TRAFFIC MANAGEMENT SYSTEM

Using IoT

**TEAM MEMBERS:**

* **PRABU T (MENTOR)**
* **REVU BHARATH KUMAR (LEADER)**
* **THIYAGARAJAN D (MEMBER)**
* **SUDHAKAR V (MEMBER)**
* **MUSALI DIVAKAR (MEMBER)**
* **MEESALA PRAVEEN (MEMBER)**

# **OBJECTIVE :**

* **Traffic Flow Optimization**: Predict congestion to implement traffic flow optimization strategies, such as adjusting traffic signal timings, managing lane configurations, and implementing dynamic speed limits to alleviate congestion.
* **Dynamic Routing**: Provide real-time traffic congestion information to drivers through navigation apps and electronic road signs, enabling them to choose alternative routes and reduce congestion on heavily congested roads.
* **Public Transportation Integration**: Integrate congestion prediction with public transportation systems to optimize bus or train schedules and routing to reduce congestion and improve transit efficiency.
* **Incident Management**: Detect and predict congestion caused by accidents, road closures, or other incidents, allowing for quicker incident response and diversion of traffic.
* **Intersection Control**: Use congestion prediction to optimize traffic signal timings at intersections, reducing wait times and minimizing congestion buildup.

# **CIRCUIT DIAGRAM:**

# **COMPONENTS:**

1. Arduino UNO R3 CH340
2. LCD display
3. Ultrasonic Sensor HC-SR04
4. Male to Male Jumper Wires
5. Connecting wires

# **Code:**

import RPi.GPIO as GPIO

import time

echoPin = 12

trigPin = 13

GPIO.setmode(GPIO.BCM)

GPIO.setup(trigPin, GPIO.OUT)

GPIO.setup(echoPin, GPIO.IN)

def measure\_distance():

GPIO.output(trigPin, GPIO.LOW)

time.sleep(0.2)

GPIO.output(trigPin, GPIO.HIGH)

time.sleep(0.00001)

GPIO.output(trigPin, GPIO.LOW)

while GPIO.input(echoPin) == 0:

pulse\_start = time.time()

while GPIO.input(echoPin) == 1:

pulse\_end = time.time()

pulse\_duration = pulse\_end - pulse\_start

distance\_cm = pulse\_duration \* 34300 / 2

distance\_inch = pulse\_duration \* 13503.9 / 2

return distance\_cm, distance\_inch

try:

while True:

distance\_cm, distance\_inch = measure\_distance()

print("Distance: {} cm".format(distance\_cm))

print("Distance: {} inch".format(distance\_inch))

# lcd.print("Distance: {} cm".format(distance\_cm))

# lcd.print("Distance: {} inch".format(distance\_inch))

time.sleep(1)

except KeyboardInterrupt:

GPIO.cleanup()}

THANK YOU