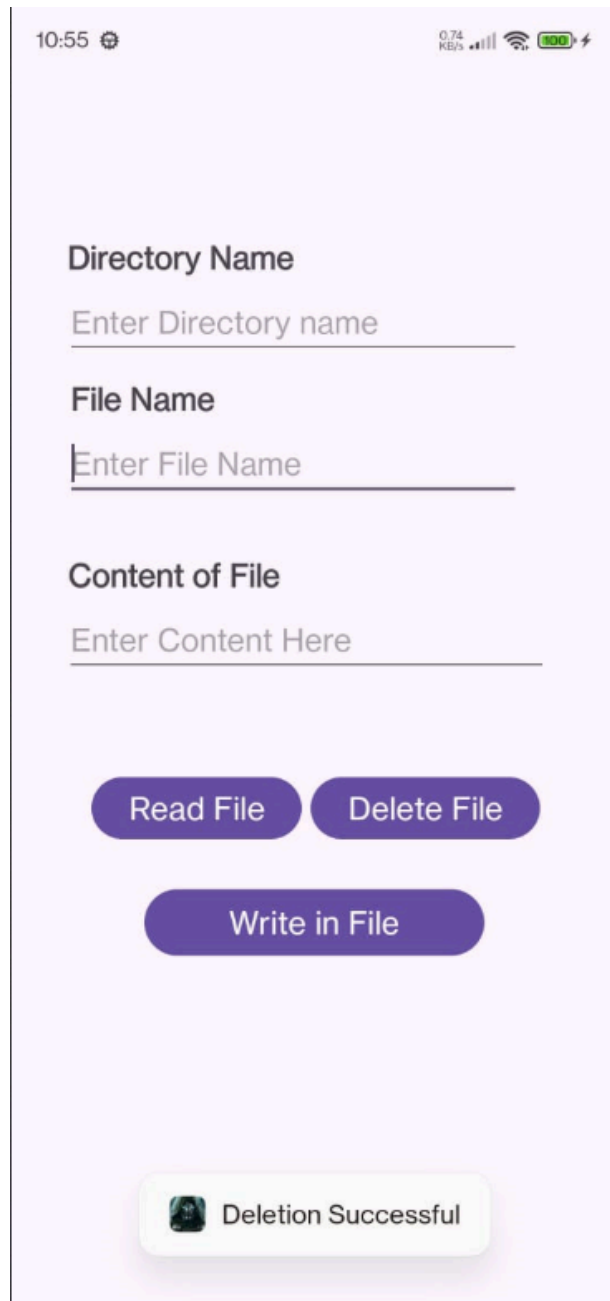
Left Screenshot (10:52):

- Directory Name:** Input field with placeholder text "Enter Directory name".
- File Name:** Input field with placeholder text "Enter File Name".
- Content of File:** Input field with placeholder text "Enter Content Here".
- Buttons:** Three purple buttons at the bottom: "Read File", "Delete File", and "Write in File".

Right Screenshot (10:54):

- Directory Name:** Input field containing the text "MCA34".
- File Name:** Input field containing the text "Onkar".
- Content of File:** Input field containing the text "Hello World".
- Buttons:** Three purple buttons at the bottom: "Read File", "Delete File", and "Write in File".





The screenshot shows a mobile application interface with a light purple background. At the top, the status bar displays the time 10:55, a gear icon, and network/battery indicators. The app interface includes three text input fields: 'Directory Name' with the placeholder 'Enter Directory name', 'File Name' with the placeholder 'Enter File Name', and 'Content of File' with the placeholder 'Enter Content Here'. Below these fields are three purple buttons: 'Read File', 'Delete File', and 'Write in File'. At the bottom, a green toast message with a trash icon and the text 'Deletion Successful' is visible.

Conclusion -

Here we learn to demonstrate application of Shared Preferences, Internal and External Storage.