Android code of External Storage

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:padding="16dp">

    <!-- EditText for user input -->
    <EditText
        android:id="@+id/editText"
        android:layout_width="0dp"
        android:layout_height="48dp"
        android:layout_marginTop="32dp"
        android:background="@drawable/edit_text_bg"
        android:ems="10"
        android:hint="Enter Text"
        android:inputType="textPersonName"
        android:textColor="@android:color/black"
        android:textColorHint="@android:color/darker_gray"
        android:textSize="16sp"
        android:paddingStart="16dp"
        android:paddingEnd="16dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <!-- LinearLayout containing the buttons -->
    <LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="24dp"
        android:weightSum="3"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText">

        <!-- Save Button -->
        <Button
            android:id="@+id/button_save"
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:layout_marginEnd="8dp"
            android:text="Save"
            android:textColor="@android:color/white"
            android:background="@color/colorPrimary"
            android:padding="12dp"
```

```
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <!-- Read Button -->
    <Button
        android:id="@+id/button_read"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:text="Read"
        android:textColor="@android:color/white"
        android:background="@color/colorPrimary"
        android:padding="12dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <!-- List Dirs Button -->
    <Button
        android:id="@+id/button_list"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:text="List Dirs"
        android:textColor="@android:color/white"
        android:background="@color/colorPrimary"
        android:padding="12dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</LinearLayout>

<!-- TextView to display output -->
<TextView
    android:id="@+id/textView"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:text="Output will be shown here"
    android:textSize="16sp"
    android:textColor="@android:color/black"
    android:padding="16dp"
    android:background="@drawable/text_view_bg"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/linearLayout" />

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

edit_text_bg.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <!-- Background Color -->
    <solid android:color="#FFFFFF"/>

    <!-- Corner Radius for Rounded Edges -->
    <corners android:radius="8dp"/>

    <!-- Border (Stroke) -->
    <stroke android:color="#B0B0B0" android:width="1dp"/>

    <!-- Padding (optional, if you want inner space) -->
    <padding
        android:left="8dp"
        android:top="8dp"
        android:right="8dp"
        android:bottom="8dp"/>
</shape>
```

text_view_bg.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <!-- Background color for the TextView -->
    <solid android:color="#FFFFFF"/> <!-- White background, you can modify this color -->

    <!-- Rounded corners -->
    <corners android:radius="12dp"/> <!-- Adjust corner radius for desired roundness -->

    <!-- Border stroke (optional, gives the TextView a border) -->
    <stroke android:color="#C1C1C1" android:width="1dp"/> <!-- Light grey border -->

    <!-- Padding around the content (optional) -->
    <padding android:left="8dp" android:top="8dp" android:right="8dp" android:bottom="8dp"/>
</shape>
```

MainActivity.kt

```
import android.Manifest
import android.content.pm.PackageManager
import android.os.Bundle
import android.os.Environment
import android.util.Log
import android.view.View
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import java.io.BufferedReader
import java.io.File
import java.io.FileInputStream
import java.io.FileOutputStream
import java.io.IOException
import java.io.InputStreamReader

class MainActivity : AppCompatActivity() {

    private lateinit var editText: EditText
    private lateinit var textView: TextView
    private lateinit var saveButton: Button
    private lateinit var readButton: Button
    private lateinit var listButton: Button

    private val LOG_TAG = "ExternalStorageDemo"

    private val REQUEST_ID_READ_PERMISSION = 100
    private val REQUEST_ID_WRITE_PERMISSION = 200

    private val fileName = "note.txt"

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        editText = findViewById(R.id.editText)
        textView = findViewById(R.id.textView)

        saveButton = findViewById(R.id.button_save)
        readButton = findViewById(R.id.button_read)
        listButton = findViewById(R.id.button_list)

        saveButton.setOnClickListener {
            askPermissionAndWriteFile()
        }

        readButton.setOnClickListener {
            askPermissionAndReadFile()
```

```
        }

        listButton.setOnClickListener {
            listExternalStorages()
        }
    }

    private fun askPermissionAndWriteFile() {
        val canWrite = askPermission(REQUEST_ID_WRITE_PERMISSION,
Manifest.permission.WRITE_EXTERNAL_STORAGE)
        if (!canWrite) {
            Toast.makeText(applicationContext, "You do not allow this app to write files.",
Toast.LENGTH_LONG).show()
            return
        }
        writeFile()
    }

    private fun askPermissionAndReadFile() {
        val canRead = askPermission(REQUEST_ID_READ_PERMISSION,
Manifest.permission.READ_EXTERNAL_STORAGE)
        if (!canRead) {
            Toast.makeText(applicationContext, "You do not allow this app to read files.",
Toast.LENGTH_LONG).show()
            return
        }
        readFile()
    }

    private fun askPermission(requestId: Int, permissionName: String): Boolean {
        Log.i(LOG_TAG, "Ask for Permission: $permissionName")
        Log.i(LOG_TAG, "Build.VERSION.SDK_INT: ${android.os.Build.VERSION.SDK_INT}")

        if (android.os.Build.VERSION.SDK_INT >= 23) {
            val permission = ActivityCompat.checkSelfPermission(this, permissionName)
            Log.i(LOG_TAG, "permission: $permission")
            Log.i(LOG_TAG, "PackageManager.PERMISSION_GRANTED:
${PackageManager.PERMISSION_GRANTED}")

            if (permission != PackageManager.PERMISSION_GRANTED) {
                requestPermissions(arrayOf(permissionName), requestId)
                return false
            }
        }
        return true
    }

    override fun onRequestPermissionsResult(requestCode: Int, permissions: Array<out String>,
grantResults: IntArray) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults)
        if (grantResults.isNotEmpty()) {
            when (requestCode) {
                REQUEST_ID_READ_PERMISSION -> {
                    if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                        readFile()
```

```
                }
            }
            REQUEST_ID_WRITE_PERMISSION -> {
                if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                    writeFile()
                }
            }
        }
    } else {
        Toast.makeText(applicationContext, "Permission Cancelled!",
Toast.LENGTH_SHORT).show()
    }
}

private fun getAppExternalFilesDir(): File {
    return if (android.os.Build.VERSION.SDK_INT >= 29) {
        getExternalFilesDir(null)!!
    } else {
        Environment.getExternalStorageDirectory()
    }
}

private fun writeFile() {
    try {
        val extStore = getAppExternalFilesDir()

        val canWrite = extStore.canWrite()
        Log.i(LOG_TAG, "Can write: ${extStore.absolutePath} : $canWrite")

        val path = "${extStore.absolutePath}/$fileName"
        Log.i(LOG_TAG, "Save to: $path")

        val data = editText.text.toString()
        Log.i(LOG_TAG, "Data: $data")

        val myFile = File(path)
        val fOut = FileOutputStream(myFile)
        fOut.write(data.toByteArray(Charsets.UTF_8))
        fOut.close()

        Toast.makeText(applicationContext, "$fileName saved", Toast.LENGTH_LONG).show()
    } catch (e: Exception) {
        Toast.makeText(applicationContext, "Write Error: ${e.message}",
Toast.LENGTH_LONG).show()
        Log.e(LOG_TAG, "Write Error: ${e.message}")
        e.printStackTrace()
    }
}

private fun readFile() {
    val extStore = getAppExternalFilesDir()

    val path = "${extStore.absolutePath}/$fileName"
    Log.i(LOG_TAG, "Read file: $path")
```

```
        var fileContent = ""
        try {
            val myFile = File(path)
            val fIn = FileInputStream(myFile)
            val myReader = BufferedReader(InputStreamReader(fIn))

            var s: String?
            while (myReader.readLine().also { s = it } != null) {
                fileContent += "$s\n"
            }
            myReader.close()

            textView.text = fileContent
        } catch (e: IOException) {
            Toast.makeText(applicationContext, "Read Error: ${e.message}",
Toast.LENGTH_LONG).show()
            Log.e(LOG_TAG, "Read Error: ${e.message}")
            e.printStackTrace()
        }
        Toast.makeText(applicationContext, fileContent, Toast.LENGTH_LONG).show()
    }

    private fun listExternalStorages() {
        val sb = StringBuilder()

        sb.append("Data Directory: \n - ${Environment.getDataDirectory()}\n")
        sb.append("Download Cache Directory: \n -
${Environment.getDownloadCacheDirectory()}\n")
        sb.append("External Storage State: \n - ${Environment.getExternalStorageState()}\n")
        sb.append("External Storage Directory: \n -
${Environment.getExternalStorageDirectory()}\n")
        sb.append("Is External Storage Emulated?: \n -
${Environment.isExternalStorageEmulated()}\n")
        sb.append("Is External Storage Removable?: \n -
${Environment.isExternalStorageRemovable()}\n")
        sb.append("External Storage Public Directory (Music): \n -
${Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_MUSIC)}\n")
        sb.append("Root Directory: \n - ${Environment.getRootDirectory()}\n")

        Log.i(LOG_TAG, sb.toString())
        textView.text = sb.toString()
    }
}
```

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.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.READ_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:supportsRtl="true"
        android:theme="@style/Theme.ExternalStorage"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

colors.xml

```
<resources>
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="colorPrimary">#6200EE</color>
    <color name="colorPrimaryDark">#3700B3</color>
    <color name="colorAccent">#03DAC5</color>
```

In the mentioned path is the code when you run the code and create write something and save it.

/sdcard/Android/data/com.example.externalstorage/files/note.txt