

PRACTICAL: 6

AIM: Create an application which turns ON or OFF Torch/Flashlight of Camera.

THEORY: About elements used:

ToggleButton: A toggle button allows the user to change a setting between two states. You can add a basic toggle button to your layout with the ToggleButton object. Android 4.0 (API level 14) introduces another kind of toggle button called a switch that provides a slider control, which you can add with a Switch object. SwitchCompat is a version of the Switch widget which runs on devices back to API 7. If you need to change a button's state yourself, you can use the CompoundButton.setChecked() or CompoundButton.toggle() method.

cameraManager: A system service manager for detecting, characterizing, and connecting to CameraDevice (public final class CameraManager extends Object).

CODE:

```
activity_main.xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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"
tools:context=".MainActivity">

    <ToggleButton
        android:id="@+id/togglebutton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textOff="FLASH OFF"
        android:textOn="FLASH ON"
        android:paddingHorizontal="20dp"
        android:paddingVertical="20dp"
        android:textSize="20dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.373" />

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="15dp"
        android:paddingVertical="20dp"
```

```

        android:paddingHorizontal="20dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/togglebutton"
        app:layout_constraintVertical_bias="0.269" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

MainActivity.java

```
package com.example.pr6_17it005;
```

```

import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Context;
import android.content.pm.PackageManager;
import android.hardware.camera2.CameraAccessException;
import android.hardware.camera2.CameraManager;
import android.os.Build;
import android.os.Bundle;
import android.widget.CompoundButton;
import android.widget.TextView;
import android.widget.Toast;
import android.widget.ToggleButton;

```

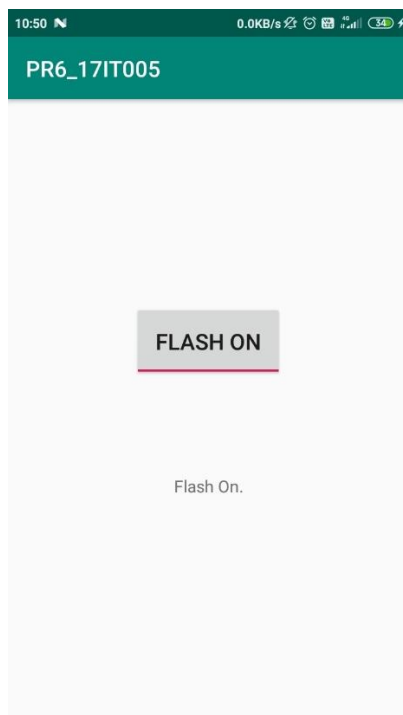
```

@RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
public class MainActivity extends AppCompatActivity {
    ToggleButton toggle;
    CameraManager cameraManager;
    TextView textView;
    String cameraID;
    @RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        if
(!getApplicationContext().getPackageManager().hasSystemFeature(PackageManager.FEATURE_CAMERA_FLASH)){
            Toast.makeText(getApplicationContext(),"No Flash",Toast.LENGTH_SHORT).show();
            System.exit(0);
        }
        toggle = findViewById(R.id.togglebutton);
        cameraManager = (CameraManager) getSystemService(Context.CAMERA_SERVICE);
        try {
            cameraID = cameraManager.getCameraIdList()[0];
        } catch (CameraAccessException e) {
        }
        textView = findViewById(R.id.textView);
        toggle.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
            @RequiresApi(api = Build.VERSION_CODES.M)
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
                try {

```

```
        cameraManager.setTorchMode(cameraID,isChecked);  
        if (isChecked){ textView.setText("Flash On."); }  
        else {textView.setText("Flash Off.");}  
    } catch (CameraAccessException e) { }  
    }  
});  
});  
}
```

OUTPUT:



Here, we have used toggle button and “flash on” on the screen indicated the flash led is on, and “flash off” indicates the flash light is off.

LATEST APPLICATIONS: Flash lights are used for notification alerts, like when you receive a call or message or any other notification, flash lights are used to draw attention to it.

LEARNING OUTCOME: Using cameraManager to turn camera flash light on and off and adding permission for the application to access hardware.