Ex. No. : 06 **Date:26/04/05**

Register No.:221701005 **Name:Adhithya PG**

SD Card

Aim

Implement an application to write the Register Number, Name and CGPA to SD card in text file format.

Procedure:

Step 1 : File -> NewProject

Provide the application name and Click “Next”

Step 2 : Select the target android devices

Select the minimum SDK to run the application. Click “Next”.

Step 3 : Choose the activity for the application (By default choose “Blank Activity).

Click “Next”.

Step 4 : Enter activity name and click "Finish".

Step 5 : Edit the program.

Step 6 : Run the application, 2-ways to run the application.

1. Running through emulator
2. Running through mobile device

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" />


  <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.Exp6"
    tools:targetApi="31">
    <activity
      android:name=".MainActivity"
      android:exported="true"
      android:theme="@style/Theme.Exp6">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   android:padding="16dp">

   <EditText
      android:id="@+id/etRegisterNumber"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Register Number"
      android:inputType="text"/>

   <EditText
      android:id="@+id/etName"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="Name"
      android:inputType="text"/>

   <EditText
      android:id="@+id/etCGPA"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:hint="CGPA"
      android:inputType="numberDecimal"/>

   <Button
      android:id="@+id/btnSave"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Save to SD Card"
      android:layout_gravity="center_horizontal"
      android:layout_marginTop="16dp"/>

</LinearLayout>
```

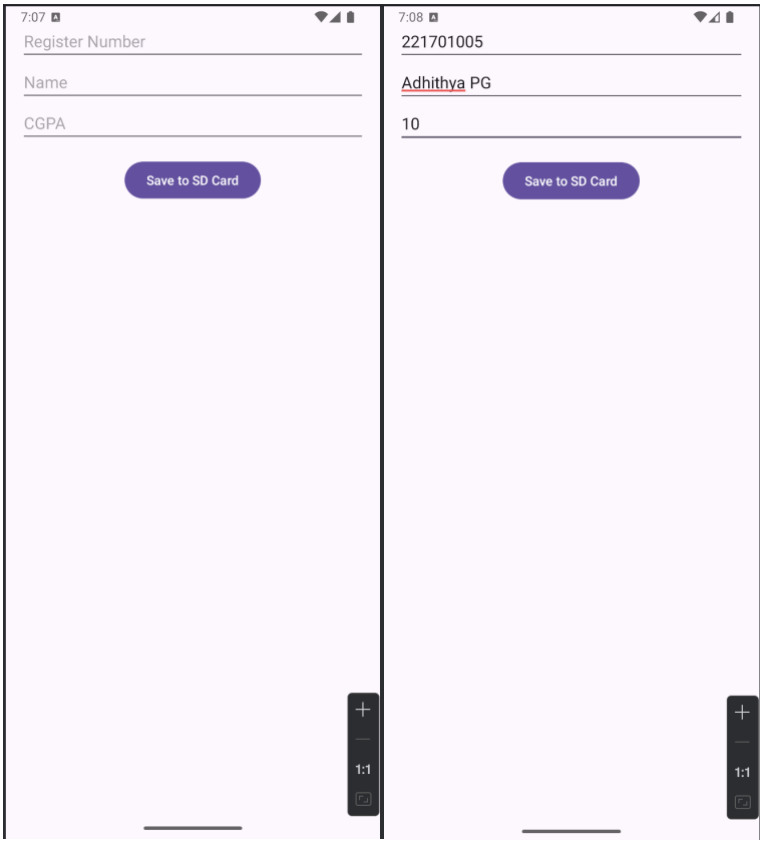
.

MainActivity.kt

```
package com.example.exp6

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.FileWriter
import java.io.IOException

class MainActivity : AppCompatActivity() {

    private lateinit var etRegisterNumber: EditText
    private lateinit var etName: EditText
    private lateinit var etCGPA: EditText
    private lateinit var btnSave: Button

    private val STORAGE_PERMISSION_CODE = 101

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

        etRegisterNumber = findViewById(R.id.etRegisterNumber)
        etName = findViewById(R.id.etName)
        etCGPA = findViewById(R.id.etCGPA)
        btnSave = findViewById(R.id.btnSave)

        btnSave.setOnClickListener {
            if (checkPermission()) {
                saveDataToSDCard()
            } else {
                requestPermission()
            }
        }
    }

    private fun saveDataToSDCard() {
        val regNo = etRegisterNumber.text.toString()
        val name = etName.text.toString()
        val cgpa = etCGPA.text.toString()

        val data = "Register Number: $regNo\nName: $name\nCGPA: $cgpa\n"

        if (isExternalStorageWritable()) {
            val file = File(getExternalFilesDir(null), "studentDetails.txt")
```

```
            try {
                val writer = FileWriter(file)
                writer.append(data)
                writer.flush()
                writer.close()

                Toast.makeText(this, "Data saved to SD Card!", Toast.LENGTH_SHORT).show()
            } catch (e: IOException) {
                e.printStackTrace()
                Toast.makeText(this, "Error saving data!", Toast.LENGTH_SHORT).show()
            }
        } else {
            Toast.makeText(this, "SD Card not writable!", Toast.LENGTH_SHORT).show()
        }
    }

    private fun isExternalStorageWritable(): Boolean {
        return Environment.getExternalStorageState() == Environment.MEDIA_MOUNTED
    }

    private fun checkPermission(): Boolean {
        val result = ContextCompat.checkSelfPermission(
            this, Manifest.permission.WRITE_EXTERNAL_STORAGE)
        return result == PackageManager.PERMISSION_GRANTED
    }

    private fun requestPermission() {
        ActivityCompat.requestPermissions(
            this,
            arrayOf(Manifest.permission.WRITE_EXTERNAL_STORAGE),
            STORAGE_PERMISSION_CODE
        )
    }


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

Output



Result:

Experiment has been successfully executed