IoT based Weather Adaptive Street lighting system

Project Report

1. INTRODUCTION :

The purpose of this project is to design an IoT-based weather adaptive street lighting system. The system will utilize various sensors and data analysis techniques to dynamically adjust street lighting based on real-time weather conditions. This design aims to improve energy efficiency and enhance safety by optimizing street lighting levels according to weather conditions

LDR sensor is used for detecting the presence of surrounding light so that during the day time when sun is bright, the street light is switched off automatically. During the night time when there is no light, the LDR sends signal to microcontroller to turn on the street light.

Project Overview :

The IoT-based Weather Adaptive Street Lighting System is an advanced solution that aims to optimize the performance and energy efficiency of street lighting infrastructure. By utilizing the power of the Internet of Things (IoT), this system enables intelligent and adaptive control of street lights based on real-time weather condition

Purpose :

1. Improve Energy Efficiency: Traditional street lighting systems operate on fixed schedules or manual controls, resulting in lights being turned on even when they are not needed. The IoT-based system utilizes weather data to dynamically adjust the lighting levels, ensuring that lights are only activated when necessary. This reduces unnecessary energy consumption and lowers electricity costs.

2. Enhance Safety and Visibility: Different weather conditions, such as rain, fog, or snow, affect visibility on the roads. The IoT-based system can automatically adjust the brightness and intensity of street lights based on real-time weather data. This ensures optimal lighting conditions for drivers, pedestrians, and cyclists, improving road safety during varying weather conditions.

3. Reduce Light Pollution: Excessive and uncontrolled lighting can contribute to light pollution, negatively impacting the environment and disrupting natural ecosystems. With the IoT-based system, street lights can be dimmed or adjusted based on weather conditions, reducing light pollution and promoting a more sustainable approach to outdoor lighting.

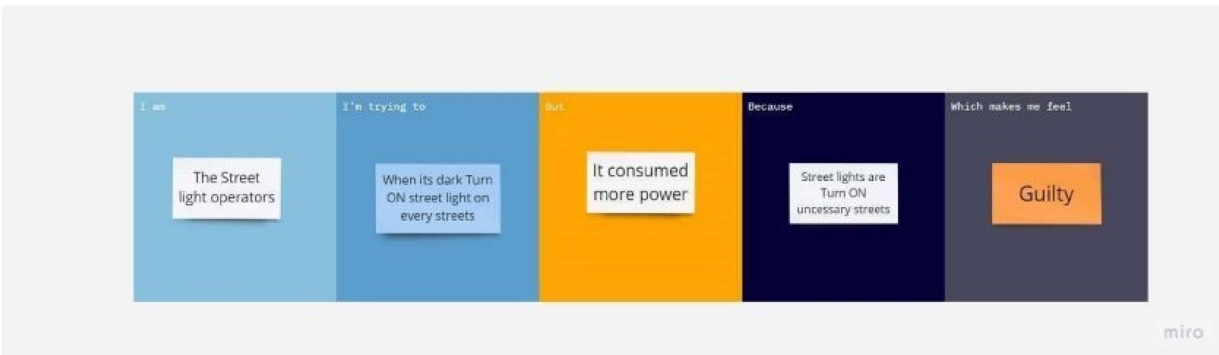
4. Optimize Maintenance: Monitoring the status and performance of street lights manually can be time-consuming and costly. The IoT-based system can provide remote monitoring and control capabilities, enabling proactive maintenance and rapid detection of faults or failures. This helps optimize maintenance schedules, reduces downtime, and improves the overall reliability of the street lighting system.

By implementing an IoT-based Weather Adaptive Street Lighting System, cities and municipalities can achieve significant energy savings, improve safety on the roads, reduce light pollution, and streamline maintenance operations. This innovative solution brings intelligence and efficiency to street lighting infrastructure, contributing to a smarter and more sustainable urban environment

2. IDEATION & PROPOSED SOLUTION :

Problem Statement Definition :

The existing street lighting systems lack intelligence and adaptability to real-time weather conditions, resulting in inefficient energy consumption, compromised visibility during varying weather conditions, increased light pollution, and inefficient maintenance processes. There is a need for an IoT-based Weather Adaptive Street Lighting System that addresses these challenges and provides an intelligent and adaptive solution for street ligh ting infrastructure.



Key Issues:

1. Inefficient Energy Consumption: Traditional street lighting systems operate on fixed schedules or manual controls, leading to lights being turned on even when not required. This results in unnecessary energy consumption and increased electricity costs.

2. Compromised Visibility: Different weather conditions such as rain, fog, or snow affect visibility on the roads. The lack of adaptive lighting systems results in suboptimal lighting conditions, compromising the safety of drivers, pedestrians, and cyclists.

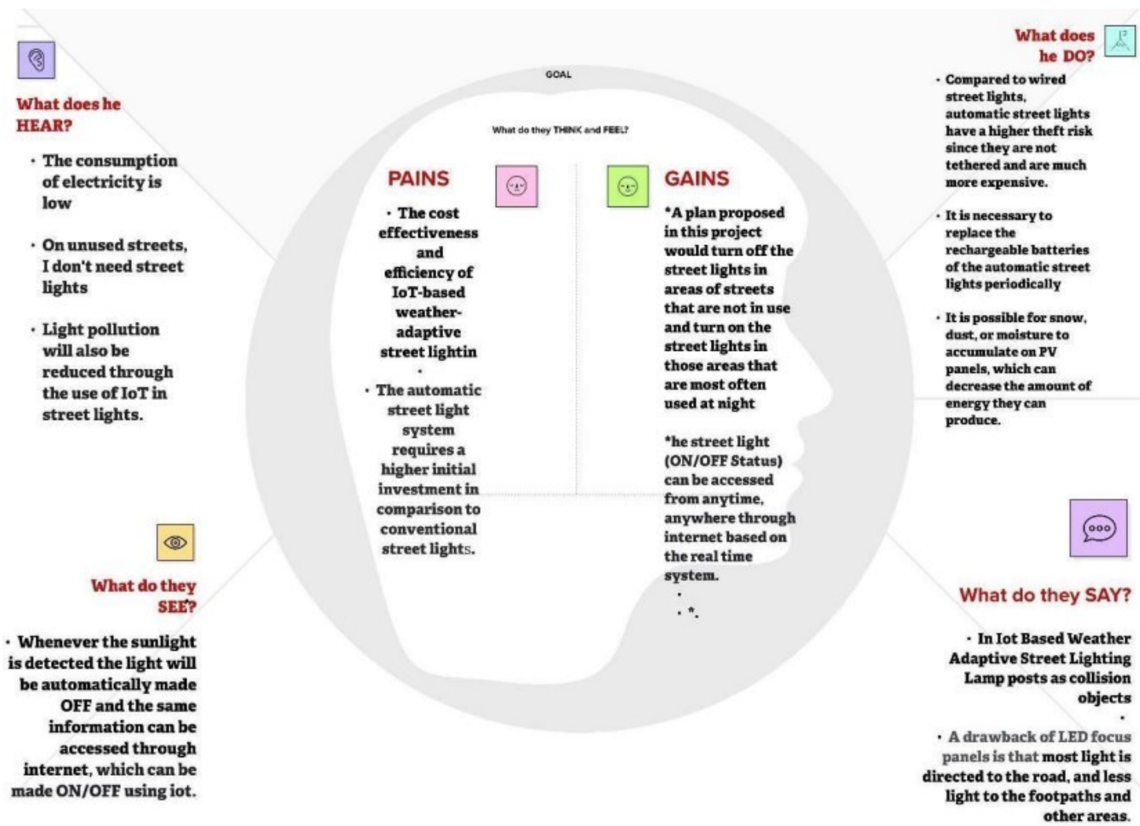
3. Light Pollution: Uncontrolled and excessive lighting contributes to light pollution, which has adverse effects on the environment and disrupts natural ecosystems. The absence of dynamic lighting control exacerbates light pollution issues.

4. Inefficient Maintenance: Manual monitoring and maintenance processes for street lighting systems are time-consuming and costly. The lack of remote monitoring and control capabilities hinders proactive maintenance and timely detection of faults or failures, leading to increased downtime and maintenance expenses.

Empathy Map Canvas :

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges

Empathy Map Canvas Diagram :



1. User: Maintenance Personnel

Says: "We need a street lighting system that is easy to maintain and manage."

Thinks: "How can we streamline the maintenance process and reduce manual interventions?"

Feels: Frustration when dealing with frequent maintenance issues and outdated technology.

Does: Performs routine maintenance checks, replaces faulty components, and manages the lighting system's operation.

2. User: Emergency Service Providers

Says: "We need well-lit streets to ensure efficient emergency response."

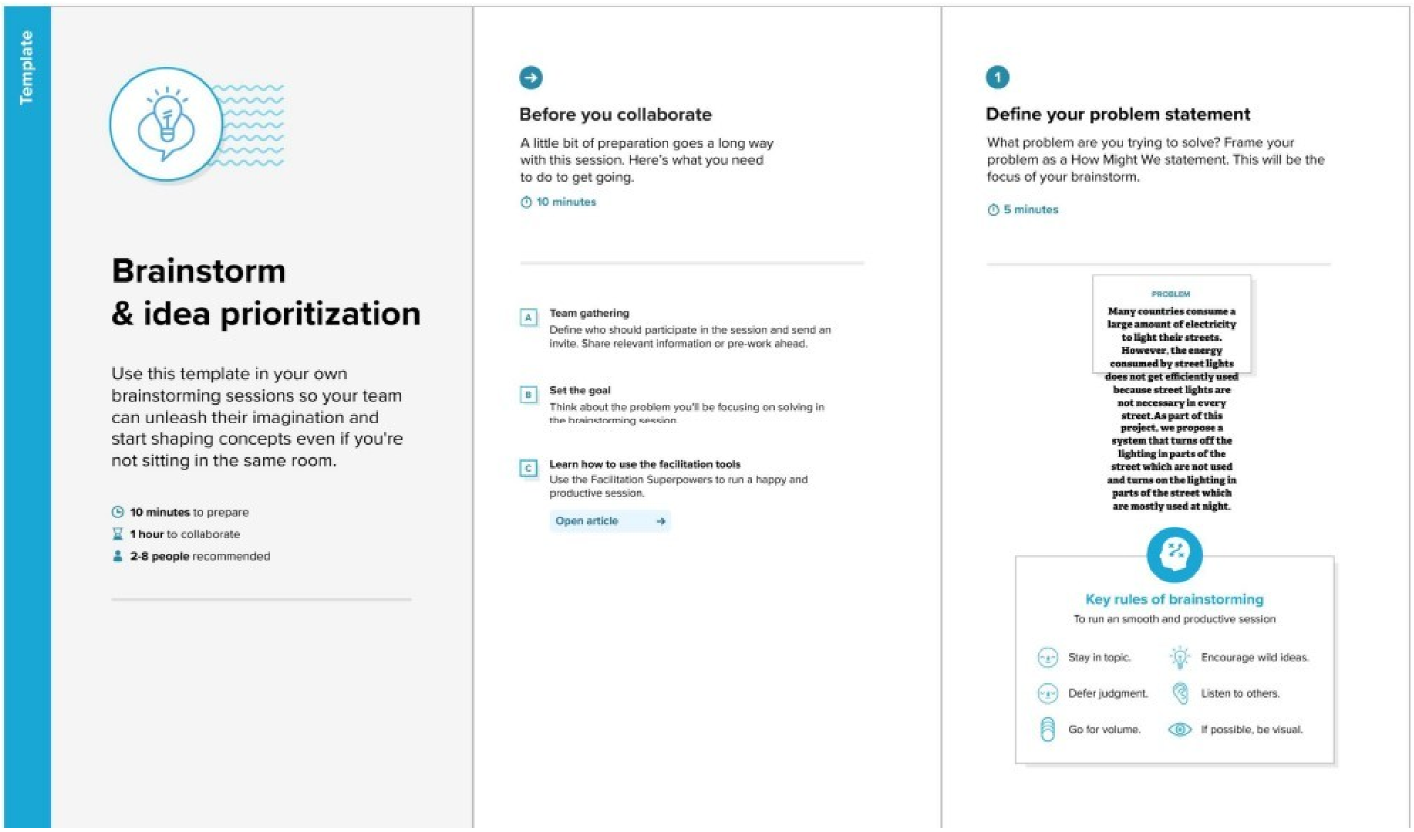
Thinks: "How can we ensure optimal visibility during emergencies, regardless of weather conditions?"

Feels: Anxious about delays caused by poor visibility or inadequate lighting.

Does: Collaborates with city officials to identify lighting needs for emergency situations, supports the implementation of adaptive lighting systems for enhanced response.

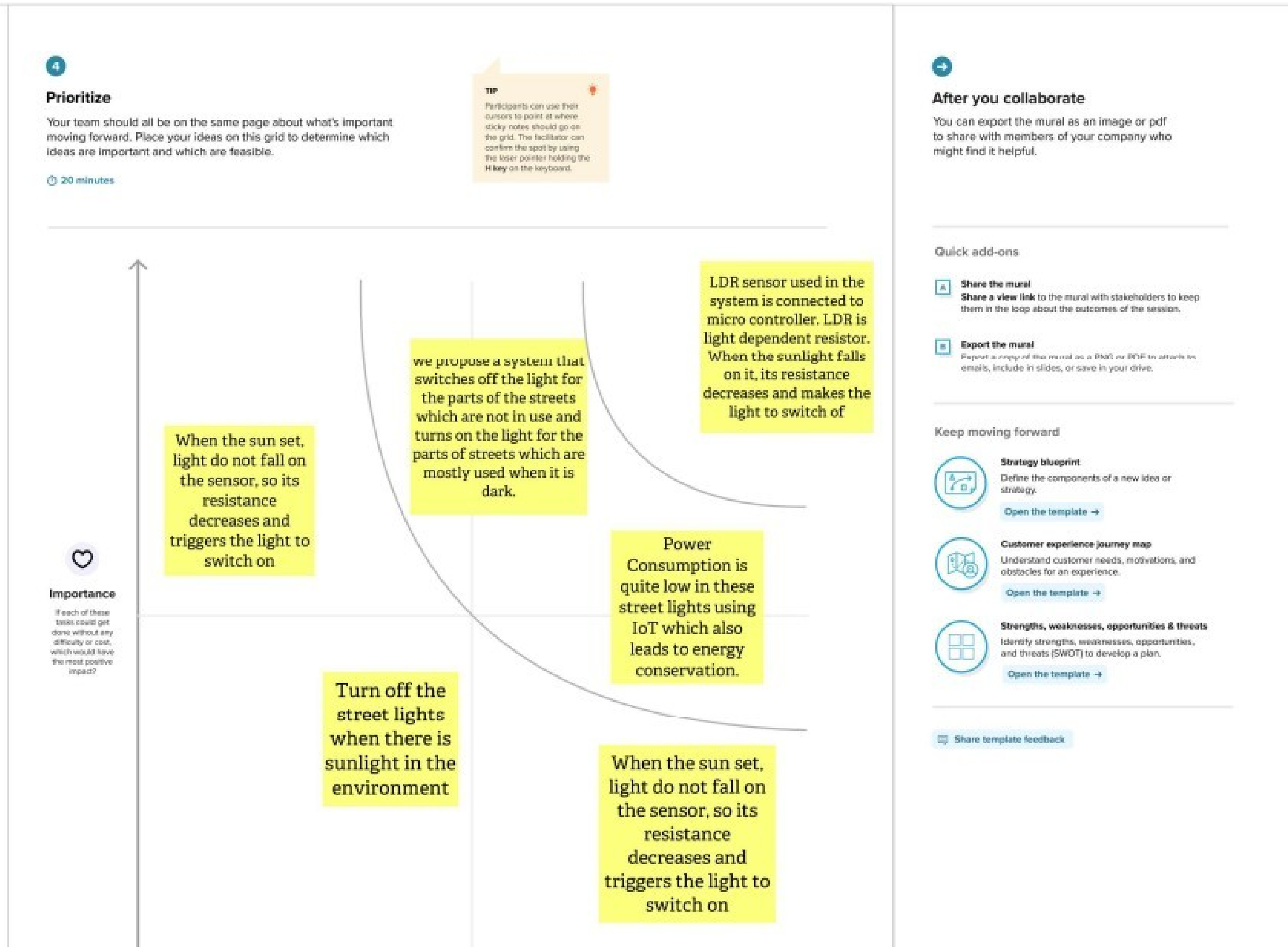
Ideation & Brainstorming :

Step-1: Team Gathering, Collaboration and Select the Problem Statement Diagram :



Step-2: Brainstorm, Idea Listing and Grouping :

Step-3: Idea Prioritization :



IDEATION :

1. Light Color Variation:

Explore the possibility of using different light colors based on weather conditions.

For example, use warm white lights during clear nights and cooler white lights during foggy or rainy conditions.

2. Pedestrian and Vehicle Detection:

Implement sensors or cameras to detect the presence of pedestrians or vehicles.

Increase lighting levels in areas with high pedestrian traffic or when vehicles are approaching.

3. Dynamic Light Pulsation:

Introduce pulsating light patterns during extreme weather conditions to grab attention.

This can help alert drivers and pedestrians to the presence of hazardous weather conditions.

Integration with Navigation Systems:

Collaborate with navigation or mapping applications to provide real-time lighting information.

Sync lighting adjustments with recommended routes to enhance safety and visibility.

4. Noise and Sound Detection:

Use sound sensors to detect noise levels or specific sounds (e.g., sirens, car horns) in the surroundings.

Increase lighting levels in response to high noise levels, indicating potential emergencies or disturbances.

5. User Customization:

Allow residents to customize their lighting preferences through a mobile app or web interface.

Offer adjustable brightness levels or personalization options based on individual needs.

6. Traffic Flow Optimization:

Analyze traffic patterns and adjust lighting levels accordingly to optimize traffic flow.

Increase lighting in congested areas or intersections during peak traffic hours.

7. Integration with Public Transportation Systems:

Collaborate with public transportation authorities to synchronize lighting with bus or tram schedules.

Increase lighting levels during arrival and departure times to enhance passenger safety

BRAINSTORMING :

1. Rain Sensor Integration:

Integrate rain sensors to detect rainfall intensity and adjust lighting accordingly.

Increase brightness during heavy rain to improve visibility for drivers and pedestrians.

2. Temperature Sensor Integration:

Use temperature sensors to adjust lighting levels based on temperature conditions.

Increase brightness during extremely cold weather to enhance safety on icy roads.

3. Wind Speed Detection:

Implement wind speed sensors to detect high winds and adjust lighting accordingly.

Increase brightness in windy conditions to aid navigation and alert users of potential hazards.

4. Fog and Mist Detection:

Use fog or mist sensors to detect low visibility conditions and adjust lighting levels accordingly.

Increase brightness during foggy conditions to improve visibility for drivers and pedestrians.

5. Snowfall Detection:

Integrate snowfall sensors to detect snow accumulation and adjust lighting levels accordingly.

Increase brightness during snowfall to ensure safe movement on the roads.

6. Machine Learning Algorithms:

Employ machine learning algorithms to analyze historical weather data and optimize lighting adjustments.

Continuously learn and adapt to weather patterns to make more accurate lighting predictions

Proposed Solution:

The proposed solution is to develop an IoT-based Weather Adaptive Street Lighting System that utilizes real-time weather data to dynamically adjust the lighting levels and optimize energy consumption. The system should be capable of automatically adapting to various weather conditions, ensuring optimal visibility and safety on the roads. Additionally, the system should incorporate features to reduce light pollution and enable remote monitoring and control for efficient maintenance operations.

By addressing these key issues, the IoT-based Weather Adaptive Street Lighting System will contribute to energy efficiency, enhance safety and visibility, minimize light pollution, and optimize maintenance processes.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Most countries consume a large amount of electric power for lighting their streets. The electrical energy consumed by street lights is not efficiently used due to the fact that street lamps are not essential on every street every time. Lights are switched off for parts of the street not in use, and turned on for parts of the street that are heavily used at night. Through a mobile application, users can manually control the street's lighting. The system is connected to a mobile application over the cloud.
2.	Idea / Solution description	**Ideas :** Panic button is fixed at the street light to trigger the system by raising an alarm signal at the nearby police station / **Solution description :** *A Cloud account is maintained to store the footages of camera whenever the panic button is pressed by the people. * Ip65 camera is installed to capture the entire movements of people moving on the particular street. * Panic button is present at the reachable height i.e., 5 feet of human beings. If a person who is in need of emergency can press the button, which raises an alarm at the nearby police station. Immediately the officer can check his account to get rid of the happening at the road
3.	Novelty / Uniqueness	**Novelty :** It is crucial to diminish the essentialness use and development the bundle level of this module to make it versatile. This will be the reason for mixing of the makers' future work, which besides joins playing out an irrefutably down to business and controlled botch examination, testing the contraption emanated

		light with change systems, for instance, PWM, testing with more people, and testing in multidoor conditions. The full scale structure should in like way join an IoT interface, which will require an extra module. **Uniqueness :** • To provide energy consumption. • To prevent energy wastage. • To ensure security to the people, especially to prevent women harassment.
4.	Social Impact / Customer Satisfaction	**Social Impact :** Average Road Luminance Lav in Cd/m ***Brightness (Road Luminance) depends on the light distribution of the luminaire, the lumen output of the luminaire, the installation design of the street lighting, and the reflective properties of the road surface. The higher the brightness level, the better the lighting effect.*** **Customer Satisfaction :** reduced energy cost and usage with flexible dimming controls; increased pedestrian satisfaction through improved safety measures; lowered repair and maintenance costs with the monitoring software; reduced carbon emissions and light pollution; These are more benefits of this IoT weather adaptive street lighting system.So,the peoples are fully Satisfied.
5.	Business Model (Revenue Model)	**The annual revenue from smart street lighting will grow by 31% between 2018 and 2026, according to a new report issued by ABI Research.** The reports states that the global smart lighting market will grow 10-fold to reach $1.7 billion in annual revenue by 2026.

3. REQUIREMENT ANALYSIS :

Main Factors in the Street Lighting Design Scheme

- Luminance Level Should be Proper. ...
- Luminance Uniformity must be Achieved. ...
- Degree of Glare Llimitation is always taken into Design Scheme. ...
- Lamp Spectra for Visual Sharpness depends on the Proper Luminaries. ...
- Effectiveness of Visual Guidance is also an important factor.

Functional requirement :

1. Weather Data Integration: The system should integrate with reliable weather data sources to receive real-time updates on weather conditions.

2. Sensor Integration: The system should incorporate various sensors to detect weather parameters such as rain, fog, snow, temperature, wind speed, and humidity.

3. Adaptive Lighting Control: The system should adjust the brightness levels of street lights based on detected weather conditions.

*****It should dynamically increase or decrease lighting intensity to optimize visibility and energy efficiency.

4. Communication and Connectivity: The system should establish reliable connectivity to receive weather data and transmit control signals to the street lights.

It should support communication protocols that enable seamless integration with other devices and systems.

5. Real-time Data Processing: The system should process weather data in real-time to ensure timely adjustments to lighting levels.

It should analyze data efficiently and make accurate decisions for adaptive lighting control.

6. Energy Optimization: The system should optimize energy usage by adjusting lighting levels based on weather conditions and occupancy.

It should incorporate energy-saving technologies, such as LED lights and motion sensors, to minimize energy consumption

7. User Interface: The system should provide a user-friendly interface for city officials to monitor and control the lighting system.

It should display real-time weather information, lighting status, and allow manual overrides if necessary.

8. Maintenance and Diagnostics: The system should include features for remote monitoring and diagnostics of street lights.

It should detect and report faults or performance issues to enable timely maintenance and minimize downtime.

9. Security and Privacy: The system should ensure the security of data transmission and protect user privacy.

It should implement encryption protocols and access controls to prevent unauthorized access to the system.

10. Integration with Emergency Services: The system should integrate with emergency services to prioritize lighting during emergencies.

It should enable seamless coordination with emergency response systems for optimal lighting adjustments.

Non-Functional requirements:

1. Reliability:

The system should operate reliably under varying weather conditions, ensuring consistent performance and accurate lighting adjustments.

2. Scalability:

The system should be scalable to accommodate a large number of street lights and handle increasing data volumes as the city expands.

3. Availability:

The system should have high availability, minimizing downtime and ensuring that the lighting system is operational at all times.

4. Performance:

The system should respond quickly to changes in weather conditions and adjust lighting levels in a timely manner.

It should process data efficiently, ensuring minimal latency and optimal system performance.

5. Security:

The system should implement robust security measures to protect against unauthorized access, data breaches, and cyber threats.

It should ensure the integrity and confidentiality of data transmitted and stored within the system.

6. Privacy:

The system should respect user privacy by securely handling personal data and adhering to applicable privacy regulations.

7. Interoperability:

The system should support interoperability with existing infrastructure, devices, and systems to facilitate seamless integration.

8. Usability:

The system should have a user-friendly interface that is intuitive and easy to navigate for city officials and maintenance personnel.

It should provide clear and meaningful visualizations of data and allow for efficient configuration and monitoring.

9. Maintainability:

The system should be designed for ease of maintenance, allowing for efficient troubleshooting, repairs, and upgrades.

It should have proper documentation and provide diagnostic tools to aid maintenance personnel.

10. Compliance:

The system should comply with relevant standards, regulations, and industry best practices related to lighting, energy efficiency, and IoT.

4. PROJECT DESIGN :

1. System Architecture:

The IoT-based weather adaptive street lighting system consists of the following components:

a. Sensors: Weather sensors such as temperature, humidity, rainfall, and ambient light sensors will be deployed at strategic locations along the street. These sensors will collect real-time weather data.

b. IoT Gateway: An IoT gateway device will be responsible for collecting data from the weather sensors and transmitting it to the cloud platform securely. It will also receive control commands from the cloud platform to adjust street lighting levels.

c. Cloud Platform: A cloud-based platform will receive and process the weather data from the IoT gateway. It will analyze the data and send appropriate control commands to the IoT gateway for adjusting street lighting levels.

d. Street Lighting Controllers: Each street lighting pole will be equipped with a lighting controller that receives control commands from the IoT gateway. The controllers will adjust the brightness levels of the street lights based on the received commands.

2. System Workflow:

a. Data Collection: The weather sensors placed along the street will continuously collect weather data such as temperature, humidity, rainfall, and ambient light levels.

b. Data Transmission: The IoT gateway will securely transmit the collected weather data to the cloud platform for analysis.

c. Data Analysis: The cloud platform will analyze the weather data to determine the appropriate lighting levels based on predefined rules and algorithms. Factors like rain, fog, or low light conditions may trigger different lighting levels.

d. Control Command Generation: The cloud platform will generate control commands based on the analysis results. These commands will be sent to the IoT gateway.

e. Lighting Adjustment: The IoT gateway will receive the control commands and transmit them to the respective street lighting controllers. The controllers will adjust the brightness levels of the street lights accordingly.

3. Key Features and Functionalities:

a. Adaptive Lighting: The system will dynamically adjust the street lighting levels based on real-time weather conditions. It will automatically increase or decrease the brightness of street lights as per the analyzed data.

b. Energy Efficiency: By adapting lighting levels based on weather conditions, the system will optimize energy consumption. It will reduce unnecessary lighting during favorable weather conditions and increase lighting during adverse weather conditions.

c. Safety Enhancement: The system will improve safety by ensuring appropriate lighting levels in different weather conditions. It will provide better visibility for pedestrians and drivers, reducing the risk of accidents.

d. Remote Monitoring and Control: The cloud platform will enable remote monitoring of the street lighting system. It will provide real-time updates on weather conditions, lighting status, and energy consumption. Authorized personnel will have the ability to remotely adjust lighting settings if required.

e. Data Analytics and Insights: The system will gather and analyze historical data to generate insights about energy consumption patterns, weather trends, and system performance. This information can be used for further optimization and decision-making.

Data Flow Diagrams :

Figure 1.

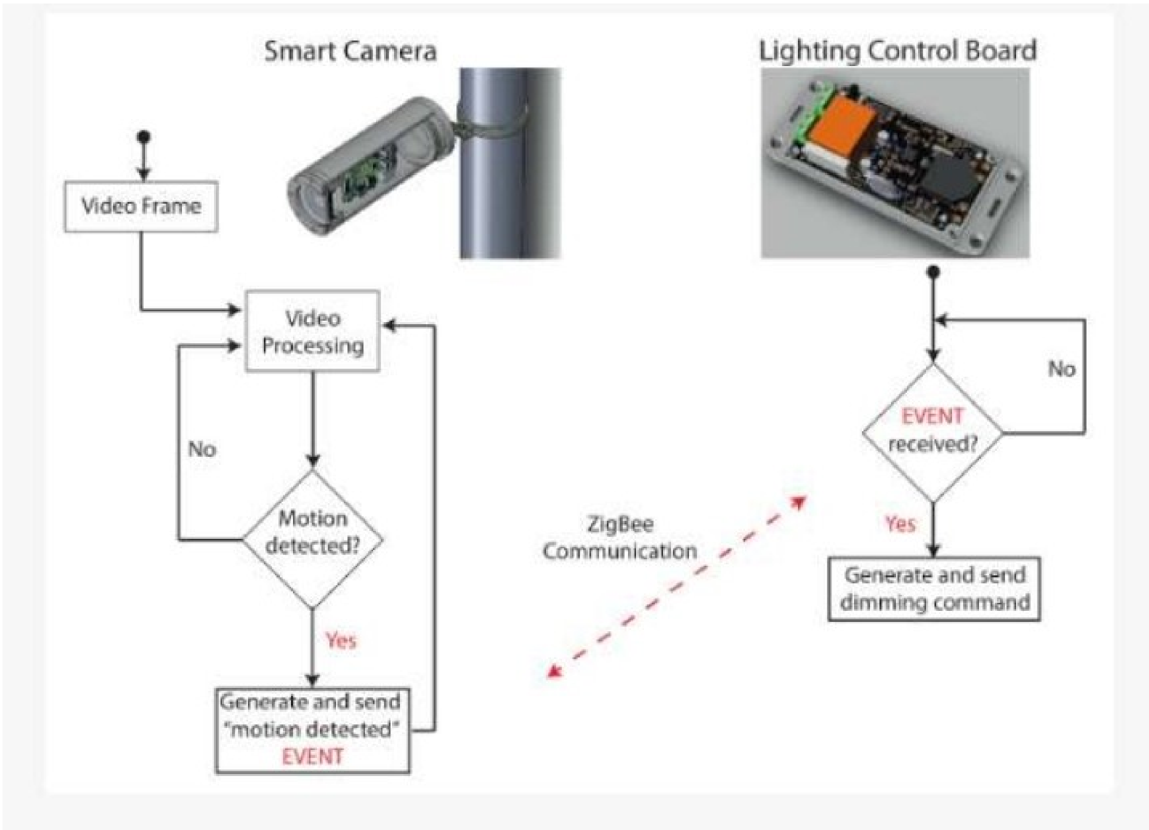
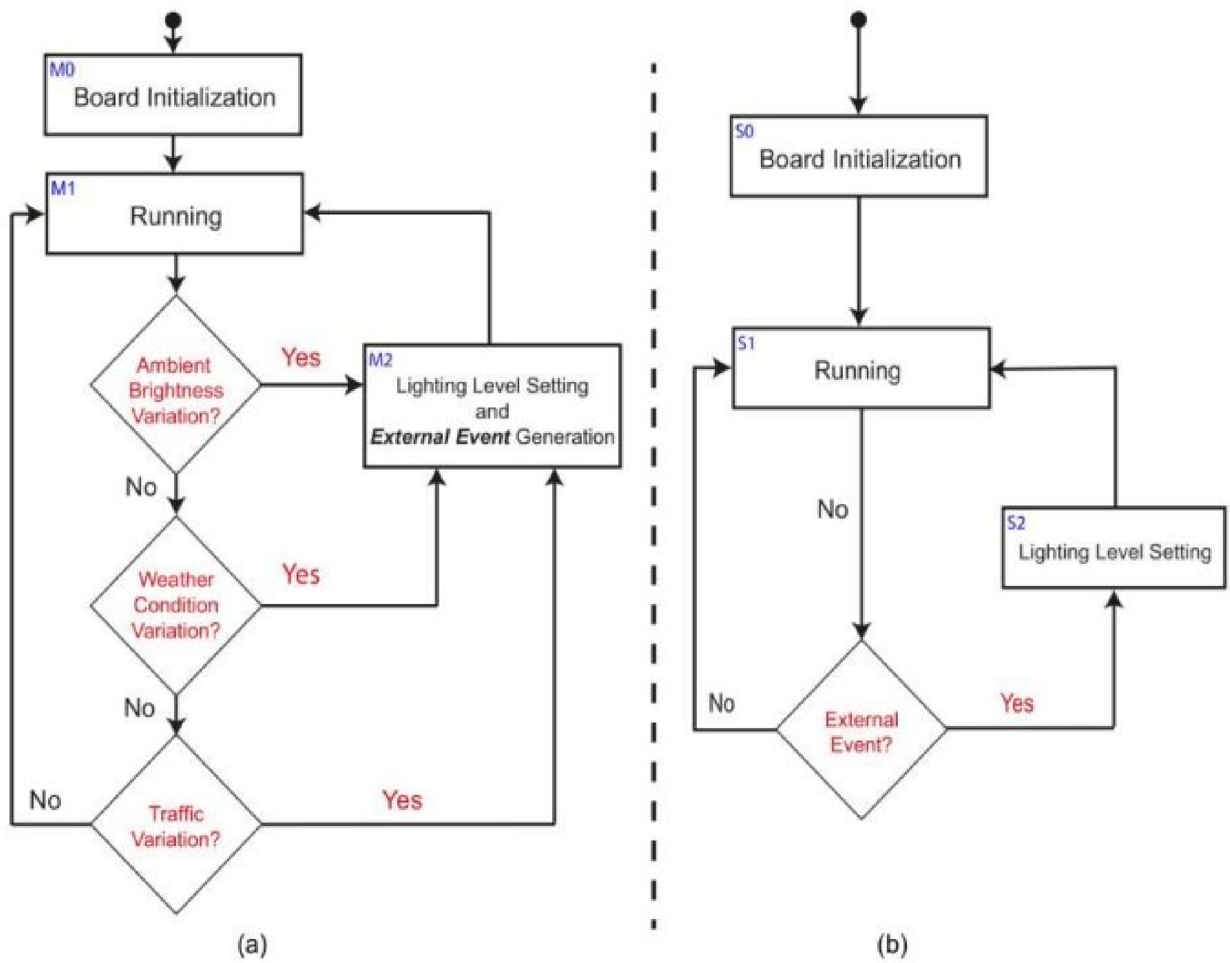


Figure 2.



M0
Board Initialization
M1
Running
Ambient Brightness Variation?
Yes
M2
Lighting Level Setting and *External Event* Generation
No
Weather Condition Variation?
Yes
No
No
Traffic Variation?
Yes
(a)
S0
Board Initialization
S1
Running
No
S2
Lighting Level Setting
No
External Event?
Yes
(b)

Solution & Technical Architecture :

Solution Architecture :

Figure 1.

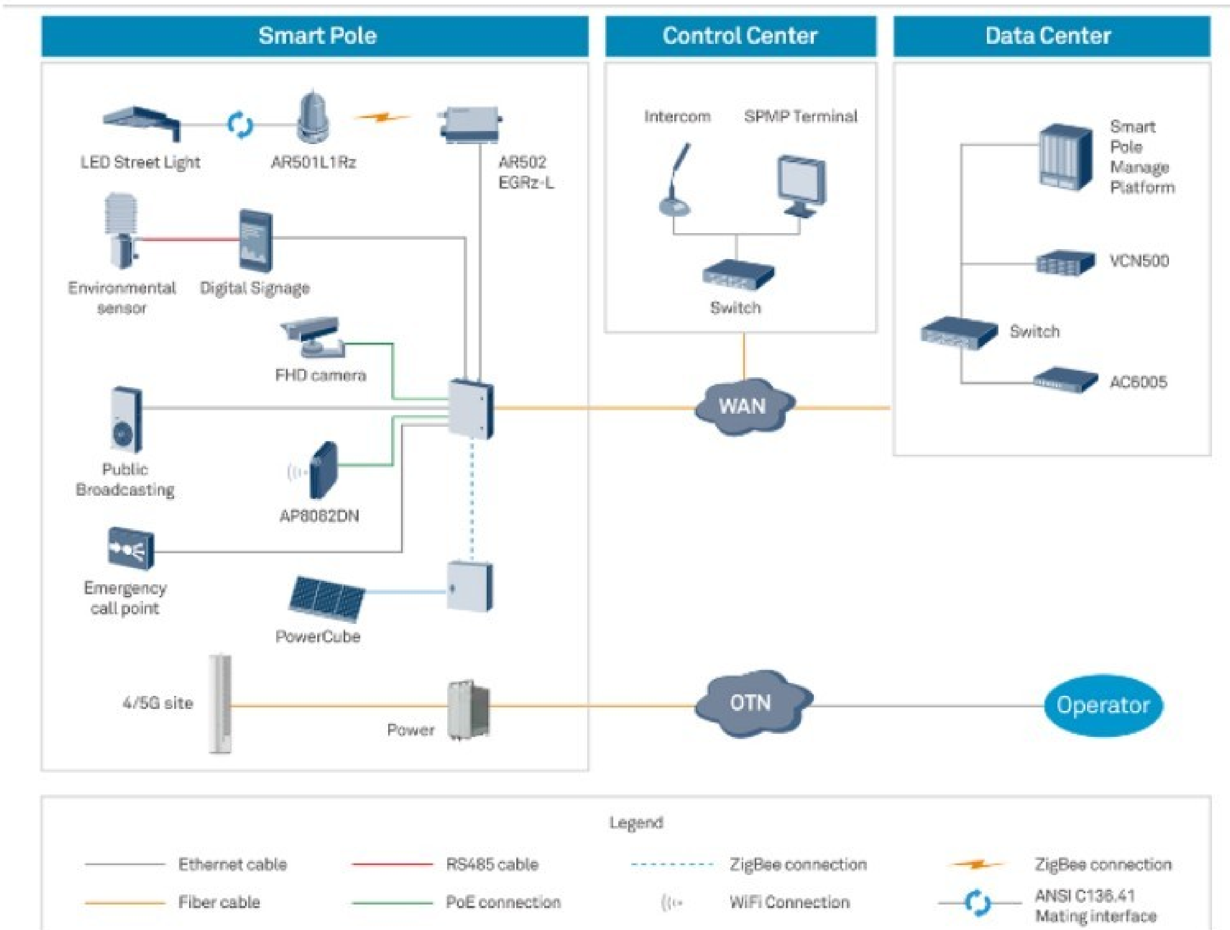
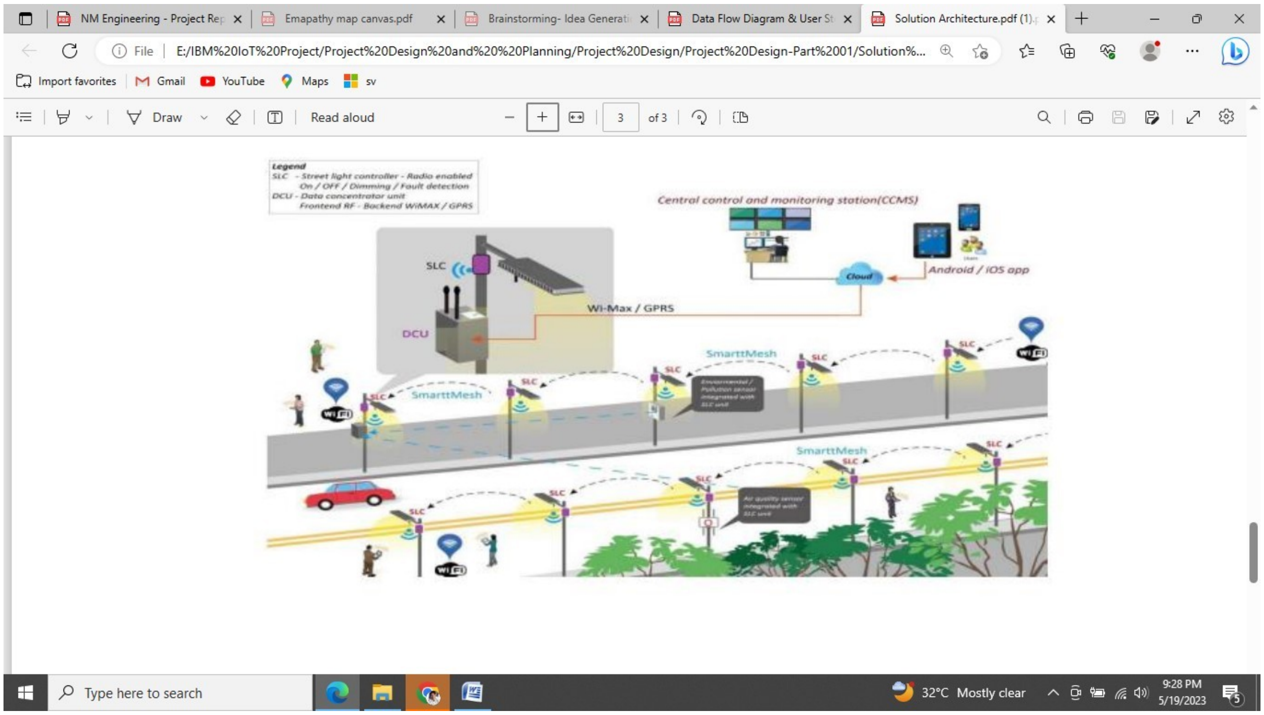


Figure : 2

Energy Efficiency Of IoT



1. IoT Gateway:

Acts as a bridge between the sensors and the cloud platform.

Collects data from weather and light sensors and transmits it securely to the cloud.

2.Cloud Platform:

Receives and processes data from the IoT gateway.

Performs advanced analytics, including weather prediction and lighting control algorithms.

Stores historical data for analysis and future enhancements.

Provides a centralized platform for managing the entire system.

3. Rules Engine: Executes predefined rules and algorithms based on weather conditions, user preferences, and lighting policies.

Determines the appropriate lighting settings for different scenarios.

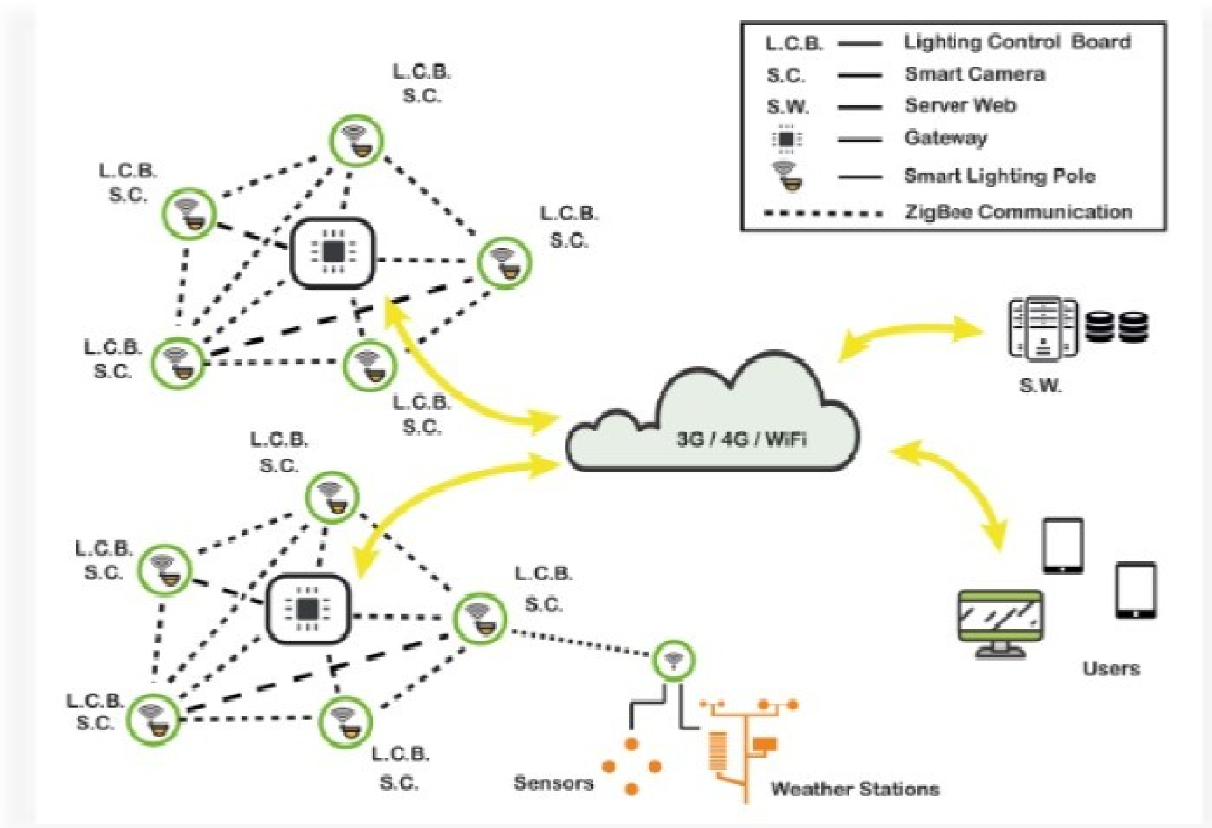
4. Street Lighting Control System:

Interfaces with the cloud platform and receives commands from the rules engine.

Controls individual streetlights, adjusting their intensity or turning them on/off as needed.

Monitors the status of streetlights and reports any operational issues.

Technical Architecture :



1. Cloud or Local Server:

Receives and stores data from the IoT gateway.
Performs data processing, analytics, and decision-making.
Hosts the control and management software components.

2. Control and Management Software:

Includes various software components responsible for controlling the street lighting system and managing its operations.
Weather prediction and lighting control algorithms are implemented in this software.
Manages the communication between different system components.
May use technologies such as MQTT (Message Queuing Telemetry Transport) for data communication.

User Stories :

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low
		USN-4	As a user, I can register for the application through Gmail		Medium
	Login	USN-5	As a user, I can log into the application by entering email & password		High
	Dashboard				
Customer (Web user)	Weather Adaptive Street Lighting System	USN-1	As a web user, I want to access a user-friendly interface for the IoT-based Weather Adaptive Street Lighting System, allowing me to monitor and control the street lighting settings remotely.	*The system should have a web interface accessible to users. *The web interface should be user-friendly and intuitive. *The web interface should allow users to monitor and control street lighting settings remotely.	Medium

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority
		USN-2	As a web user, I want to receive regular updates and notifications from the IoT-based Weather Adaptive Street Lighting System, informing me of any system updates, maintenance schedules, or changes in lighting settings.	*The system should send regular updates and notifications to web users. *The updates and notifications should cover system updates, maintenance schedules, and changes in lighting settings. *The updates and notifications should be timely and informative	Low
Customer Care Executive	Weather Adaptive Street Lighting System	USN-1	As a customer care executive, I want the IoT-based Weather Adaptive Street Lighting System to provide a customer support portal, allowing me to assist users with any issues or inquiries related to the system.	*The system should have a customer support portal *The customer support portal should be accessible to customer care	Medium

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority
		USN-2	As a customer care executive, I want the IoT-based Weather Adaptive Street Lighting System to provide a customer support portal, allowing me to assist users with any issues or inquiries related to the system.	executives . *The customer support portal should enable assistance with system issues or inquirie The system should have a customer support portal. The customer support portal should be accessible to customer care executives. The customer support portal should enable assistance with system issues or inquirie	Medium

5. CODING & SOLUTIONING :

Feature 1:

Adaptive Lighting Control:
Use the weather data to determine the appropriate lighting level based on specific weather conditions.
Implement a function to adjust the brightness of street lights accordingly.

```
def adjust_lighting(weather_data):
  brightness = 0

  if weather_data["weather"][0]["main"] == "Rain":
    brightness = 70
  elif weather_data["weather"][0]["main"] == "Fog":
    brightness = 50
  elif weather_data["weather"][0]["main"] == "Snow":
    brightness = 60
  else:
    brightness = 100

  # Code to adjust the brightness of street lights based on the calculated brightness level

  return brightness.
```

2.Weather Monitoring: The system can gather real-time weather data from sensors or weather APIs to adjust the street lighting accordingly. For example, during rainy or foggy weather, the lighting intensity can be increased for better visibility
import requests

PYTHON CODE :

```
# Function to fetch weather data from an API
def get_weather_data():
  api_key = 'your_api_key'
  url = 'https://api.weather.com/data/2.5/weather?q=city_name&appid=' + api_key
```

```python
    response = requests.get(url)
    data = response.json()
    return data

# Example function to adjust lighting based on weather conditions
def adjust_lighting_based_on_weather(weather_data):
    if 'rain' in weather_data['weather']:
        # Increase lighting intensity for rainy weather
        set_lighting_intensity(80)
    elif 'fog' in weather_data['weather']:
        # Increase lighting intensity for foggy weather
        set_lighting_intensity(60)
    else:
        # Use default lighting intensity
        set_lighting_intensity(50)

# Function to set the lighting intensity
def set_lighting_intensity(intensity):
    # Code to control the street lighting intensity
    # ...

# Main program
weather_data = get_weather_data()
adjust_lighting_based_on_weather(weather_data)
```

Feature 2 : Energy efficiency :

LDR sensor is used for detecting the presence of surrounding light so that during the day time when sun is bright, the street light is switched off automatically. During the night time when there is no light, the LDR sends signal to microcontroller to turn on the street light.

C++ :

```cpp
#include <WiFi.h>//library for wifi
#include <PubSubClient.h>//library for MQtt
#define LED 5
#define LED2 4
```

```
#define LED3 2
int LDR = 32;
int LDRReading = 0;
int threshold_val = 800;
int lEDBrightness = 0;
int flag=0;

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength);

//-------credentials of IBM Accounts------

#define ORG "utln5h"//IBM ORGANITION ID
#define DEVICE_TYPE "streetlight"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "12345" //Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678"     //Token
String data3;
float h, t;


//-------- Customise the above values --------
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";// Server Name
char publishTopic[] = "iot-2/evt/Data/fmt/json";// topic name and type of event
perform and format in which data to be send
char subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT command
type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id



//-----------------------------------------------------------------------------
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined
client id by passing parameter like server id,portand wificredential
void setup()// configureing the ESP32
{
  Serial.begin(115200);

  pinMode(LED,OUTPUT);
  pinMode(LED2,OUTPUT);
  pinMode(LED3,OUTPUT);
  delay(10);
```

```
  Serial.println();
  wificonnect();
  mqttconnect();
}

void loop()// Recursive Function
{



  //PublishData(t, h);
  //delay(1000);

  /* LDRReading = analogRead(LDR);
  Serial.print("LDR READING:");
  Serial.println(LDRReading);

  if (LDRReading >threshold_val){
  lEDBrightness = map(LDRReading, 0, 1023, 0, 255);
  Serial.print("LED BRIGHTNESS:");
  Serial.println(lEDBrightness);

  analogWrite(LED, lEDBrightness);
  analogWrite(LED2, lEDBrightness);
  analogWrite(LED3, lEDBrightness);
  }
  else{
  analogWrite(LED, 0);
  analogWrite(LED2, 0);
  analogWrite(LED3, 0);
  }

  delay(300);*/

  if (!client.loop()) {
    mqttconnect();
  }
}



/*.....................................retrieving to
Cloud ............................................................... */
```

```
/*void PublishData(float temp, float humid) {
  mqttconnect();//function call for connecting to ibm*/
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  /*String payload = "{\"temperature\":";
  payload += temp;
  payload += "," "\"humidity\":";
  payload += humid;
  payload += "}";


  Serial.print("Sending payload: ");
  Serial.println(payload);


  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully upload data on the cloud
then it will print publish ok in Serial monitor or else it will print publish
failed
  } else {
    Serial.println("Publish failed");
  }

}  */
void mqttconnect() {
  if (!client.connected()) {
    Serial.print("Reconnecting client to ");
    Serial.println(server);
    while (!!!client.connect(clientId, authMethod, token)) {
      Serial.print(".");
      delay(500);
    }

     initManagedDevice();
     Serial.println();
  }
}
void wificonnect() //function defination for wificonnect
{
  Serial.println();
  Serial.print("Connecting to ");
```

```
  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish
the connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void initManagedDevice() {
  if (client.subscribe(subscribetopic)) {
    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  } else {
    Serial.println("subscribe to cmd FAILED");
  }
}

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{


  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);

  for (int i = 0; i < payloadLength; i++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
  }

  Serial.println("data: "+ data3);
  if(data3=="lighton1")
  {
Serial.println(data3);
digitalWrite(LED,HIGH);

  }

  else if(data3=="lightoff1")
  {
Serial.println(data3);
```

```
digitalWrite(LED,LOW);


  }
  else if(data3=="lighton2")
  {
Serial.println(data3);
digitalWrite(LED2,HIGH);

  }

  else if(data3=="lightoff2")
  {
Serial.println(data3);
digitalWrite(LED2,LOW);

  }
  else if(data3=="lighton3")
  {
Serial.println(data3);
digitalWrite(LED3,HIGH);

  }

  else if(data3=="lightoff3")
  {
Serial.println(data3);
digitalWrite(LED3,LOW);

  }
data3="";



}
```

* When its dark Turn ON street light on every streets.
* When its sunlight present,Turn OFF street light on every streets

6. RESULTS :

Huge reduction of energy and maintenance cost. increased public safety from improved lighting. safer traffic due to increased visibility of hazards; measurable environmental impact due to reduced energy consumption

Performance Metrics :

1. Energy Efficiency: This metric evaluates how effectively the system utilizes energy resources. It can be measured by monitoring the energy consumption of the street lighting system under different weather conditions and comparing it to a baseline scenario without adaptive lighting. The energy efficiency can be quantified in terms of energy savings achieved.

2. Lighting Intensity Control: This metric assesses the accuracy and responsiveness of the system in adjusting the lighting intensity based on weather conditions. It can be evaluated by comparing the actual lighting intensity with the desired intensity set by the system based on real-time weather data. The system should be able to achieve the desired lighting levels consistently and promptly.

3. Adaptability to Weather Changes: This metric measures how well the system adapts to rapid weather changes. It evaluates the system's ability to quickly detect and respond to changes in weather conditions, such as sudden rainfall or fog. The system should be able to adjust the lighting settings accordingly in a timely manner to maintain optimal visibility and safety.

4. Accuracy of Weather Data: This metric assesses the accuracy and reliability of the weather data used by the system. It can be evaluated by comparing the weather data collected by the system with data from reputable weather sources or weather stations. The system should have a low margin of error in weather data collection to ensure accurate lighting adjustments.

5. System Reliability: This metric measures the reliability and uptime of the street lighting system. It evaluates the system's ability to function consistently and without disruptions. Reliability can be measured by monitoring the occurrence of system failures, downtime, or malfunctions and assessing the system's availability over a specific period.

6. User Experience: This metric focuses on user satisfaction and ease of use. It can be evaluated through user surveys or feedback regarding the convenience, intuitiveness, and effectiveness of the remote monitoring and control functionalities. User experience metrics can help identify areas for improvement and enhance the overall usability of the system.

7. Maintenance and Cost Efficiency: This metric assesses the ease of maintenance and cost-effectiveness of the system. It can be evaluated by analyzing the frequency and cost of maintenance activities required for the street lighting system. A well-designed system should minimize the need for frequent maintenance, reduce maintenance costs, and ensure long-term sustainability

7. ADVANTAGES & DISADVANTAGES :

ADVANTAGES :

1. Energy Efficiency:
By dynamically adjusting the lighting intensity based on real-time weather conditions, the system can significantly improve energy efficiency. It reduces energy waste by lowering lighting intensity during periods of sufficient natural light and increasing it when visibility is compromised due to rain, fog, or other weather conditions. This results in substantial energy savings and contributes to sustainability efforts.

2. Cost Savings: The energy efficiency achieved by the IoT-based system translates into cost savings. By reducing energy consumption, municipalities and organizations can lower their electricity bills associated with street lighting. The system's adaptive capabilities help optimize resource allocation, ensuring that lighting is used only when necessary, reducing unnecessary costs.

3.Improved Visibility and Safety: The system enhances visibility on the streets by automatically adjusting the lighting based on weather conditions. During inclement weather like rain or fog, the lighting intensity is increased, enhancing visibility for pedestrians and drivers. This improves safety by reducing accidents and increasing overall security in public spaces.

4. Environmental Impact: The energy-saving features of the IoT-based system contribute to a reduced carbon footprint and environmental impact. By optimizing energy consumption, the system helps in reducing greenhouse gas emissions associated with electricity generation. This aligns with sustainability goals and promotes environmentally responsible practices.

5. Flexibility and Adaptability: The system offers flexibility and adaptability to changing weather conditions. It can quickly and automatically adjust the lighting settings based on real-time data, ensuring that the lighting is always optimized for the prevailing conditions. This flexibility allows the system to cater to different climates and weather patterns, making it suitable for various geographic locations.

6.Remote Monitoring and Control: The IoT capabilities of the system enable remote monitoring and control functionalities. Administrators and operators can monitor the system's performance, receive alerts or notifications for maintenance or faults, and remotely control the lighting settings. This improves operational efficiency, reduces the need for manual interventions, and facilitates proactive maintenance.

7. Data-Driven Insights: The system generates valuable data on weather patterns, energy consumption, and lighting performance. This data can be analyzed to gain insights into energy usage patterns, identify trends, and optimize system performance further. Data-driven insights enable evidence-based decision-making for maintenance planning, infrastructure improvements, and energy management.

8. Scalability and Expandability: The IoT-based system is highly scalable and expandable. It can be easily integrated with additional sensors, devices, or smart city infrastructure as needed. This scalability allows for future enhancements and the integration of emerging technologies, making it a future-proof solution.

DISADVANTAGES :

1. Initial Setup and Infrastructure Costs: Implementing an IoT-based system requires upfront investment in infrastructure, including sensors, communication devices, and network infrastructure. The initial setup costs can be higher compared to traditional street lighting systems. Additionally, ongoing maintenance and upgrades may be necessary, adding to the overall expenses.

2. Reliance on Technology and Connectivity: The IoT-based system heavily relies on technology and connectivity. If there are issues with the network or communication infrastructure, such as power outages or connectivity disruptions, it may affect the system's functionality. Dependence on technology introduces a certain level of vulnerability and the need for backup systems or contingency plans.

3. Data Security and Privacy Concerns: IoT systems generate and process a significant amount of data. This data may include sensitive information about weather conditions, lighting patterns, and even user behavior. Ensuring data security and protecting user privacy becomes crucial. Adequate measures must be implemented to safeguard data, prevent unauthorized access, and comply with privacy regulations.

4. Complexity and Technical Expertise: Developing and managing an IoT-based system requires technical expertise and knowledge. It involves understanding various technologies, programming, data analysis, and integration of different components. Organizations may need to invest in training or hire skilled personnel to handle the complexity of the system effectively.

5. Compatibility and Interoperability Challenges: Integrating an IoT-based street lighting system with existing infrastructure or legacy systems can be challenging. Compatibility issues may arise due to different communication protocols, hardware dependencies, or system architectures. Ensuring seamless interoperability with other smart city components may require additional effort and customization.

6. Environmental Factors and Limitations: Despite the system's adaptability, certain environmental factors can still pose challenges. For example, heavy rainfall or extreme weather conditions might affect the reliability and accuracy of sensors, leading to incorrect lighting adjustments. External factors like obstructions or reflections may impact the sensor readings, affecting the system's performance.

7. Learning Curve and User Acceptance: Introducing a new technology like IoT-based street lighting may involve a learning curve for users and stakeholders. It may take time for operators, maintenance personnel, and administrators to become familiar with the system's functionalities and adapt to new processes. Ensuring user acceptance and providing proper training and support are essential.

(This disadvantages are Possibilities to implemented and Changed to Disadvantage to advantage)

8. CONCLUSION :

In conclusion, the IoT-based weather adaptive street lighting system offers a range of advantages that contribute to energy efficiency, improved safety, and enhanced functionality compared to traditional street lighting systems. By dynamically adjusting the lighting intensity based on real-time weather conditions, the system optimizes energy usage, resulting in cost savings and reduced environmental impact.

The adaptability of the system to changing weather conditions ensures optimal visibility and safety on the streets, reducing accidents and increasing overall security. The remote monitoring and control capabilities enable efficient management and maintenance of the system, allowing for proactive interventions and minimizing downtime.

Data-driven insights generated by the system provide valuable information for optimizing energy consumption, identifying trends, and making informed decisions regarding maintenance and infrastructure planning. The scalability and expandability of the system make it adaptable to future technologies and smart city integration.

However, it is essential to consider the potential disadvantages of the system, such as initial setup costs, data security concerns, compatibility challenges, and the need for technical expertise. Addressing these challenges through proper planning, risk assessment, and implementation strategies can mitigate the drawbacks and maximize the benefits of the system.

Overall, the IoT-based weather adaptive street lighting system represents a significant advancement in street lighting technology, offering energy efficiency, improved safety, remote control capabilities, and data-driven insights. It has the potential to contribute to sustainable and smart city initiatives, creating more efficient and safer urban environment

9. FUTURE SCOPE :

1. Artificial Intelligence and Machine Learning: Integration of artificial intelligence (AI) and machine learning (ML) algorithms can further enhance the system's capabilities. AI algorithms can learn from historical and real-time data to optimize lighting adjustments based on complex patterns and correlations. ML techniques can be employed to improve the accuracy of weather predictions and enhance the system's adaptability to changing weather conditions.

2. Predictive Maintenance: Implementing predictive maintenance techniques can enable proactive maintenance and reduce downtime. By analyzing real-time and historical data, the system can predict when specific components or lighting fixtures may require maintenance or replacement. Predictive maintenance helps optimize maintenance schedules, extend equipment lifespan, and reduce operational disruptions.

3. Smart City Integration: Integrating the IoT-based street lighting system with other smart city components can create a more comprehensive and interconnected urban environment. For example, integrating with smart traffic management systems can enable coordinated lighting adjustments based on traffic patterns, reducing congestion and improving road safety. Integration with smart parking systems can optimize lighting in parking areas based on occupancy levels.

4. Advanced Sensor Technologies: Advancements in sensor technologies can further enhance the accuracy and reliability of the system. For example, the integration of advanced visibility sensors can provide more precise information about visibility conditions, enabling finer-grained adjustments to lighting intensity. Integration of environmental sensors, such as air quality sensors, can contribute to a holistic approach to urban sustainability.

5. Human-Centric Lighting: Future developments may focus on human-centric lighting, which takes into account human circadian rhythms and preferences. The system can adjust lighting color temperature and intensity to align with natural lighting patterns, promoting well-being and improving human comfort and productivity.

6. Integration with Smart Grids and Energy Storage: Integrating the system with smart grids and energy storage technologies can enable more efficient energy management. The system can leverage renewable energy sources and optimize energy storage to further reduce reliance on the grid and maximize energy savings.

7. Data Analytics and Insights: Continuous advancements in data analytics techniques can unlock deeper insights from the vast amount of data generated by the system. Advanced analytics can identify long-term trends, energy consumption patterns, and provide valuable information for urban planning, infrastructure optimization, and policy-making.

8. Connectivity and 5G Networks: The evolution of communication networks, particularly the widespread deployment of 5G, can enhance the connectivity and responsiveness of the system. Faster and more reliable communication networks can enable real-time data exchange, seamless remote monitoring, and control of the street lighting system.

9. Integration with Autonomous Vehicles: As autonomous vehicles become more prevalent, integrating the IoT-based lighting system with vehicle-to-infrastructure communication can enhance road safety. The system can communicate with autonomous vehicles to provide real-time lighting adjustments based on the vehicle's location and weather conditions, optimizing visibility for safe navigation.

10. APPENDIX :

Source Code :

Implementing the entire source code for an IoT-based weather adaptive street lighting system is beyond the scope of a text-based conversation. However, I can provide you with a high-level overview of the components and technologies typically involved in building such a system.

Hardware Components:

Microcontroller or Single-Board Computer (e.g., Arduino, Raspberry Pi)
Sensors (e.g., light sensor, rain sensor, temperature and humidity sensor)
Communication module (e.g., Wi-Fi, Ethernet, LoRa, GSM)

Software Components:
Firmware for the microcontroller or single-board computer to handle sensor data acquisition and communication.

Weather data retrieval: You can use an API or a web scraping technique to fetch real-time weather data from a reliable weather service provider.

Decision-making algorithm: Develop an algorithm that takes into account the received weather data and determines the appropriate lighting intensity based on predefined rules or thresholds.

Control logic: Implement the logic to control the street lighting system based on the decision made by the algorithm. This can involve adjusting the intensity of individual lights or groups of lights.
User interface: Develop a user interface (UI) for monitoring and controlling the system. This can be a web-based interface or a mobile application that allows users to view system status, adjust settings, and receive notifications.

It's important to note that the actual implementation details can vary depending on the chosen hardware, programming language, and specific requirements of the project. It's recommended to consult appropriate documentation, tutorials, and resources related to the selected hardware and software components to guide you through the implementation process.

Additionally, you may consider leveraging existing open-source projects or frameworks that provide a foundation for building IoT-based systems. These resources can provide code examples, libraries, and APIs that can accelerate your development process and ensure a more robust and reliable implementation.

GitHub & Project Video Demo Link :

GitHub :

https://github.com/naanmudhalvan-SI/IBM--15721-1681800394

Project Video Demo Link:

https://youtu.be/uIgb6hPF9lw

Wokwi link:

https://wokwi.com/projects/365177028056371201