```
package com.example.sahilstorch;

import android.Manifest;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.hardware.camera2.CameraAccessException;
import android.hardware.camera2.CameraManager;
import android.os.Bundle;
import android.widget.Button;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;

public class MainActivity extends AppCompatActivity {
    private boolean isTorchOn = false;
    private CameraManager cameraManager;
    private String cameraId;

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

```
    // Set the background color to black
    findViewById(R.id.main_layout).setBackgroundColor(Color.BLACK);

    // Request Camera permission
    if (ContextCompat.checkSelfPermission(this, Manifest.permission.CAMERA)
            != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.CAMERA}, 50);
    } else {
        setupTorch();
    }

    Button torchButton = findViewById(R.id.torch_button);
    torchButton.setOnClickListener(v -> toggleTorch());
}

private void setupTorch() {
    cameraManager = (CameraManager) getSystemService(CAMERA_SERVICE);
    try {
        cameraId = cameraManager.getCameraIdList()[0];
    } catch (CameraAccessException e) {
        e.printStackTrace();
    }
}

private void toggleTorch() {
    if (cameraManager != null) {
```

```
        try {
            if (isTorchOn) {
                cameraManager.setTorchMode(cameraId, false);
                isTorchOn = false;
            } else {
                cameraManager.setTorchMode(cameraId, true);
                isTorchOn = true;
            }
        } catch (CameraAccessException e) {
            e.printStackTrace();
        }
    }
}

@Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
    if (requestCode == 50) {
                                        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            setupTorch();
        } else {
                        Toast.makeText(this, "Camera permission is required to use the torch",
Toast.LENGTH_SHORT).show();
        }
    }
```

```
    }

}
```