# **Energy saving using Automated Street Lights**

*Submitted in partial fulfillment of the requirements for the degree of*

**Bachelor of Technology**

in

**Electronics and Communication Engineering**

*by*

**Maneesh Busi**

**16BEC0710**

**Under the guidance of**

**Prof. / Dr. Elizabeth Rufus**

**SENSE**

**VIT, Vellore**

May, 2020

## **DECLARATION**

I hereby declare that the thesis entitled “Energy saving using automated smart street lights” submitted by me, for the award of the degree of *Bachelor of Technology in* *Electronics and Communication Engineering* to VIT is a record of bonafide work carried out by me under thesupervision of **Prof. Elizabeth Rufus**.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore

Date :

**Signature of the Candidate**

## **CERTIFICATE**

This is to certify that the thesis entitled “Energy saving using automated smart street lights ” submitted by **Maneesh Busi, 16BEC0710, SENSE**, VIT University, for the award of thedegree of *Bachelor of Technology in Electronics and Communication Engineering*, is a record of bonafide work carried out by him under my supervision during the period, 01. 12. 2018 to 30.04.2019, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place

Date

: Vellore

:

**Signature of the Guide**

**Internal Examiner** **External Examiner**

Head of the Department

SENSE

## **ACKNOWLEDGEMENTS**

I wish to express my sincere thanks to Dr.G.Viswanathan, Chancellor, Mr. Sankar Viswanathan, Vice President, Ms. Kadhambari S. Viswanathan, Assistant Vice President, Dr.Anand A. Samuel, Vice Chancellor and Dr. P. Gunasekaran, Pro-Vice Chancellor for providing me an excellent academic environment and facilities for pursuing B.Tech. programme. I am grateful to Dr. Vaidehi Vijayakumar, Dean of School of Com-puting Science and Engineering, VIT University, Vellore. I would like to express my gratitude to my internal guide Prof. Elizabeth Rufus and my external guide Mr. A.V Satish Kumar who inspite of their busy schedule guided me in the correct path. I am thankful to **Quantela Technologies Pvt. Ltd**., Hyderabad for giving me an opportunity to work on my project and helped me gain knowledge. I thank my family and friends who motivated me during the course of the project work.

**Maneesh Busi**

## **Executive Summary**

Currently, in the world, enormous electric energy is consumed by the street lights, which are automatically turned on when it becomes dark and automatically turned off when it becomes bright. This is a huge waste of energy. This paper discusses a smart street light system.

The main aim of smart street light systems is that lights turn on when needed with the minimum required intensity so as to not use too much power, and light turn off when not needed.

The smart street light system consists of LED lights, brightness sensors, motion sensors and short-distance communication networks. The lights turn on before pedestrians and vehicles come and turn off or reduce brightness when there is no one. It will be difficult for pedestrians and drivers of vehicles to distinguish our smart street lights and the conventional street lights because our street lights all turn on before they come.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **CONTENTS** | **Page** |
|  |  |  | **No.** |
|  | **Acknowledgement** | | i |
|  | **Executive Summary** | | ii |
|  | **Table of Contents** | | iii |
|  |  | |  |
| 1 | **INTRODUCTION** | | 1 |
|  | 1.1 | Objective | 1 |
|  | 1.2 | Motivation | 1 |
|  | 1.3 | Background | 2 |
| 2 **PROJECT DESCRIPTION AND GOALS** | | | 2 |
| 3 | **TECHNICAL SPECIFICATION** | | 3 |
| 4 **DESIGN APPROACH AND DETAILS (as applicable)** | | | . |
|  | 4.1 | Design Approach / Materials & Methods | . |
|  | 4.2 | Codes and Standards | . |
|  |  |  | . |
| 5 **SCHEDULE, TASKS AND MILESTONES** | | | . |
| 6 | **PROJECT DEMONSTRATION** | | . |
| 7 | **COST ANALYSIS / RESULT & DISCUSSION (as applicable)** | |  |
| 8 | **SUMMARY** | |  |
| 9 | **REFERENCES** | |  |

|  |  |  |
| --- | --- | --- |
|  | **List of Figures** |  |
| **Figure No.** | **Title** | **Page No.** |
| 3.1 | Pinout of Arduino MEGA | 3 |
| 3.2 | GPIO Pin diagram of NodeMCU | 5 |
| 3.3 | 1. LDR 2. Circuit symbol of LDR | 6 |
| 3.4 | 1. IR model 2. IR model components 3. IR working | 7 |
| 3.5 | LED Pinout | 8 |

**List of tables**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Title** | **Page No.** |
| 3.1 | Specification of Arduino | 4 |
| 3.2 | Specification of NodeMCU | 5 |
| 3.3 | Specification of LDR | 5 |
| 3.4 | Specification of IR sensor | 8 |
| 3.5 | Specification of LED | 8 |
| 7.1 | Light Technology Comparison | 52 |

## **1. INTRODUCTION**

### **1.1. OBJECTIVE**

Automation’s main goal is to reduce manual labour or man power with the help of intelligent systems so that humans can be used for more creative and thought-based work instead of physical work. But usage of machines usually requires a lot of electrical power. So one of the main considerations when automating a particular system is cost effectiveness and power consumption.

The main aim of the project is Automatic street power saving system using Ambient Light resistor and Infra-red sensors. We can save power automatically using this system instead of someone manually doing it. So it’s also cost effective.

### **1.2. MOTIVATION**

Nowadays, in the whole world, street lights consume enormous electric energy. The number of street lights is not known accurately, but it is said that one hundred million or one billion street lights exist in the whole world. If one hundred million street lights exist, each street light consumes 20W and a half of all street lights always turn on in the whole world, 8760GWh electric power, which is about 0.8% of annual consumption of electric power in Japan. In the future, many developing countries will install many street lights and consume much electric power. Thus, to save electric power that is consumed by street lights is important to reduce greenhouse gases.

Current street lights are controlled only by means of the embedded brightness sensors; they are automatically turn on when it becomes dark and automatically turn off when it becomes bright. This is the huge waste of energy in the whole world and should be changed. There are some attempts in which the energy wastes of the street lights are reduced. A sensor light, which is controlled by a brightness sensor and a motion sensor, is sometimes used to reduce wastes of energy. It only turns on for a while when the motion is detected in front of the light and it is dark. However, usually a sensor light is too late to turn on when pedestrians or vehicles come in front of it. The light should turn on before pedestrians or vehicles come. Ideally, it is desirable that smart street lights look like usual street lights; no one notices that smart street lights are usual street lights.

### **1.3. BACKGROUND**

When you get up in the morning you may have noticed that street lights are still ON when it’s not necessary & when you travel to rural areas either there is no street lamp or there are not so many vehicles to fully utilize that facility. Simply it means the wastage of electricity. At the beginning, street lamps were controlled manually in which the control switch was set in each of the street lamps. That was called the first generation of the original street light. Another method that has been used after that was based on the optical control method. In this method the high pressure sodium lamps were used. Even the new automated street lights usually require some kind of human intervention to run perfectly. The amount of resources in the above examples is a little too much for something as simple as a street light, something which even rural areas have. Most rural areas won’t be able to afford the cost required for these resources. This project suggests an automatic system which uses less power and provides more features as compared to other street light systems.

## **2. PROJECT DESCRIPTION AND GOALS**

The automated light system controls the street light to give out the minimum intensity required to light up the surrounding area. It is a simple yet powerful concept using ambient light sensors and IR sensors.

The lights switch on and glow at a low intensity at a given time. This intensity depends on the light falling on the ambient light sensor placed near the chosen lane. The intensity is recorded and updated in our custom-made IoT UI Dashboard using a messaging protocol called MQTT.

If a vehicle happens to use the lane where the street lights are installed, the IR sensor which is placed a few metres before the street light senses the vehicle. It then sends that data to the microcontroller to turn up the intensity of the light to a 100% output.

Each IR sensor will control 3 street lights that come after it. The lights wont switch off/turn down their intensity as long as the car doesn’t hit the next IR sensor i.e. the lights are not programmes to switch off after a certain delay, instead they are made to turn off when the vehicle moves forward and passes the next “checkpost” which is the IR sensor. This little algorithm will be useful when the vehicle is not in a condition to move in the middle of the road.

The project also has a parking system which uses the same technology and code used above, but instead switches off the surrounding lights except the light placed near the parking spot.

## 

## 

## **3. TECHNICAL SPECIFICATIONS**

### **3.1 Hardware Specifications**

#### **3.1.1 Arduino Mega**

Fig 3.1 is a pinout diagram of Arduino MEGA, the microcontroller being used for this project.

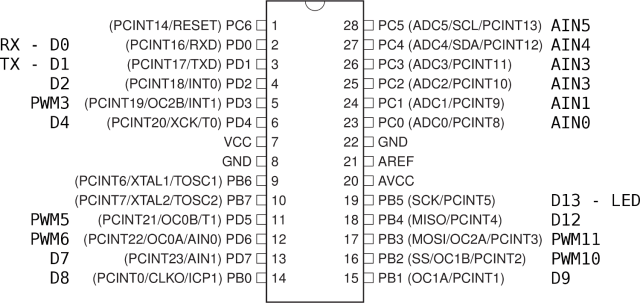
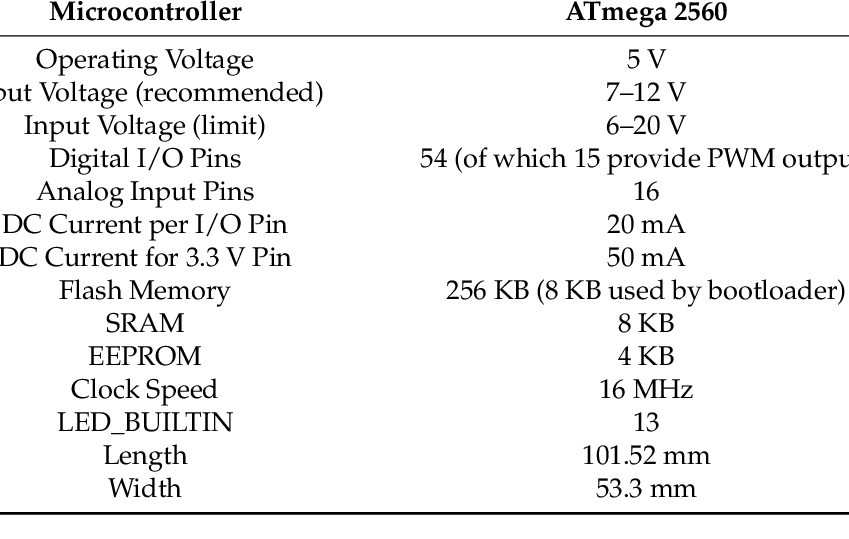


Fig. 3.1 : Pin diagram of Arduino MEGA

And Table 3.1 is the specification of the microcontroller.

Table 3.1 :- Specification of Arduino MEGA



#### **3.1.2 NodeMCU**

NodeMCU is also a microcontroller but doesn’t have enough power to control the whole system. So instead it is being used as a Wi-Fi Shield for the Arduino to connect to the internet.

Fig 3.2 contains the pinout diagram of the module. We will be using only 4 pins here. The Tx , Rx pins and 2 other pins which are set as a secondary input and output for serial communication.

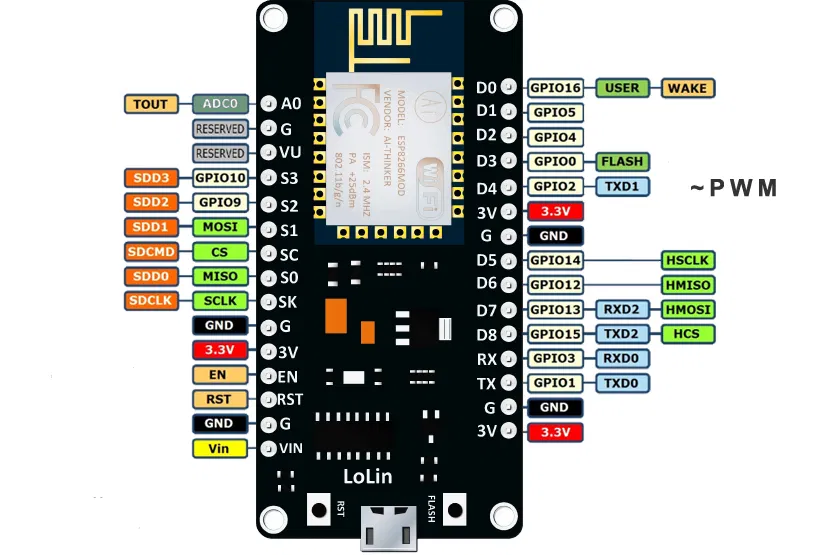


Fig 3.2 GPIO Pinout of NodeMCU

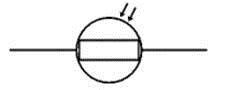
Table 3.2:- Specification of the NodeMCU

|  |  |
| --- | --- |
| **Microcontroller** | **ESP-8266 32-bit** |
| NodeMCU Size | 49mm x 26mm |
| Pin Spacing | 22.86mm |
| Clock Speed | 80MHz |
| USB to Serial | CP2102 |
| USB Connected | Micro USB |
| Operating Voltage | 3.3v |
| Input Voltage | 4.5V-10V |
| SRAM | 4MB/ 64KB |
| Digital I/O Pins | 11 |
| Analog Pins | 1 |
| ADC Range | 0-3.3V |
| UART/SPI/I2C | 1/1/1 |
| WiFi Built-in | 802.11 b/g/n |
| Temperature Range | -40C -125C |

#### **3.1.3 Light Dependent Resistor (LDR)**

An LDR is a component that has a (variable) resistance that changes with the light intensity that falls upon it. This allows them to be used in light sensing circuits.

Fig 3.3 (a) is an LDR and (b) is the circuit symbol of an LDR.

(a) (b)

Table 3.3 :- Specification for LDR

|  |  |
| --- | --- |
| **Product** | **GM5539 LDR** |
| Diameter | 5mm |
| Maximum voltage | 150V DC |
| Maximum wattage | 90mW |
| Spectral peak | 540nm |
| Light Resistance | 50-100Kohm |
| Operating temperature | -30 ~ + 70 degree Celsius |

#### 

#### 

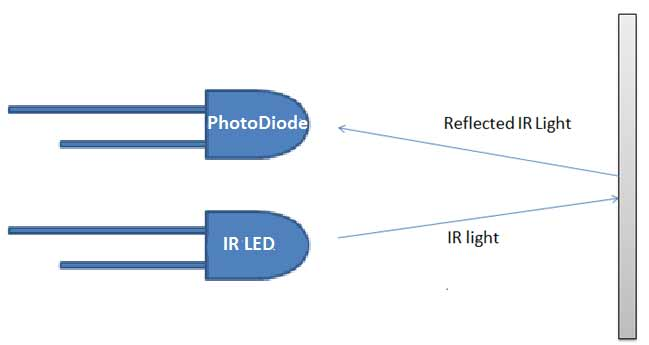
#### 

#### **3.1.4 Infrared Sensors (IR Sensor)**

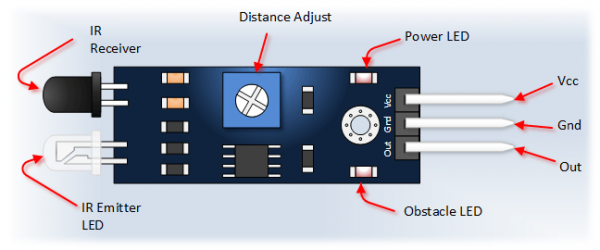
An infrared sensor is a set of an electronic transmitter and receiver device that emits IR rays in order to sense some objects in the surroundings. The range of IR sensors is normally 8-10 meters.

Fig 3.4 (a) is the IR module being used for the project, (b) labels its components and Fig 3.4 (c) explains the working of the sensor. Both the IR LED and PhotoDiode are in the same IR Module here, which is why the range of this sensor is far less than other IR sensors.

Fig 3.4

(a) (c)



(b)

Table 3.4:- Specification of IR Module

|  |  |
| --- | --- |
| **Product** | **IR Obstacle Sensor** |
| Operating Voltage | 3.0V - 5.0V |
| Detection range | 2cm -30cm |
| Current Consumption | At 3.3V: ~23mA, at 5.0V: ~43mA |
| Active output level | Outputs Low logic level when obstacle is detected |

#### **3.1.5 Street Light (LED)**

A light-emitting diode(LED) is a two lead semiconductor light source which is made up of InGaN material. It is a pn-junction diode, which emits light when activated.When a suitable voltage is applied to the leads of LED, electrons can recombine with electron holes within the device, and release energy in the form of photons.

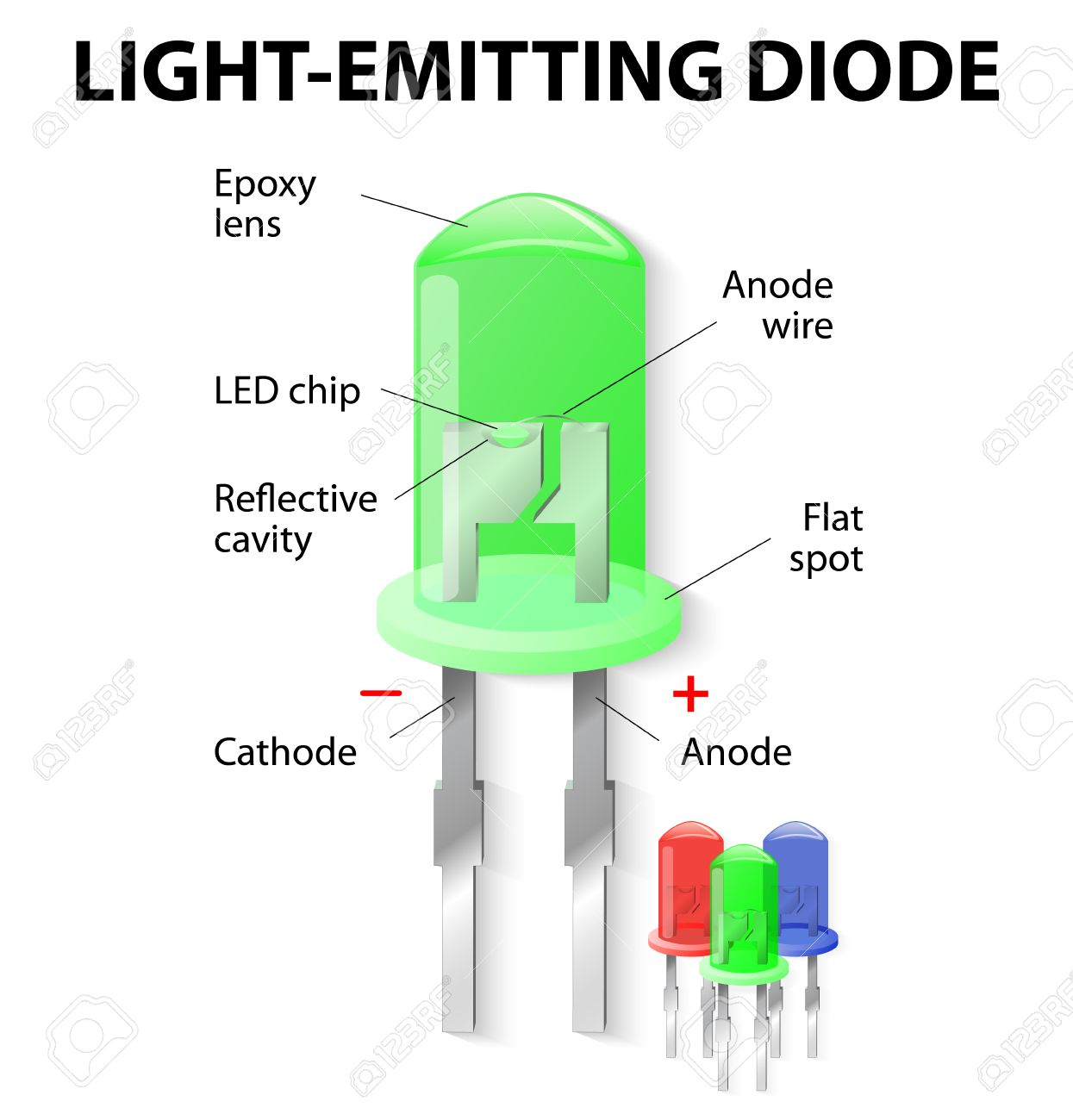


Fig 3.5 LED pinout

Table 3.5 Specifications of LED used

|  |  |
| --- | --- |
| **Product** | **Generic LEDs** |
| Diameter | 5mm |
| Voltage | 3.0V - 3.3V |

### 

### **3.2 Software Specifications**

* Arduino IDE (C)
* HTML/CSS
* JavaScript
* JQuery
* Web Browser: Explorer, Mozilla, Chrome
* Operating System: Windows

## 

## **4. DESIGN APPROACH AND DETAILS**

### **4.1 Design Approach / Materials & Methods**

#### **4.1.1 Hardware design**

* The model consists of two lanes, the first being a parking space and the other a regular one-way highway. The parking space has two spaces available and there are ambient light sensors at the end of each lane.
* When switched on, the ambient light sensor detects the light falling on it, and calculates how much light intensity is required to light up the lane i.e., how much voltage is supplied to the street lights.
* The LDR detects the light and converts it into an analog value between 0 and 255. This value is then subtracted from 255 and stored.

**Y=255-ldr**

* This Y value is then used to vary the intensity of the street lights. This is done using PWM pins on the Arduino Board.
* As this is happening, the infrared sensors (IR Sensors) are looking for objects (vehicles) that pass in front of them.
* When they detect a vehicle, the IR sensor sends a boolean value to the Arduino and the arduino increases the intensity of **2** street lights, that come right after the sensor, to full intensity regardless of the Y value.
* One of the major drawbacks of an automatic light system is that it assumes that the vehicles are constantly moving without stopping in the middle of the road. But that is not always the case. It could be a medical emergency or maybe the vehicle broke down.
* If the car is stopped right next to the sensor i.e., blocking the sensor then the street light will be on. But if the vehicle stops between two sensors, then most street lights would switch off after a while.
* This drawback is rectified. Let’s assume the first sensor is X and the first street light is Y. Let us also assume that a sensor is placed after every street light. So the lane would somewhat like this:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| X | Y | X+1 | Y+1 | X+2 | Y+2 | X+3 | Y+3 | X+4 | Y+4 | X+5 |

* Sensor X triggers lights Y and Y+1. And usually they switch off if nothing is blocking X. But in this system, the lights will only switch off if the vehicle hits the sensor that comes after the lights.
* Here X triggers Y and Y+1 to switch on. And they stay on indefinitely until the vehicle passes the X+1 sensor. Then Y is switched off and Y+2 and Y+3 are switched on. When the vehicle passes X+2 then Y+1 is switched off. This goes on until X+5 when all lights until Y+4 are switched off.
* The system also takes into account that multiple vehicles use the lane, so there is a loop that checks the IR status constantly.
* The parking lot also employs a similar working system except when it comes to the lights installed at the parking spot. In the parking lot, there are regular street lights put up along with a separate street light for each parking space.
* When a vehicle parks in the space, an IR sensor placed there detects the vehicle’s presence and informs it to the controller. Then all the surrounding street lights are switched off and only the parking space light is on.
* When the vehicle is leaving, the driver usually takes a reverse to come out of the parking spot. When going back, the vehicle will trigger an IR sensor placed in the regular lane, which will switch on the regular street light of the lane and switch off the parking space light.

#### **4.1.2 Software design**

* The Arduino board and the NodeMCU communicate serially with short messages. For example, to switch on light 1 from lane 1 the NodeMCU sends the message L11O, which means lane 1, light 1, ON. To switch it off we send L11F.
* The messages sent through MQTT are even shorter as we don’t have to specify the lane in the message. The topic to which the message is sent to tells us the lane number.
* The light levels are communicated by sending the data in a JSON Format from arduino to NodeMCU via serial communication line. Then the NodeMCU sends it to the server using MQTT protocol and the website plots the data in real time.
* There are 4 different html files that can be rendered. Namely, the home page or the dashboard, the control page for lane 1, lane 2 respectively and the contact page.
* The Paho-MQTT library for JavaScript is used to receive the data i.e. the LDR readings of both the lanes and are plotted in real time.
* The control pages have “buttons” that are used to send messages using the MQTT-protocol to the Arduino board to switch on the street lights.

### 

### **4.2 Codes and Standards**

#### **4.2.1 NodeMCU**

|  |
| --- |
| #include<ESP8266WiFi.h> #include<PubSubClient.h> #include<SoftwareSerial.h>  SoftwareSerial **s**(4,5);  const char\* ssid="VIT 2.4G"; //Wifi const char\* password="\*\*\*\*\*\*\*\*\*\*\*\*"; const char\* mqtt\_server="broker.mqttdashboard.com";  #define MQTT\_PUB "Sensors"  WiFiClient wifiClient; PubSubClient **client**(wifiClient);  //////////////////////////WiFiSETUP//////////////////////////// void **setup\_wifi**(){  delay(10);  Serial.println();  Serial.print("Connecting to ");  Serial.print(ssid);  WiFi.begin(ssid,password);  while(WiFi.status()!=WL\_CONNECTED){  delay(500);  Serial.print(".");  }  Serial.println();   Serial.println("WiFi connected");  Serial.println("IP Address: ");  Serial.println(WiFi.localIP()); }  ////////////////////////////RECONNECT////////////////////////// void **reconnect**(){  while(!client.connected()){  Serial.print("Attempting MQTT connection..");  client.publish("outTopic","hello");  if(client.connect("ESP8266Client123456789")){  Serial.println("connected");  client.subscribe("lane1");  client.subscribe("lane2");  }  else{  Serial.print("failed,rc=");  Serial.print(client.state());  Serial.println("try again in 5 seconds");  delay(5000);  }  } }    ///////////////////////////CALLBACK/////////////////////////// void **callback**(const char\* topic,byte\* payload,unsigned int length1){  Serial.print("Message arrived[");  Serial.print(topic);  Serial.print("]");   char t1[]="lane1";  char t2[]="lane2";   String msgt;   for(int i=0;i<length1;i++){  Serial.print((char)payload[i]);  msgt+=(char)payload[i];  }  Serial.println();   String so="L";    if(strcmp(topic,t1)==0){  so=so+"1";  }else if(strcmp(topic,t2)==0){  so=so+"2";  }  so=so+msgt;  s.println(so);  Serial.println(so); }    ////////////////////////////PUBLISHING/////////////////////////void publishSerialData(char \*serialData){  if(!client.connected()){  reconnect();  }  Serial.println(serialData);  client.publish(MQTT\_PUB,serialData); }   /////////////////////////////SETUP//////////////////////////// void **setup**() {  Serial.begin(115200);  s.begin(115200);  Serial.setTimeout(500);  setup\_wifi();  client.setServer(mqtt\_server,1883);  client.setCallback(callback);  reconnect(); }  /////////////////////////////LOOP////////////////////////////// void **loop**() {  client.loop();   if(Serial.available()>0){  char bfr[101];  memset(bfr,0,101);  Serial.readBytesUntil('\n',bfr,100);  publishSerialData(bfr);  } } |
|  |

#### 

#### **4.2.2 Arduino**

|  |
| --- |
| #include<SoftwareSerial.h> #include<DS3231.h>  SoftwareSerial **sw**(52, 53);  ///////////////////////PARKING/LANE1/////////////////////////// int ir\_l1[] = {A0, A1, A2, A3}; const int ir\_l1\_length = sizeof(ir\_l1) / sizeof(ir\_l1[0]); int ir\_park[] = {A8, A9}; int ir\_park\_values[2]; int ir\_l1\_values[ir\_l1\_length]; int led\_l1[] = {11, 10, 9, 8, 7, 6}; const int led\_l1\_length = sizeof(led\_l1) / sizeof(led\_l1[0]); int led\_l1\_var[led\_l1\_length]; int led\_park[] = {22, 23}; int led\_park\_values[2];   ///////////////////////////////LDR/////////////////////////////int id[] = {1, 2}; int ldr[] = {A10, A11}; const int ldr\_length = sizeof(ldr) / sizeof(ldr[0]); int ldr\_values[ldr\_length]; int y[ldr\_length]; int irm[2];   ////////////////////////////LANE2////////////////////////////// int ir\_l2[] = {A7, A6, A5, A4}; const int ir\_l2\_length = sizeof(ir\_l2) / sizeof(ir\_l2[0]); int ir\_l2\_values[ir\_l2\_length]; int led\_l2[] = {5, 4, 3}; const int led\_l2\_length = sizeof(led\_l2) / sizeof(led\_l2[0]); int led\_l2\_var[led\_l2\_length];  /////////////////////////////////////////////////////////////// ///////////////////////////FUNCTIONS//////////////////////////////////////////////////////////////////////////////////////////   //////////////////////////RECIEVE\_SUBFUNC////////////////////// void **cons**(char bfr[4]) {  Serial.print("LANE ");  Serial.print(bfr[1] + ": ");  Serial.print("LED ");  Serial.print(bfr[2] + ": ");  int k = (int)bfr[2] - 48;  int lane = (int)bfr[1] - 48;  if (lane == 1) {  if (bfr[3] == 'O') {  if (k > 6) {  digitalWrite(led\_park[k - 7], HIGH);  }  else {  analogWrite(led\_l1[k - 1], 255);  }  Serial.println("ON");  } else if (bfr[3] == 'F') {  if (k > 6) {  digitalWrite(led\_park[k - 7], LOW);  }  else {  analogWrite(led\_l1[k - 1], 0);  }  Serial.println("OFF");  }  } else if (lane == 2) {  if (bfr[3] == 'O') {  analogWrite(led\_l2[k - 1], 255);  Serial.println("ON");  } else if (bfr[3] == 'F') {  analogWrite(led\_l2[k - 1], 0);  Serial.println("OFF");  }  } }  ////////////////////////////RECEIVE//////////////////////////// void **receive**() {  if (Serial.available() > 0) {  char bfr[4];  memset(bfr, 0, 4);  Serial.readBytesUntil('\n', bfr, 3);  Serial.println(bfr);  cons(bfr);  } }  //////////////////////////IRMODE\_LANE2///////////////////////// void **irmode\_l2**() {  if (irm[1] == 0) {  ldr\_values[1] = analogRead(ldr[1]);  y[1] = 255 - ldr\_values[1];  Serial.print("Light level of lane 2: ");  Serial.print(y[1]);  delay(100);  for (int i = 0; i < 3; i++) {  analogWrite(led\_l2[i], y[1]);  }  irm[1] = irm[1] + 1;  return;  }   else {  ldr\_values[1] = analogRead(ldr[1]);  y[1] = 255 - ldr\_values[1];  if (y[1] < 0) {  y[1] = 0;  }   for (int i = 0; ir\_l2\_length; i++) {  ir\_l2\_values[i] = digitalRead(ir\_l2[i]);  if (ir\_l2\_values[i] == LOW) {  analogWrite(led\_l2[i], 255);  analogWrite(led\_l2[i + 1], 255);  led\_l2\_var[i] = 1;  led\_l2\_var[i + 1] = 2;  if (i > 0) {  led\_l2\_var[i - 1] = 0;  }  break;  }  }   for (int i = 0; i < 3; i++) {  if (led\_l2\_var[i] == 1) {  analogWrite(led\_l2[i], 255);  analogWrite(led\_l2[i + 1], 255);  }  else if (led\_l2\_var[i] == 0) {  analogWrite(led\_l2[i], y[1]);  }  }    } }  //////////////////////////IRMODE\_LANE1///////////////////////// void **irmode\_l1**() {  if (irm[0] == 0) {  ldr\_values[0] = analogRead(ldr[0]);  y[0] = 255 - ldr\_values[0];  Serial.print("Light level of lane 1: ");  Serial.print(y[0]);  delay(100);  for (int i = 0; i < 3; i++) {  analogWrite(led\_l1[i], y[0]);  }  irm[0] = irm[0] + 1;  return;  }  else {  ldr\_values[0] = analogRead(ldr[0]);  y[0] = 255 - ldr\_values[0];  if (y[0] < 0) {  y[0] = 0;  }   for (int i = 0; i < ir\_l1\_length; i++) {  ir\_l1\_values[i] = digitalRead(ir\_l1[i]);  if (ir\_l1\_values[i] == LOW) {  analogWrite(led\_l1[i], 255);  if (i > 0) {  analogWrite(led\_l1[i + 1], 255);  led\_l1\_var[i - 1] = 0;  }  led\_l1\_var[i] = 1;  led\_l1\_var[i + 1] = 2;  break;  }  }   for (int i = 0; i < 2; i++) {  led\_park\_values[i] = digitalRead(ir\_park[i]);  if (led\_park\_values[i] == LOW) {  digitalWrite(led\_park[i], HIGH);  } else {  delay(1000);  digitalWrite(led\_park[i], LOW);  }  }   for (int i = 0; i < led\_l1; i++) {  if (led\_l1\_var[i] == 1) {  analogWrite(led\_l1[i], 255);  if (i > 0) {  analogWrite(led\_l1[i], 255);  }  }  else if (led\_l1\_var[i] == 0) {  analogWrite(led\_l1[i], y[0]);  }  }  } }  void **setup**() {  Serial.begin(115200);  Serial.println("Interfacing with NodeMCU");  sw.begin(115200);  for (int i = 0; i < 2; i++) {  pinMode(ldr[i], INPUT);  pinMode(ir\_park[i], INPUT);  pinMode(led\_park[i], OUTPUT);  }   for (int i = 0; i < 3; i++) {  pinMode(led\_l2[i], OUTPUT);  }   for (int i = 0; i < 4; i++) {  pinMode(ir\_l1[i], INPUT);  pinMode(ir\_l2[i], INPUT);  }   for (int i = 0; i < led\_l1\_length; i++) {  pinMode(led\_l1[i], OUTPUT);  } }   void **loop**() {  ldr\_values[0] = analogRead(ldr[0]);  ldr\_values[1] = analogRead(ldr[1]);  Serial.print("{\"sensorid\":");  Serial.print(id[0]);  Serial.print("\"ldr");  Serial.print(1);  Serial.print("\":");  Serial.print(ldr\_values[0]);  Serial.print("}");  Serial.println();  Serial.print("{\"sensorid\":");  Serial.print(id[1]);  Serial.print("\"ldr");  Serial.print(2);  Serial.print("\":");  Serial.print(ldr\_values[1]);  Serial.print("}");  Serial.println();   sw.print("{\"sensorid\":");  sw.print(id[0]);  sw.print("\"ldr");  sw.print(1);  sw.print("\":");  sw.print(ldr\_values[0]);  sw.print("}");  sw.println();  sw.print("{\"sensorid\":");  sw.print(id[1]);  sw.print("\"ldr");  sw.print(2);  sw.print("\":");  sw.print(ldr\_values[1]);  sw.print("}");  sw.println();   if (ldr\_values[0] < 100 || ldr\_values[1] < 100) {  irmode\_l1();  irmode\_l2();  receive();  } else {  receive();  } } |

**4.2.3 HTML**

##### **4.2.3.1 Dashboard**

|  |
| --- |
| **<!DOCTYPE html>** <html lang="en" dir="ltr">  <head>  <meta charset="utf-8">  <script src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.js" type="text/javascript"></script>   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">   <script src="https://cdn.plot.ly/plotly-latest.min.js" charset="utf-8"></script>  <script src="https://cdn.jsdelivr.net/npm/moment@2.24.0/min/moment.min.js"></script>  <script src="https://cdn.jsdelivr.net/npm/chart.js@2.8.0"></script>  <script src="https://cdn.jsdelivr.net/npm/chartjs-plugin-streaming@1.8.0"></script>   <script src="https://kit.fontawesome.com/1aa08a628b.js" crossorigin="anonymous"></script>   <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300&display=swap" rel="stylesheet">  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@600&display=swap" rel="stylesheet">   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>  <link rel="stylesheet" href="styles.css">  <script src="app.js" charset="utf-8"></script>   <title>Dashboard</title> </head>  <body>  <div id="mySidebar" class="sidebar" onmouseover="toggleSidebar()" onmouseout="toggleSidebar()">  <a href="dashboard.html"><i class="fas fa-tachometer-alt sidebar-icons"></i>DASHBOARD</a>  <a href="lane1.html"><i class="fas fa-road sidebar-icons"></i>LANE 1</a>  <a href="lane2.html"><i class="fas fa-road sidebar-icons"></i>LANE 2</a>  <a href="contact.html"><i class="far fa-envelope sidebar-icons"></i>CONTACT</a>  </div>   <div id="main" class="container">  <div class="row row-1">  <div class="col col-lg-3 col-md-12">  <div class="card xs">  <div class="card-body">  <h4 class="card-title"><i class="fas fa-car dash-icon"></i>Cars Parked</h4>  <h1 class="card-text">  2  </h1>  </div>  </div>  </div>  </div>   <div class="row row-2">  <div class="col col-lg-6 col-md-12">  <div id="sensor-1" class="card sensor">  <h4 class="graph-title">Sensor-1</h4>  <canvas id="sens-1" class="sens-graph"></canvas>  </div>  </div>   <div class="col col-lg-6 col-md-12">  <div id="sensor-2" class="card sensor">  <h4 class="graph-title">Sensor-2</h4>  <canvas id="sens-2" class="sens-graph"></canvas>  </div>  </div>  </div>  </div>  </body> </html> |
|  |

##### **4.2.3.2 Lane 1**

|  |
| --- |
| **<!DOCTYPE html>** <html lang="en" dir="ltr">  <head>  <meta charset="utf-8">  <script src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.js" type="text/javascript"></script>   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">   <script src="https://cdn.plot.ly/plotly-latest.min.js" charset="utf-8"></script>  <script src="https://cdn.jsdelivr.net/npm/moment@2.24.0/min/moment.min.js"></script>  <script src="https://cdn.jsdelivr.net/npm/chart.js@2.8.0"></script>  <script src="https://cdn.jsdelivr.net/npm/chartjs-plugin-streaming@1.8.0"></script>   <script src="https://kit.fontawesome.com/1aa08a628b.js" crossorigin="anonymous"></script>   <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300&display=swap" rel="stylesheet">  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@600&display=swap" rel="stylesheet">   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>  <link rel="stylesheet" href="styles.css">  <script src="app.js" charset="utf-8"></script>   <title>Lane-1</title>  </head>  <body>  <div id="mySidebar" class="sidebar" onmouseover="toggleSidebar()" onmouseout="toggleSidebar()">  <a href="dashboard.html"><i class="fas fa-tachometer-alt sidebar-icons"></i>DASHBOARD</a>  <a href="lane1.html"><i class="fas fa-road sidebar-icons"></i>LANE 1</a>  <a href="lane2.html"><i class="fas fa-road sidebar-icons"></i>LANE 2</a>  <a href="contact.html"><i class="far fa-envelope sidebar-icons"></i>CONTACT</a>  </div>   <div id="main" class="container">  <div class="card controls">  <div class="card-body">  <h1 class="card-title control-title">Lane-1 Manual</h1>  <br />  <ul class="switches">  <div class="row l1-r1">  <div class="col col-lg-4 col-4">  <li id="l1\_1" class="switch" onclick="ss(event)"><div class="status">ON</div><br /><div class=stat-var>Lane-1 Light-1</div></li>  </div>  <div class="col col-lg-4 col-4">  <li id="l1\_2" class="switch" onclick="ss(event)"><div class="status">ON</div><br /><div class=stat-var>Lane-1 Light-2</div></li>  </div>  <div class="col col-lg-4 col-4">  <li id="l1\_3" class="switch" onclick="ss(event)"><div class="status">ON</div><br /><div class=stat-var>Lane-1 Light-3</div></li>  </div>  <div class="col col-lg-4 col-4">  <li id="l1\_4" class="switch" onclick="ss(event)"><div class="status">ON</div><br /><div class=stat-var>Lane-1 Light-4</div></li>  </div>  <div class="col col-lg-4 col-4">  <li id="l1\_5" class="switch" onclick="ss(event)"><div class="status">ON</div><br /><div class=stat-var>Lane-1 Parking-1</div></li>  </div>  <div class="col col-lg-4 col-4">  <li id="l1\_6" class="switch" onclick="ss(event)"><div class="status">ON</div><br /><div class=stat-var>Lane-1 Parking-2</div></li>  </div>    </ul>  </div>   </div>   </div>  </body> </html> |

##### **4.2.3.3 Lane 2**

|  |
| --- |
| **<!DOCTYPE html>** <html lang="en" dir="ltr">  <head>  <meta charset="utf-8">  <script src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.js" type="text/javascript"></script>   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">   <script src="https://cdn.plot.ly/plotly-latest.min.js" charset="utf-8"></script>  <script src="https://cdn.jsdelivr.net/npm/moment@2.24.0/min/moment.min.js"></script>  <script src="https://cdn.jsdelivr.net/npm/chart.js@2.8.0"></script>  <script src="https://cdn.jsdelivr.net/npm/chartjs-plugin-streaming@1.8.0"></script>   <script src="https://kit.fontawesome.com/1aa08a628b.js" crossorigin="anonymous"></script>   <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300&display=swap" rel="stylesheet">  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@600&display=swap" rel="stylesheet">   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>  <link rel="stylesheet" href="styles.css">  <script src="app.js" charset="utf-8"></script>   <title>Lane-2</title>  </head>  <body>  <div id="mySidebar" class="sidebar" onmouseover="toggleSidebar()" onmouseout="toggleSidebar()">  <a href="dashboard.html"><i class="fas fa-tachometer-alt sidebar-icons"></i>DASHBOARD</a>  <a href="lane1.html"><i class="fas fa-road sidebar-icons"></i>LANE 1</a>  <a href="lane2.html"><i class="fas fa-road sidebar-icons"></i>LANE 2</a>  <a href="contact.html"><i class="far fa-envelope sidebar-icons"></i>CONTACT</a>  </div>   <div id="main" class="container">  <div class="card controls">  <div class="card-body">  <h1 class="card-title control-title">Lane-2 Manual</h1>  <br />  <ul class="switches">  <div class="row l2-r1">  <div class="col col-lg-4 col-4">  <li id="l2\_1" class="switch"><div class="status">ON</div><br /><div class=stat-var>Lane-2 Light-1</div></li>  </div>  <div class="col col-lg-4 col-4">  <li id="l2\_2" class="switch"><div class="status">ON</div><br /><div class=stat-var>Lane-2 Light-2</div></li>  </div>  <div class="col col-lg-4 col-4">  <li id="l3\_3" class="switch"><div class="status">ON</div><br /><div class=stat-var>Lane-2 Light-3</div></li>  </div>  </ul>  </div>   </div>   </div>   </body> </html> |

##### 

##### **4.2.3.4 Contact**

|  |
| --- |
| **<!DOCTYPE html>** <html lang="en" dir="ltr">  <head>  <meta charset="utf-8">  <script src="https://cdnjs.cloudflare.com/ajax/libs/paho-mqtt/1.0.1/mqttws31.js" type="text/javascript"></script>   <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">   <script src="https://cdn.plot.ly/plotly-latest.min.js" charset="utf-8"></script>  <script src="https://cdn.jsdelivr.net/npm/moment@2.24.0/min/moment.min.js"></script>  <script src="https://cdn.jsdelivr.net/npm/chart.js@2.8.0"></script>  <script src="https://cdn.jsdelivr.net/npm/chartjs-plugin-streaming@1.8.0"></script>   <script src="https://kit.fontawesome.com/1aa08a628b.js" crossorigin="anonymous"></script>   <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300&display=swap" rel="stylesheet">  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@600&display=swap" rel="stylesheet">   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>  <link rel="stylesheet" href="styles.css">  <script src="app.js" charset="utf-8"></script>   <title>Contact</title>  </head>  <body>  <div id="mySidebar" class="sidebar" onmouseover="toggleSidebar()" onmouseout="toggleSidebar()">  <a href="dashboard.html"><i class="fas fa-tachometer-alt sidebar-icons"></i>DASHBOARD</a>  <a href="lane1.html"><i class="fas fa-road sidebar-icons"></i>LANE 1</a>  <a href="lane2.html"><i class="fas fa-road sidebar-icons"></i>LANE 2</a>  <a href="contact.html"><i class="far fa-envelope sidebar-icons"></i>CONTACT</a>  </div>   <div id="main" class="container">  <div class="card controls">  <div class="contact-details">  <ul class="contact-list">  <li>  <img class="contact-image" src="download.png" />  </li>  <li class="contact-item">  Maneesh Busi  </li>  <li class="contact-item">  16BEC0710  </li>  <li class="contact-item">  email : moses.maneesh@gmail.com  </li>  <li class="contact-item">  Phone : 7675990088  </li>  </ul>   </div>  </div>   </div>   </body> </html> |

#### **4.2.4 CSS**

|  |
| --- |
| **html**{  width:100%;  height:100%;  display:table; }  **body**{  min-height:100%;  background-color: #1e1e2f;  color:#ffffff; }  */\*-----------------------------mainpagepart---------------------------\*/* @**media** screen and (max-height:450px) {  .sidebar {padding-top: 0.6vh;}  .sidebar **a** {font-size: 7.2vh;}  .sidebar {width:2vw;} }  .container{  max-height: 100vh;  margin:5vh;  max-width:95vw;  padding:5px;  margin-top:1vmax; }  **#main**{  transition: 0.5s;  margin-left:5.4vmax; }  .row{  margin-left:0; }   */\*--------------------------sidebarstuff------------------------------\*/*  .sidebar{  height:100vh;  width:4.5vmax;  position:fixed;  z-index: 1;  top:0;  left:0;  background: linear-gradient(to bottom, #fd5d92 0%, #ec250d 100%);  overflow-x: hidden;  padding-top:10vh;  white-space: nowrap;  transition: 0.5s; }  .sidebar **a**{  padding:1.4vh 1.4vh 1.4vh 1.5vmax;  text-decoration: none;  display:block;  color:#ffffff;  font-size:1.4vmax;  transition: 0.5s; }  .sidebar-icons{  padding-bottom:1.1vh;  margin-bottom:1.7vh;  margin-right:2vw; }  */\*----------------------------cardstuff------------------------------\*/*  .card{  text-align:left;  */\* margin-left:3.4vmax; \*/*  margin-bottom:2.4vmax;  margin-right:1vmax;  border-radius: 5px;   background-color: #27293d;  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19); }  .card-body{  padding:1vmax;  padding-bottom:0; }  .card-title{  font-size: 1.5vmax;  font-family: 'Poppins',sans-serif;   margin-bottom:0.5vmax;   transition: all 0.5s; }  .card-text{  padding-top: 1vmax;  font-size: 6vmax;  text-align: center; }  .dash-icon{  margin-right:0.5vmax;  color:#fff; }  .xs{  width:15vmax;  height:15vmax; }  .no\_of\_parked{  font-size:20vh;  text-align: center;  position: fixed; }  .sensor{  width:43vmax;  height:24vmax;  padding:20px 30px 3.3vmax 20px; }  .graph-title{  font-size:1.5vmax;  font-family:'Poppins',sans-serif;  margin-bottom:0.5vmax;   transition:all 0.5s }  .sens-graph{  padding-top:25px; }  */\*----------------------------switches--------------------------------\*/*  .controls{  width:91vmax;  height:auto; }  .control-title{  font-size:2.5vmax;  }  .switches {  list-style: none;  padding: 0;  margin-left:10vmax; }  .switch{  margin:2vmax 3vmax;  }  .status{  font-size:2vmax;  height:10vmax;  line-height:6.4vmax;  text-align: center;  padding-top:1.8vmax;  width:10vmax;   border-radius: 100%;  color:#fff;  box-shadow: inset 0 0 0 1px #d7d7d7;  position:relative;   cursor: grab;  -moz-transition: all 0.2s ease-out;  -o-transition:all 0.2s ease-out;  -webkit-transition:all 0.2s ease-out;   transition:all 0.2s ease-out; }  .status:hover, .status:focus{  color:#fff;  box-shadow: inset 0 0 0 2px rgba(255,255,255,0.8);  transform: scale(1.1); }  .status.clicked{  background-color: #fff;  color:#ec250d;  border-style: solid;  border-color: #ec250d;  transform: scale(1.1);  padding-top:1.5vmax; }  .stat-var{  font-family: 'Poppins',sans-serif;  font-size: 1.2vmax;  margin-left:1vmax;  margin-bottom:3vmax; }  /\*--------------------------------contact--------------------------\*/  .contact-details{  text-align: center;  }  .contact-list{  list-style: none;  }  .contact-image{  height:10%;  width:10%;  border-radius: 100%;  margin:2vmax  }  .contact-item{  margin:0.5vmax;  font-family: 'Poppins',sans-serif;  } |

#### **4.2.5 JavaScript**

|  |
| --- |
| **var** bar\_status = true; **var** msg; **var** x,y; client = **new** Paho.MQTT.Client("broker.mqttdashboard.com", Number("8000"), "clientId-Y5IcOSSr2J");  *// set callback handlers* client.onConnectionLost = onConnectionLost; client.onMessageArrived = onMessageArrived;  *// connect the client* client.connect({onSuccess:onConnect});   *// called when the client connects* **function** **onConnect**() {  *// Once a connection has been made, make a subscription and send a message.*  console.log("onConnect");  client.subscribe("World");  client.subscribe("/Sensors");  message = **new** Paho.MQTT.Message("on 1");  message.destinationName = "World";  client.send(message); }  *// called when the client loses its connection* **function** **onConnectionLost**(responseObject) {  **if** (responseObject.errorCode !== 0) {  console.log("onConnectionLost:"+responseObject.errorMessage);  } }  *// called when a message arrives* **function** **onMessageArrived**(message) {  console.log(message.destinationName)  console.log(message.payloadString);  **if**(message.destinationName==="/Sensors"){   **var** sensorJSON=JSON.parse(message.payloadString);  x=Number(sensorJSON.ldr1);   y=Number(sensorJSON.ldr2);  console.log("ARRIVED "+y);   }   }  **function** **toggleSidebar**() {  **if** (bar\_status) {  $("#mySidebar").css('width', '16vmax')  $("#main").css('marginLeft', '20vmax')  **this**.bar\_status = false;  } **else** {  $("#mySidebar").css('width', '4.5vmax')  $("#main").css('marginLeft', '5.5vmax')  **this**.bar\_status = true;  } }  **function** **ss**(event){  **var** ev=event.path[1].id.split("");  **var** lane=ev[1];  **var** light=ev[3];   **var** msgid="l"+lane+"\_"+light;  **var** k;  console.log(document.getElementById(msgid).querySelector(".status").innerText);  **if**(document.getElementById(msgid).querySelector(".status").innerText=="OFF"){  k=light+"O";  console.log(k);  msg=**new** Paho.MQTT.Message(k);  }**else** **if**(document.getElementById(msgid).querySelector(".status").innerText=="ON"){  k=light+"F";  console.log(k);  msg=**new** Paho.MQTT.Message(k);  }  msg.destinationName="Lane1";  client.send(msg); }  $(document).ready(**function**() {   $(".status").click(**function**() {  $(**this**).toggleClass("clicked");  **if**($(**this**).hasClass("clicked")){  $(**this**).text("OFF");  }**else**{  $(**this**).text("ON");  }  });   **function** **randomScalingFactor**() {  **return** (Math.random());  }   **function** **sensdata1**(){  **return** x;  }   **function** **sensdata2**(){  console.log("ASSIGNED "+y);  **return** y;  }   **function** **onRefresh**(chart) {  chart.config.data.datasets[0].data.push({  x: Date.now(),  y: sensdata1()  });  }   **function** **onRefresh2**(chart) {  chart.config.data.datasets[0].data.push({  x: Date.now(),  y: sensdata2()  });  }   **var** ctx = document.getElementById("sens-1").getContext('2d')  **var** ctx2 = document.getElementById("sens-2").getContext('2d')   **var** chart = **new** Chart(ctx, {  *// The type of chart we want to create*  type: 'line',   *// The data for our dataset*  data: {  datasets: [{  backgroundColor: '#ec250d',  borderColor: '#ec250d',  lineTension: 0,  fill: false,  *// borderDash:[8,4],*  data: []  }]  },   *// Configuration options go here*  options: {  *// title:{*  *// display:true,*  *// text:'Charrt'*  *// },*   scales: {  xAxes: [{  type: 'realtime',  realtime: {  duration: 20000,  refresh: 1000,  delay: 2000,  onRefresh: onRefresh  },  gridLines: {  drawOnChartArea: true  }  }],   yAxes: [{  scaleLabel: {  display: false,  labelString: 'value'  },  gridLines: {  drawOnChartArea: true  }  }]  },  legend: {  display: false  },  tooltips: {  mode: 'nearest',  intersect: false  },  hover: {  mode: 'nearest',  intersect: false  },  maintainAspectRatio: false  }  });   **var** chart2 = **new** Chart(ctx2, {  *// The type of chart we want to create*  type: 'line',   *// The data for our dataset*  data: {  datasets: [{  backgroundColor: '#ec250d',  borderColor: '#ec250d',  lineTension: 0,  fill: false,  *// borderDash:[8,4],*  data: []  }]  },   *// Configuration options go here*  options: {  scales: {  xAxes: [{  type: 'realtime',  realtime: {  duration: 20000,  refresh: 1000,  delay: 2000,  onRefresh: onRefresh2  },  gridLines: {  drawOnChartArea: true  }  }],   yAxes: [{  scaleLabel: {  display: false,  labelString: 'value'  },  gridLines: {  drawOnChartArea: true  }  }]  },  legend: {  display: false  },  tooltips: {  mode: 'nearest',  intersect: false  },  hover: {  mode: 'nearest',  intersect: false  },  maintainAspectRatio: false  }  });  }) |

## **5. SCHEDULE, TASKS AND MILESTONES**

### 

### **5.1 Schedule**

* **January:** Learning about the MQTT protocol and Node-red Software to exchange messages with the microcontroller (Arduino)
* **February:** Creating the “flow” in node-red to manually control the Arduino and using the LDR to control the intensity of the lights and make a graph that plots the intensity.
* **March:** Using a weather API to get information about the sunset and sunrise so that the system can be automated on the basis of these timings. Trying to write a machine learning algorithm to recognize ambulances so that the next street can be informed about it.
  + Making a draft of the report and ppt.
* **April:** Started a web development course to make a custom UI for the project instead of the in-built UI of the Node-red Software.
* **May:** Completed creating a website for the project and integrating it with the hardware. Making changes in the report to add the new software specifications and design process.

### **5.2 Tasks**

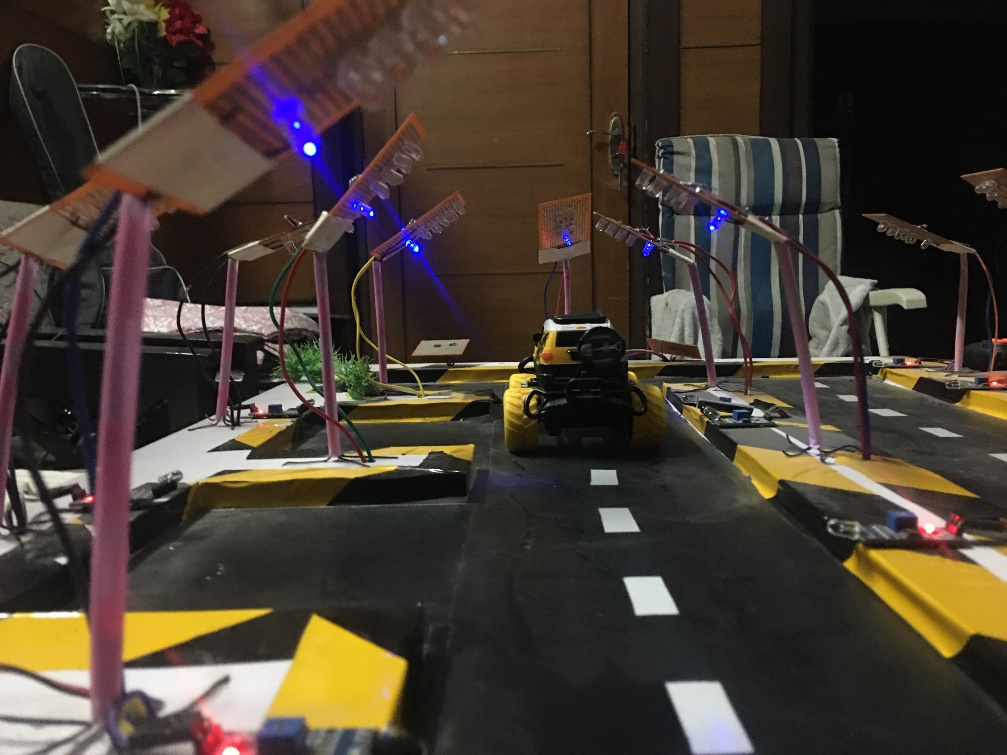
* Making a small scale working model of the project i.e., with only 3 IR sensors and 2 LEDs and basic code.
* Use Node-red and MQTT Protocol to manually control the lights from Dashboard.
* Send light intensity data from arduino onto the dashboard and map the values to the corresponding lane.
* Make a full scale model.
* Create the outline of the website for a custom UI dashboard.
* Add functionality.

### **5.3 Milestones**

* Created a basic circuit design of the project and tested it.
* Manually controlled lights using MQTT messages.
* Uploaded light data to the web and plotted it.
* Varied the intensity of the street lights in accordance with the light level reading taken by the ambient light sensor.
* Made a full scale model.
* Created a website and added MQTT functionality to it.

## 

## **6. PROJECT DEMONSTRATION**

****



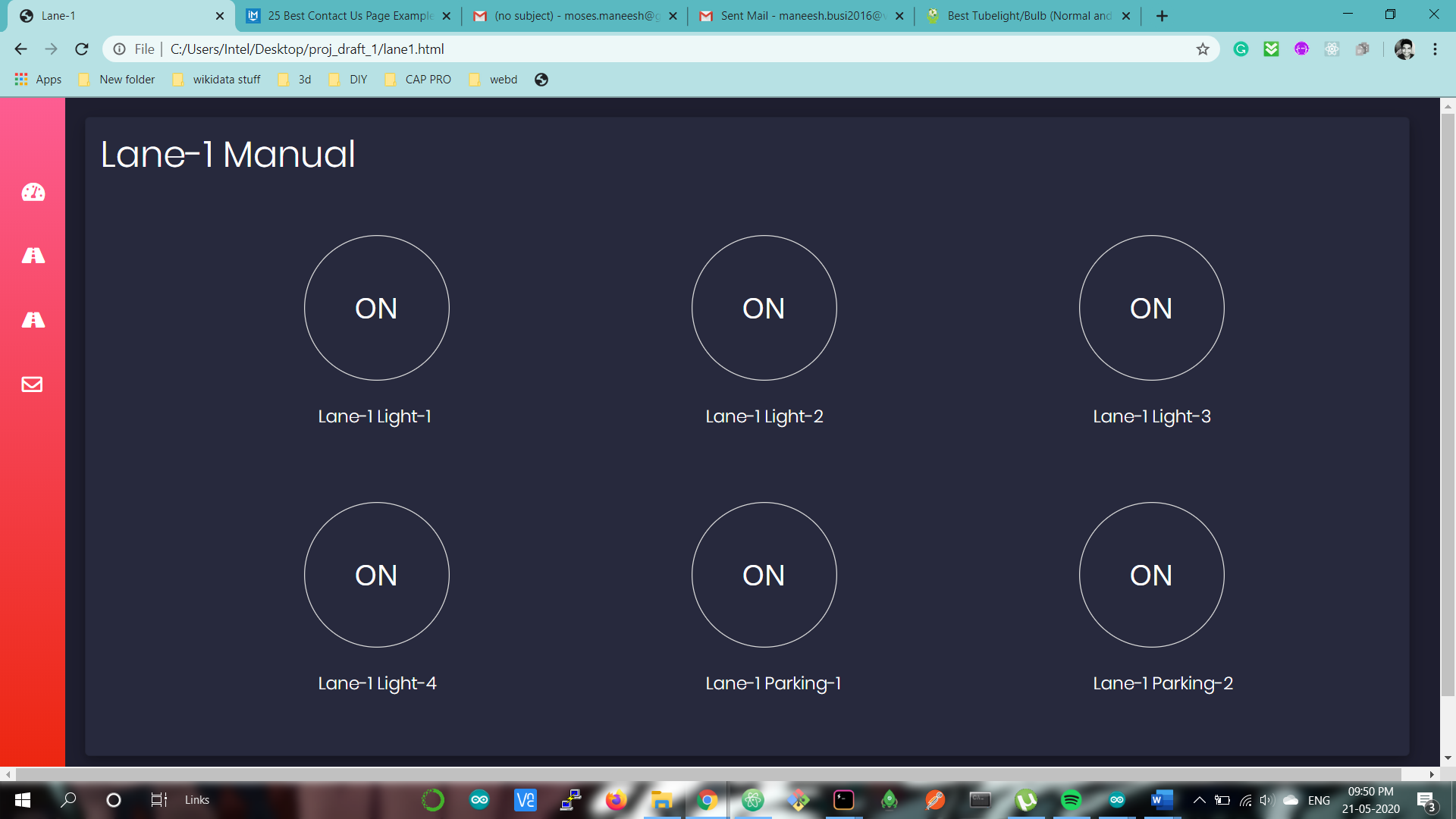


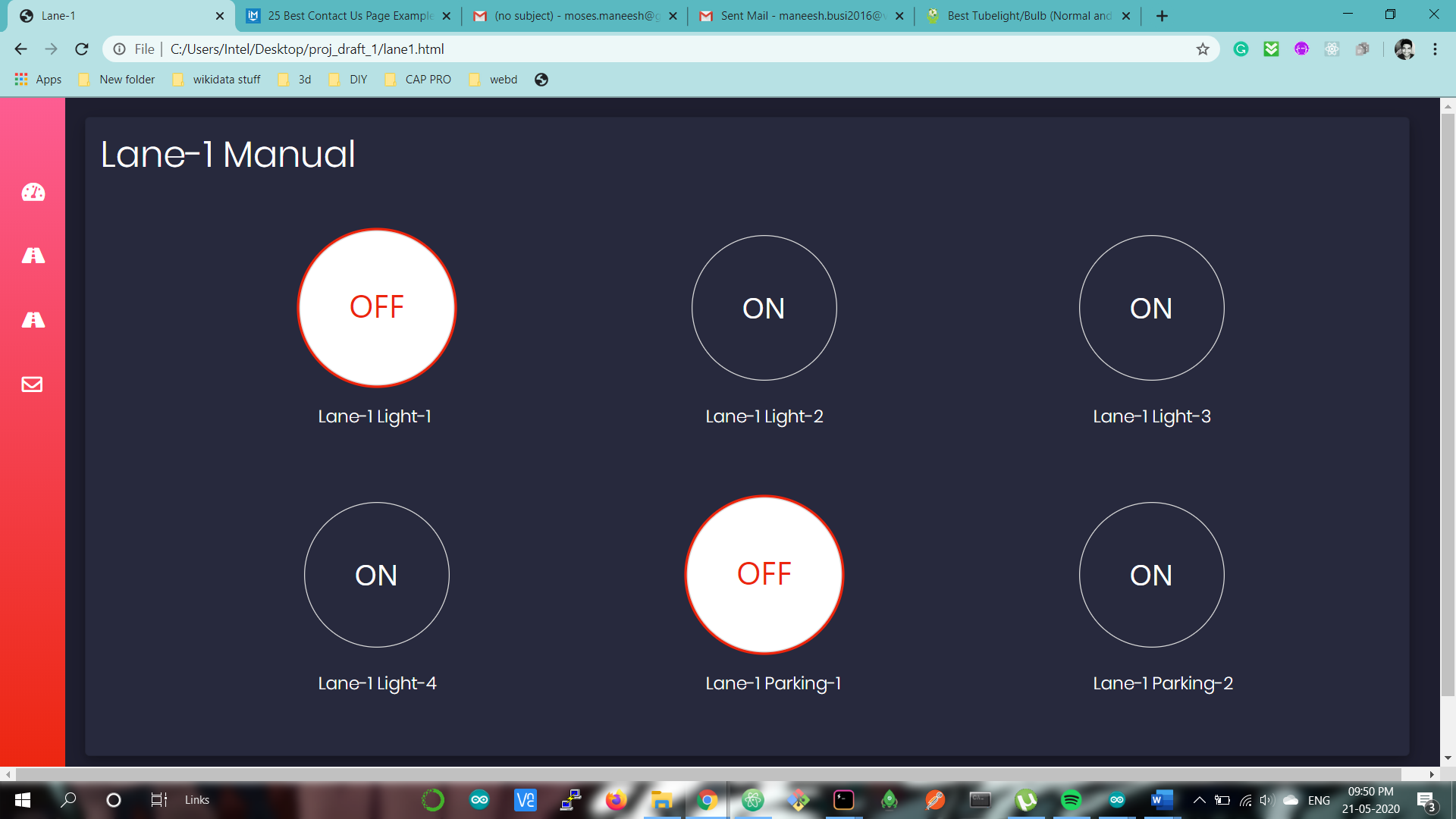


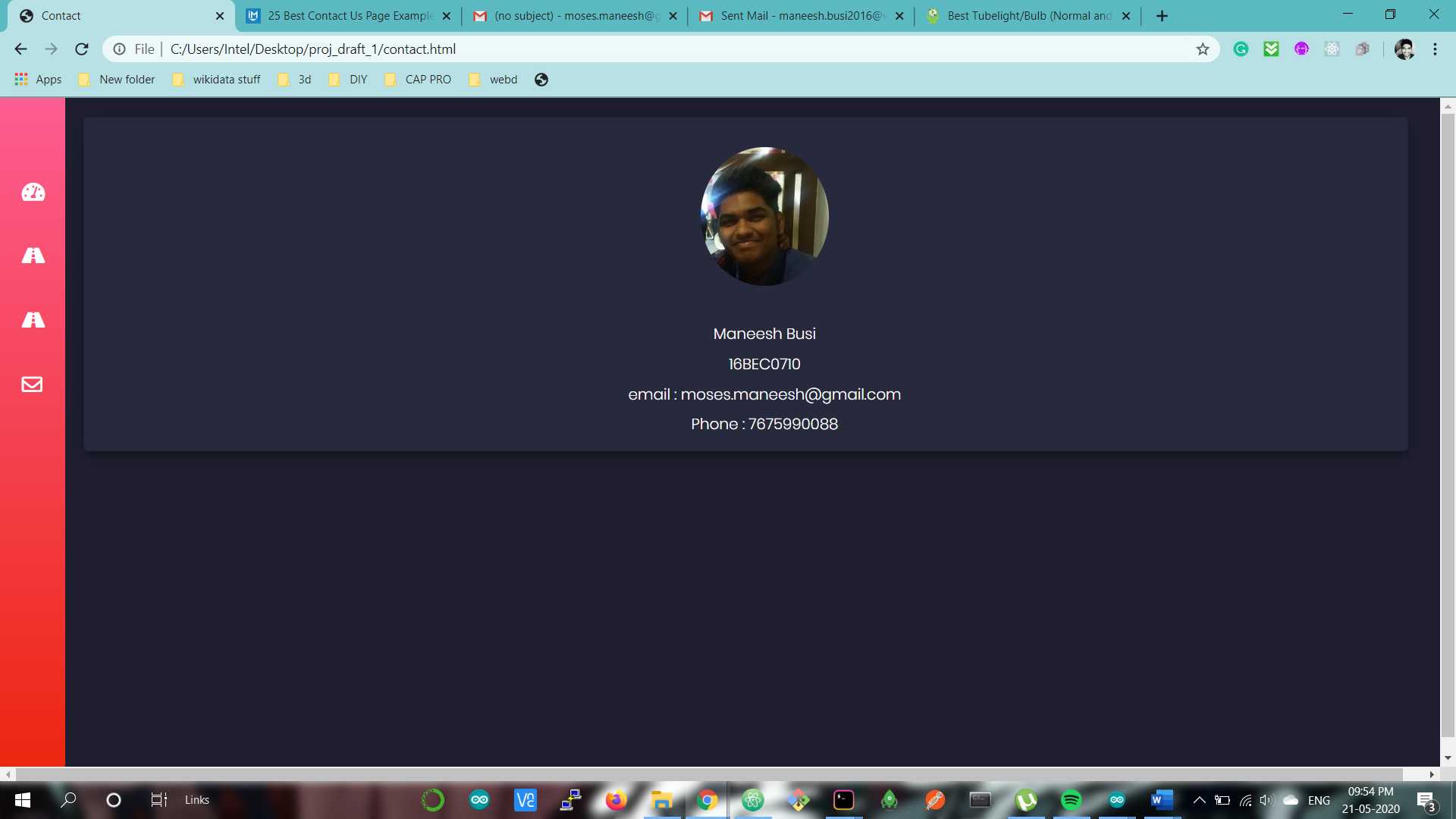












## 

## **7. COST ANALYSIS / RESULT & DISCUSSION**

Most developed countries around the world have adopted LED street light technology whereas our country still uses CFL (Compact Fluorescent Lamp), metal halides or sodium vapor lights. There are more than 27 million street lights in India and take up between 20 to 40% of the energy produced in our country.

Table 7.1 shows the comparison between different light technologies.

Table 7.1 : Light Technology Comparison

|  |  |  |  |
| --- | --- | --- | --- |
| **Light Technology** | **Average Lamp Life in Hours** | **Lumens per Watt** | **Consideration** |
| Incandescent | 1000-5000 | 11-15 | Very inefficient and short life time |
| Mercury Vapor | 12000-24000 | 13-48 | Very inefficient, ultraviolet radiation and contains mercury |
| Metal Halide | 10000-15000 | 60-100 | High maintenance UV radiation contains mercury and lead, risk of bursting at the end of life |
| High Pressure Sodium | 12000-24000 | 45-130 | Contains mercury and lead |
| Low Pressure Sodium | 10000-18000 | 80-180 | Contains mercury and lead |
| Fluorescent | 10000-20000 | 60-100 | UV radiation contains mercury, prone to glass breaking and diffused non-directional light. |
| Compact Fluorescent | 12000-20000 | 50-72 | Low life/burn out, dimmer in cold weather and contains mercury |
| Induction | 60000-100000 | 70-90 | Higher initial cost,limited directionality, contains lead and negatively affected by heat |
| LED | 50000-100000 | 70-150 | Relatively higher initial cost |

We can tell from the above table that LED street lights have a higher lamp life and lumens per watt consumed, making it ideal for street lights. And by using the system proposed in this project, we can further reduce the power consumption. The lights only turn on with full intensity when a vehicle passes by, so if there is a lot of traffic the lights would be at full intensity throughout but if we take a highway as an example, not many vehicles use a highway during night except for goods carriers.

Let us take a single LED street light that consumes 60W of power per hour at full intensity. If it is turned on from 6pm to 6am then it consumes 720W of power. If this system is implemented, then when there are no vehicles the street light will be at 40 to 50% brightness depending on its surroundings, because if it is located near a city then the city lights would illuminate the road and the sensor will record a higher light intensity causing the LED street light to glow at 20% or less intensity. So, if we assume it goes down to 40% i.e., 24W of power. Depending on the number of vehicles passing through the number of times it burns with full intensity changes.

The best case scenario is that no vehicles used the road that night which means only 288W were consumed which is far less than 720W. And the worst case is that the road was really busy that night and the lights were always at full intensity which means it is at 720W. By this we can tell that the amount of power consumed will always be 720 or less, and the average power consumed will be 504W.

The street lights we use today use 250W of power per hour. So in a month the energy consumed by a single street light is 90kW. If we were to use the proposed system with automated LED lights then the average power consumed will be 21.6kW. Which is already 75% less than normal street lights. Using our system, average power consumed per night is 504W, so in a month it is 15.12kW.

Therefore by using the proposed system energy consumption can be reduced by almost 85%.

## **8. SUMMARY**

## The aim of this project was to automate street light systems using light sensors and infrared sensors to reduce the consumption of power. This system uses many different ideas to further reduce manual intervention. Light sensors are used to detect the light intensity in the area and adjust the light emitted by the street light accordingly. IR sensors have been used to detect any vehicle, so that the street light can glow with full intensity as long as the vehicle stays. The light intensity recorded is sent to the web using MQTT protocol and plotted. The lights can also be manually controlled if necessary using the same messaging protocol. A custom dashboard has also been created along with the IoT project to further enhance user experience.

## 

## **9. REFERENCES**

[1] <https://ieeexplore.ieee.org/document/6518523>

[2] <https://ieeexplore.ieee.org/document/8389131>

[3] <https://ieeexplore.ieee.org/document/8389131>

[4] <https://ieeexplore.ieee.org/document/6072726?arnumber=6072726>

[5] <https://ijssst.info/Vol-13/No-1/paper5.pdf>