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="20dp">

    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter text to save" />

    <Button
        android:id="@+id/saveButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Save to SD Card" />
</LinearLayout>
```

Main java

```
package com.example.ja;

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.os.Environment;
import android.view.View;
import android.widget.Button;


import android.widget.EditText;
import android.widget.Toast;
import androidx.annotation.NonNull;
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;

public class MainActivity extends AppCompatActivity {
    private static final int STORAGE_PERMISSION_CODE = 101;
```

```
EditText editText;
Button saveButton;

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

  editText = findViewById(R.id.editText);
  saveButton = findViewById(R.id.saveButton);

  saveButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
      if (checkPermission()) {
        writeToSDCard(editText.getText().toString());
      } else {
        requestPermission();


      }
    }
  });
}

// Check if permission is granted
private boolean checkPermission() {
  int result = ContextCompat.checkSelfPermission(this,
      Manifest.permission.WRITE_EXTERNAL_STORAGE);
  return result == PackageManager.PERMISSION_GRANTED;
}

// Request permission
private void requestPermission() {
  ActivityCompat.requestPermissions(this, new
        String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE},
      STORAGE_PERMISSION_CODE);
}

// Handle permission result
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
                    @NonNull int[] grantResults) {
  super.onRequestPermissionsResult(requestCode, permissions, grantResults);
  if (requestCode == STORAGE_PERMISSION_CODE) {
    if (grantResults.length > 0 && grantResults[0] ==
        PackageManager.PERMISSION_GRANTED) {
      Toast.makeText(this, "Permission granted!", Toast.LENGTH_SHORT).show();
```

```
            } else {
                Toast.makeText(this, "Permission denied!", Toast.LENGTH_SHORT).show();
            }
        }


    }

    // Function to write data to SD Card
    private void writeToSDCard(String data) {
        if (Environment.getExternalStorageState().equals(Environment.MEDIA_MOUNTED)) {
            File sdCard = Environment.getExternalStorageDirectory();
            File dir = new File(sdCard.getAbsolutePath() + "/MyAppData");
            if (!dir.exists()) {
                dir.mkdirs();
            }

            File file = new File(dir, "sample.txt");
            try {
                FileOutputStream fos = new FileOutputStream(file);
                fos.write(data.getBytes());
                fos.close();
                Toast.makeText(this, "Data written to SD Card!", Toast.LENGTH_SHORT).show();
            } catch (IOException e) {
                e.printStackTrace();
                Toast.makeText(this, "Error writing file!", Toast.LENGTH_SHORT).show();
            }
        } else {
            Toast.makeText(this, "SD Card not available!", Toast.LENGTH_SHORT).show();
        }
    }
}
```