Project Report Format

Weather Adaptive Street Lighting System

1. INTRODUCTION

1.1 Project Overview

1.2 Purpose

2. IDEATION & PROPOSED SOLUTION

2.1 Problem Statement Definition

2.2 Empathy Map Canvas

2.3 Ideation & Brainstorming

2.4 Proposed Solution

3.REQUIREMENT ANALYSIS

3.1 Functional requirement

3.2 Non-Functional requirements

4. PROJECT DESIGN

4.1 Data Flow Diagrams

4.2 Solution & Technical Architecture

4.3 User Stories

5.CODING & SOLUTIONING (Explain the features added in the project along with code)

5.1 Feature 1

5.2 Feature 2

5.3 Database Schema (if Applicable)

6. RESULTS

6.1 Performance Metrics

7. ADVANTAGES & DISADVANTAGES

INTRODUCTION

1.1 Project Overview

The IoT Based Weather Adaptive Street Lighting System is a project aimed at developing an intelligent street lighting system that adjusts its brightness based on real-time weather conditions. By leveraging IoT technology, the system can dynamically adapt to the changing environmental factors and optimize energy consumption.

1.2 Purpose

The purpose of this project is to create a smart street lighting solution that enhances energy efficiency, reduces operational costs, and provides improved visibility and safety for pedestrians and drivers. By integrating weather data into the lighting system, it can automatically adjust the brightness levels of the streetlights, ensuring optimal illumination based on weather conditions.

IDEATION & PROPOSED SOLUTION

2.1 Problem Statement Definition

The existing street lighting systems are typically static and do not consider environmental factors such as weather conditions. This leads to inefficient energy usage and inadequate lighting during certain weather conditions. The problem statement is to design an intelligent street lighting system that can dynamically adjust the brightness levels based on real-time weather data.

2.2 Empathy Map Canvas

The empathy map canvas helps us understand the needs, desires, and pain points of the stakeholders involved in the street lighting system. It includes stakeholders such as pedestrians, drivers, municipality authorities, and energy providers.

2.3 Ideation & Brainstorming

During the ideation and brainstorming phase, various solutions and ideas were generated to address the problem. This involved considering different IoT technologies, weather APIs, and lighting control mechanisms.

2.4 Proposed Solution

The proposed solution is to develop a system that utilizes IoT devices, such as weather sensors and streetlight controllers, to monitor weather conditions and adjust the brightness of streetlights accordingly. The system will leverage weather APIs to fetch real-time weather data and use it to determine the appropriate lighting levels.

3.REQUIREMENTS ANALYSIS

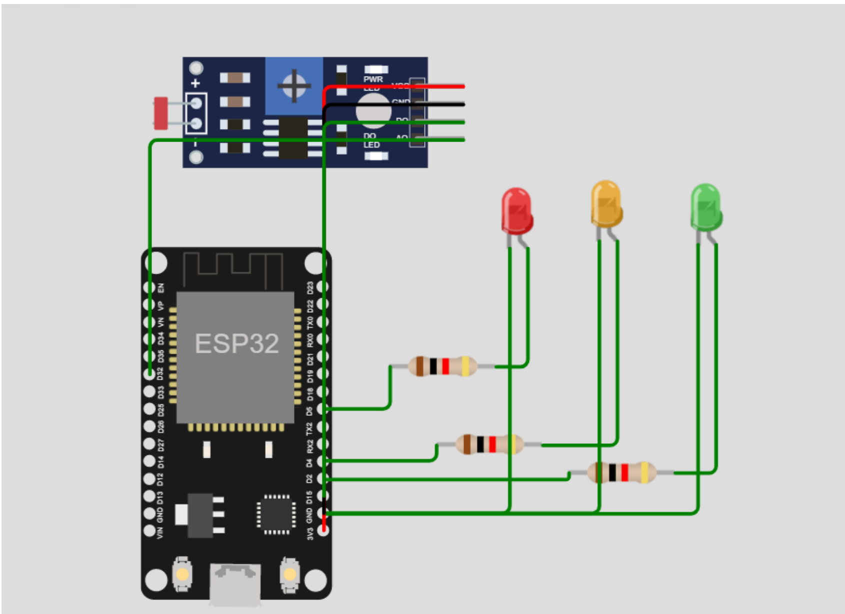
3.1 Functional Requirements

- Real-time weather data integration.
- Weather sensor integration.
- Streetlight control mechanism.
- Automatic brightness adjustment based on weather conditions.
- Remote monitoring and control.

3.2 Non-Functional Requirements

- Energy efficiency.
- Reliability and robustness.
- Scalability.
- Security and privacy.

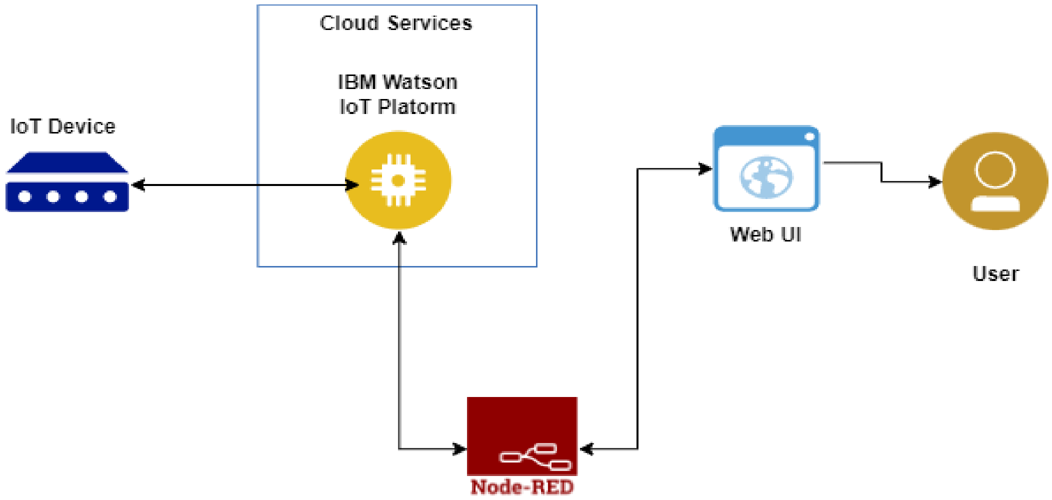
PROJECT DESIGN



4.1 Data Flow Diagrams

Provide relevant data flow diagrams depicting the flow of data and control between different components of the system, such as weather sensors, IoT devices, and streetlight controllers.

4.2 Solution & Technical Architecture



Explain the overall architecture of the system, including the integration of weather sensors, IoT devices, and streetlight controllers. Describe the communication protocols, data storage mechanisms, and other relevant technical aspects.

4.3 User Stories

Outline user stories that describe the interactions and requirements from the perspective of different stakeholders, such as pedestrians, drivers, and municipality authorities.

5.CODING & SOLUTIONING (Explain the features added in the project along with code)

5.1 Feature 1

Describe one of the key features implemented in the project. Explain the purpose, functionality, and importance of the feature. If possible, provide code snippets or pseudocode to illustrate the implementation.

5.2 Feature 2

Describe another key feature implemented in the project. Provide an explanation of its purpose, functionality, and significance. Include code snippets or pseudocode to demonstrate the implementation.

5.3 Database Schema (if Applicable)

If the project involves a database, provide the schema or structure of the database tables along with a brief description of their purpose.

RESULTS

6.1 Performance Metrics

Evaluate the performance of the IoT Based Weather Adaptive Street Lighting System. Measure and present performance metrics such as energy savings, lighting efficiency, and user satisfaction.

Discuss the advantages and disadvantages of implementing the IoT Street Lighting System

ADVANTAGES & DISADVANTAGES Street Lighting System:

Advantages:

- ✓ Energy efficiency: The IoT-based weather adaptive street lighting system can significantly reduce energy consumption by adjusting the brightness of streetlights based on real-time weather conditions. This helps in saving electricity and reducing operational costs.

- ✓ Cost savings: By optimizing the usage of streetlights, the system can lead to cost savings in terms of electricity bills and maintenance expenses.
- ✓ Enhanced safety: The system ensures appropriate lighting levels based on weather conditions, improving visibility and safety for pedestrians and motorists.
- ✓ Environmental benefits: The reduced energy consumption translates to a smaller carbon footprint, contributing to environmental sustainability.
- ✓ Remote monitoring and control: With IoT technology, the system allows remote monitoring and control of streetlights, enabling efficient maintenance and quick troubleshooting.
- ✓ Real-time data analysis: The system collects and analyzes data on weather conditions and lighting patterns, which can be valuable for city planning and infrastructure improvements.
- ✓ Scalability and adaptability: The IoT-based system can be easily scaled up to cover larger areas and can adapt to changing weather conditions.

Disadvantages:

- ➢ Initial setup cost: Implementing an IoT-based weather adaptive street lighting system requires investment in hardware, sensors, connectivity infrastructure, and software development, which can be costly.
- ➢ Technical challenges: Building and maintaining the IoT infrastructure, ensuring reliable connectivity, and addressing compatibility issues among various components can pose technical challenges.
- ➢ Data privacy and security: The system collects and processes data, raising concerns about data privacy and the need for robust security measures to protect sensitive information.
- ➢ Dependency on technology: The system's performance is reliant on the proper functioning of sensors, connectivity, and software components. Any technical issues or failures can impact its effectiveness.

- Potential complexities: Integrating the system with existing street lighting infrastructure and coordinating with multiple stakeholders may introduce complexities and require careful planning and coordination.

CONCLUSION:

In conclusion, the IoT-based weather adaptive street lighting system offers numerous advantages in terms of energy efficiency, cost savings, enhanced safety, and environmental benefits. By leveraging real-time weather data and IoT technology, the system can dynamically adjust the brightness of streetlights to optimize energy consumption while ensuring appropriate lighting levels based on weather conditions. However, the implementation of such a system comes with challenges, including initial setup costs, technical complexities, data privacy concerns, and dependency on technology. Despite these challenges, the system's benefits make it a promising solution for smart cities aiming to improve their lighting infrastructure.

FUTURE SCOPE:

The future scope of the IoT-based weather adaptive street lighting system

Includes:

Integration with smart city initiatives: The system can be integrated with other smart city initiatives such as traffic management, parking systems, and public safety to create a more comprehensive and interconnected urban environment.

Predictive analytics: By analyzing historical weather data and lighting patterns, the system can develop predictive models to anticipate lighting requirements and optimize energy usage proactively.

Sensor advancements: Advancements in sensor technology can enhance the accuracy and reliability of weather data collection, enabling more precise adjustments in streetlight brightness.

Artificial Intelligence (AI) integration: AI algorithms can be incorporated into the system to improve its decision-making capabilities, enabling more sophisticated and efficient lighting control.

Smart maintenance and fault detection: The system can be enhanced to detect and report faults or failures in streetlights automatically, enabling proactive maintenance and minimizing downtime.

APPENDIX

Source Code:

```
#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 "6yocvj"//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
successfully 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="";





}
```