Ex. No. : 06 **Date :**

Register No. : 221701032 **Name : LINGESH VK**

SD Card

Aim:

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

Procedure:

Step 1 : File → New Project

Provide the application name (e.g.,"SD Card") 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

Request storage permissions in the manifest file.

Use file input/output streams in MainActivity.kt to read from and write to

SD card or internal storage.

Step 6 : Run the application

Two ways to run the application:

1. Running through emulator (with SD card configured)

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">
    <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.SdCard"
        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>
```

Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
    android:text="SD Card"
    android:textSize="24sp"
    android:textStyle="bold"
    android:textColor="@android:color/white"
    android:background="#6200EE"
    android:padding="12dp"/>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:layout_marginTop="16dp">
    <EditText
        android:id="@+id/editTextRegisterNumber"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter the register number..."
        android:inputType="text"
        android:padding="12dp"
        android:background="@drawable/edit_text_background"
        android:layout_marginBottom="8dp"/>
    <EditText
        android:id="@+id/editTextName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter the name..."
        android:inputType="textPersonName"
        android:padding="12dp"
        android:background="@drawable/edit_text_background"
        android:layout_marginBottom="8dp"/>
    <EditText
        android:id="@+id/editTextCGPA"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter the CGPA..."
        android:inputType="numberDecimal"
        android:padding="12dp"
```

```
        android:background="@drawable/edit_text_background"
        android:layout_marginBottom="16dp"/>
    <Button
        android:id="@+id/buttonSave"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Save"
        android:textColor="@android:color/white"
        android:backgroundTint="#6200EE"
        android:padding="12dp"
        android:layout_marginBottom="8dp"/>
    <Button
        android:id="@+id/buttonLoad"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Load"
        android:textColor="@android:color/white"
        android:backgroundTint="#6200EE"
        android:padding="12dp"/>
  </LinearLayout>
</LinearLayout>
```

MainActivity.kt:

```
package com.example.sdcard
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 lateinit var registerNumberEditText: EditText
  private lateinit var nameEditText: EditText
  private lateinit var cgpaEditText: EditText
  private lateinit var saveButton: Button
  private lateinit var loadButton: Button
  private val STORAGE_PERMISSION_CODE = 100
  override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_main)
     registerNumberEditText = findViewById(R.id.editTextRegisterNumber)
    nameEditText = findViewById(R.id.editTextName)
    cgpaEditText = findViewById(R.id.editTextCGPA)
    saveButton = findViewById(R.id.buttonSave)
    loadButton = findViewById(R.id.buttonLoad)
     requestStoragePermission()
    saveButton.setOnClickListener {
      saveDataToSD()
    }
    loadButton.setOnClickListener {
      loadDataFromSD()
    }
  }
  private fun requestStoragePermission() {
    if (ContextCompat.checkSelfPermission(
        this,
        Manifest.permission.WRITE_EXTERNAL_STORAGE
```

```
        ) != PackageManager.PERMISSION_GRANTED
    ) {
        ActivityCompat.requestPermissions(
            this,
            arrayOf(
                Manifest.permission.WRITE_EXTERNAL_STORAGE,
                Manifest.permission.READ_EXTERNAL_STORAGE
            ),
            STORAGE_PERMISSION_CODE
        )
    }
}
private fun saveDataToSD() {
    val registerNumber = registerNumberEditText.text.toString().trim()
    val name = nameEditText.text.toString().trim()
    val cgpa = cgpaEditText.text.toString().trim()
    if (registerNumber.isEmpty() || name.isEmpty() || cgpa.isEmpty()) {
        Toast.makeText(this, "Please fill all fields", Toast.LENGTH_SHORT).show()
        return
    }
    try {
        val file = try {
            val appDir = getExternalFilesDir(null)
            val dir = File(appDir, "SDCardApp")
            if (!dir.exists()) {
                dir.mkdirs()
            }
            File(dir, "$registerNumber.txt")
        } catch (e: Exception) {
            val dir = File(Environment.getExternalStorageDirectory(), "SDCardApp")
            if (!dir.exists()) {
```

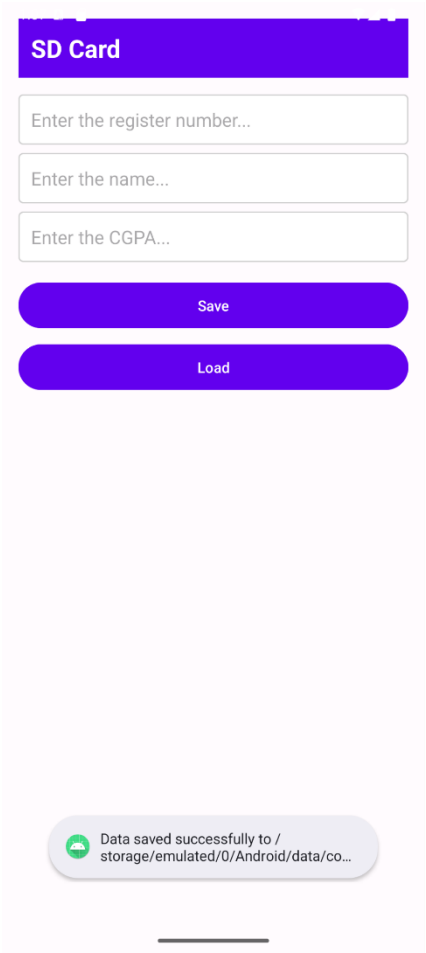
```
            dir.mkdirs()
        }
        File(dir, "$registerNumber.txt")
    }
    val fileOutputStream = FileOutputStream(file)
    val data = "Register Number: $registerNumber\nName: $name\nCGPA: $cgpa"
    fileOutputStream.write(data.toByteArray())
    fileOutputStream.close()
    Toast.makeText(this, "Data saved successfully to ${file.absolutePath}", Toast.LENGTH_LONG).show()
    registerNumberEditText.text.clear()
    nameEditText.text.clear()
    cgpaEditText.text.clear()
  } catch (e: IOException) {
    e.printStackTrace()
    Toast.makeText(this, "Error saving data: ${e.message}", Toast.LENGTH_SHORT).show()
  }
}
private fun loadDataFromSD() {
  val registerNumber = registerNumberEditText.text.toString().trim()
  if (registerNumber.isEmpty()) {
    Toast.makeText(this, "Please enter a register number", Toast.LENGTH_SHORT).show()
    return
  }
  try {
    val file = findDataFile(registerNumber)
    if (file == null || !file.exists()) {
      Toast.makeText(this, "No data found for this register number", Toast.LENGTH_SHORT).show()
      return
    }
    val fileContent = file.readText()
    val lines = fileContent.split("\n")
```

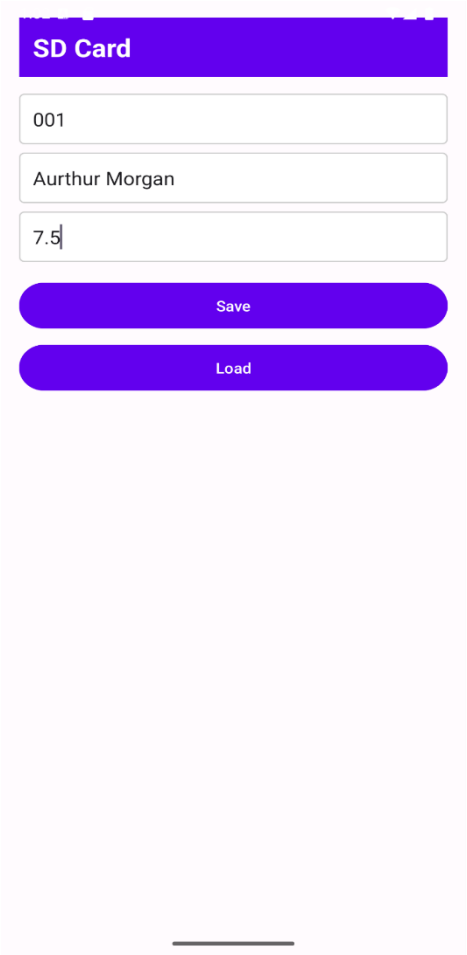
```
            for (line in lines) {
                when {
                    line.startsWith("Name:") -> {
                        val name = line.substring(line.indexOf(":") + 1).trim()
                        nameEditText.setText(name)
                    }
                    line.startsWith("CGPA:") -> {
                        val cgpa = line.substring(line.indexOf(":") + 1).trim()
                        cgpaEditText.setText(cgpa)
                    }
                }
            }
            Toast.makeText(this, "Data loaded successfully from ${file.absolutePath}",
Toast.LENGTH_SHORT).show()
        } catch (e: IOException) {
            e.printStackTrace()
            Toast.makeText(this, "Error loading data: ${e.message}", Toast.LENGTH_SHORT).show()
        }
    }
    private fun findDataFile(registerNumber: String): File? {
        val appDir = getExternalFilesDir(null)
        val appSpecificDir = File(appDir, "SDCardApp")
        var file = File(appSpecificDir, "$registerNumber.txt")
        if (file.exists()) {
            return file
        }
        try {
            val externalDir = File(Environment.getExternalStorageDirectory(), "SDCardApp")
            file = File(externalDir, "$registerNumber.txt")
            if (file.exists()) {
                return file
```

```
        }
    } catch (e: Exception) {
    }
    val cacheDir = File(cacheDir, "SDCardApp")
    if (!cacheDir.exists()) {
        cacheDir.mkdirs()
    }
    file = File(cacheDir, "$registerNumber.txt")
    if (file.exists()) {
        return file
    }
    return null
}
private fun isExternalStorageWritable(): Boolean {
    return Environment.getExternalStorageState() == Environment.MEDIA_MOUNTED
}
private fun isExternalStorageReadable(): Boolean {
    return Environment.getExternalStorageState() in
            setOf(Environment.MEDIA_MOUNTED, Environment.MEDIA_MOUNTED_READ_ONLY)
}
override fun onRequestPermissionsResult(
    requestCode: Int,
    permissions: Array<String>,
    grantResults: IntArray
) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults)
    if (requestCode == STORAGE_PERMISSION_CODE) {
        if (grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            Toast.makeText(this, "Storage permission granted", Toast.LENGTH_SHORT).show()
        } else {
            Toast.makeText(this, "Storage permission denied", Toast.LENGTH_SHORT).show()
```

```
    }
}
```

Output:





Result:

The SD Card Access application successfully reads and writes data to the SD card or internal storage when tested on a mobile device with appropriate permissions.