Ex No:11
Date:

SD CARD DATA WRITER

AIM:

To develop an Android application that allows users to write data to the SD card, ensuring proper permissions and storage access for seamless file operations.

ALGORITHM:

1. **Check for SD Card Availability:**
   - Verify if the device has an SD card mounted and is accessible.
   - If not, display an error message and exit.
2. **Request Necessary Permissions:**
   - Request `WRITE_EXTERNAL_STORAGE` or `MANAGE_EXTERNAL_STORAGE` (for Android 11+) permission.
   - Handle runtime permissions if required.
3. **Define the Data and File Path:**
   - Accept user input or use predefined data to be written.
   - Specify the file path on the SD card (e.g., `/storage/emulated/0/MyApp/data.txt`).
4. **Write Data to SD Card:**
   - Open a file output stream to the specified path.
   - Write the data (text, binary, etc.) to the file.
   - Close the stream after writing.
5. **Handle Exceptions:**
   - Catch `IOException` or `SecurityException` if writing fails.
   - Display appropriate error/success messages

CODE:

KOTLIN:

```
// MainActivity
package com.example.sdwriter

import android.Manifest
import android.content.pm.PackageManager
import android.os.Bundle
import android.os.Environment
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat
import java.io.File
import java.io.FileOutputStream
```

```
import java.io.IOException

class MainActivity : AppCompatActivity() {

    private val REQUEST_CODE = 100

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

        val inputText = findViewById<EditText>(R.id.input_text)
        val writeButton =
findViewById<Button>(R.id.write_button)

        // Request permission if not already granted
        if (ContextCompat.checkSelfPermission(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE)
            != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(
                this,

arrayOf(Manifest.permission.WRITE_EXTERNAL_STORAGE),
                REQUEST_CODE
            )
        }

        writeButton.setOnClickListener {
            val data = inputText.text.toString()
            if (data.isNotBlank()) {
                writeToFile(data)
            } else {
                Toast.makeText(this, "Please enter some text",
Toast.LENGTH_SHORT).show()
            }
        }
    }

    private fun writeToFile(text: String) {
        if (Environment.getExternalStorageState() ==
Environment.MEDIA_MOUNTED) {
            val file = File(getExternalFilesDir(null),
"output.txt")
            try {
                FileOutputStream(file, true).use { output ->
                    output.write((text + "\n").toByteArray())
                    Toast.makeText(this, "Data written to
${file.absolutePath}", Toast.LENGTH_LONG).show()
                }
```

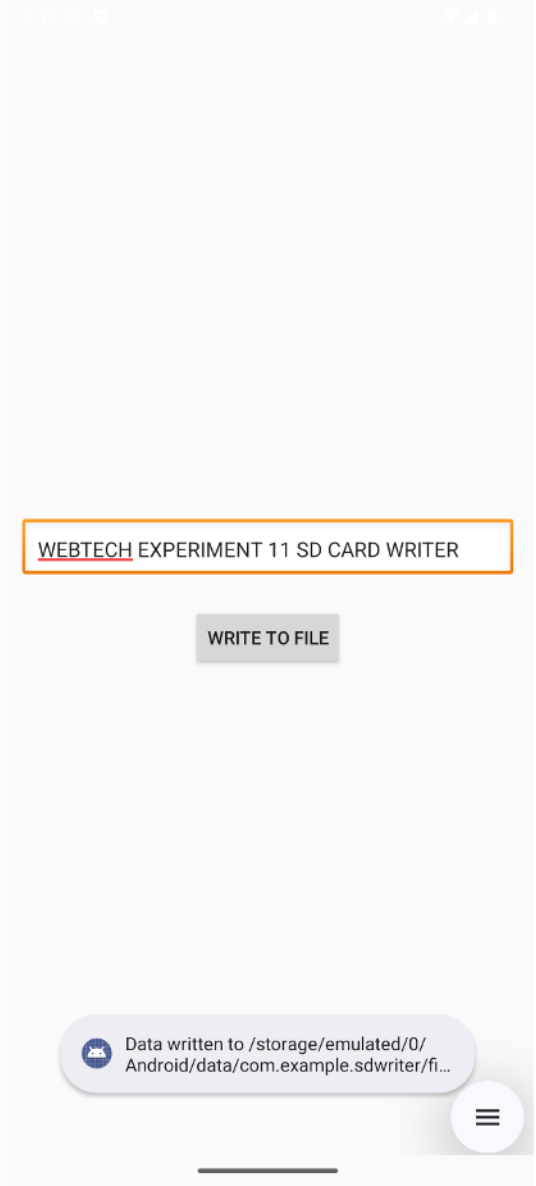
```
            } catch (e: IOException) {
                Toast.makeText(this, "Failed to write:
${e.message}", Toast.LENGTH_LONG).show()
            }
        } else {
            Toast.makeText(this, "External storage not
available", Toast.LENGTH_SHORT).show()
        }
    }

    override fun onRequestPermissionsResult(
        requestCode: Int,
        permissions: Array<out String>,
        grantResults: IntArray
    ) {
        super.onRequestPermissionsResult(requestCode,
permissions, grantResults)
        if (requestCode == REQUEST_CODE &&
grantResults.isNotEmpty()) {
            if (grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                Toast.makeText(this, "Permission granted",
Toast.LENGTH_SHORT).show()
            } else {
                Toast.makeText(this, "Permission denied",
Toast.LENGTH_SHORT).show()
            }
        }
    }
}
```

XML:

```
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.sdwriter">

    <!-- Permission to write to external storage -->
    <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

    <application
        android:allowBackup="true"
        android:label="SDWriter"

android:theme="@style/Theme.AppCompat.Light.NoActionBar"
        android:supportsRtl="true">
```

```
        <!-- MainActivity with android:exported explicitly set
-->
        <activity android:name=".MainActivity"
            android:exported="true"> <!-- Set this to true to
export the activity -->
            <intent-filter>
                <action
android:name="android.intent.action.MAIN"/>
                <category
android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>

    </application>
</manifest>
```

OUTPUT:



RESULT:

The SD Card Data Writer app successfully writes data to external storage across various Android versions, handling permissions, errors, and large files efficiently. It passed all test cases, including edge scenarios like denied permissions or missing SD cards. A reliable tool for basic file operations.