Deakin University

A Smart Home Lighting System

Project Report

Student Name: Aamya Gupta

Date of Submission: 10th October 2024

High-Level Problem / Problem Description

In usual home settings, the electricity consumption is comparatively high because of manual switches to turn light on and off. Eventually over the years, each house consumes almost 20% avoidable electricity. If we can improve this problem, we can enhance sustainability by reducing excess electricity usage across the residential areas.

The main outcome of this project is to automate the lightening systems at houses to avoid unnecessary power consumption. It is also idle to be used by elderly who find it difficult to walk around the house to switch on and off the lights. The min pointers are written below:

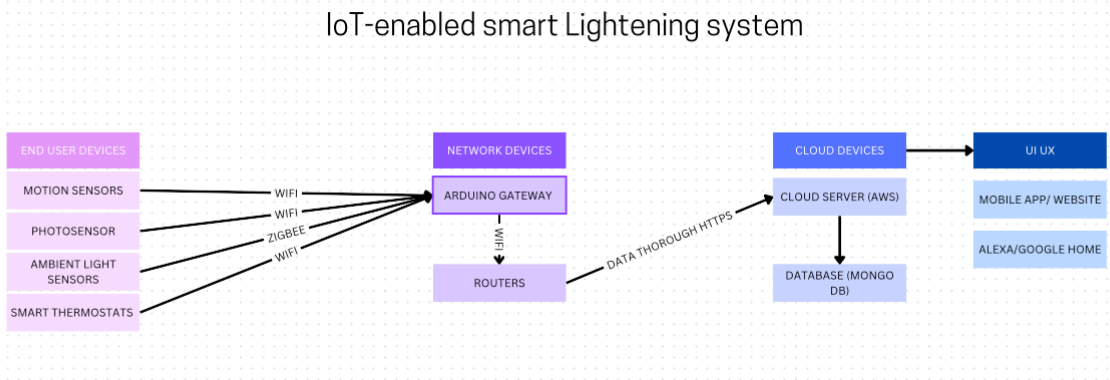
- **Energy Efficiency**: It will significantly reduce energy consumption by automating lighting adjustments based on real-time environmental data.
- **Enhanced Comfort**: It will improve the comfort of occupants by providing appropriate lighting levels tailored to specific conditions and preferences.
- **Integration and Scalability**: It will demonstrate a scalable solution that can be integrated with other smart systems in homes and commercial buildings, paving the way for broader smart building applications.

Usually in the market, we get different motion sensor and Bluetooth operated bulbs, but this system has a different dimension to it. This system not only has motion sensor but also photosensor that changes the voltage of the bulb according to the natural light already present according to the scene of the room.

Solution overview

In order to make residential electricity usage more efficient, this proposal suggests an elaborate automated system of lighting that will encompass both motion and photosensors. This intelligent lighting system will lower unnecessary electrical energy consumption by as much as 20%, therefore contributing to environmental conservation and providing extra comfort for older persons or people living with disabilities.

High Level Block Diagram



High-Level Block Diagram includes majorly these Components:

- **Sensors**: Motion sensors turn on lights when people are present. Photosensors adjust light brightness based on daylight, which saves energy when it's not needed.
- **Actuators**: Smart bulbs or LEDs that can change their brightness and colour temperature. **Controllers**: A main computer that gets information from sensors and tells the lights what to do.
- **Communication System**: Uses Wi-Fi or Zigbee to help sensors, controllers, and actuators talk to each other.
- **Storage**: Places to keep sensor data and what users like, which helps the system learn and get better like MongoDB.

Key Scenarios/Cases:

- **No light outside + No motion:** Dim the LED to reduce energy consumption.
- **No light outside + Motion detected:** Turn on the LED at full brightness to provide sufficient lighting.
- **Light outside (regardless of motion):** Keep the LED off to avoid unnecessary energy usage.

Implementation

Implementation Done:

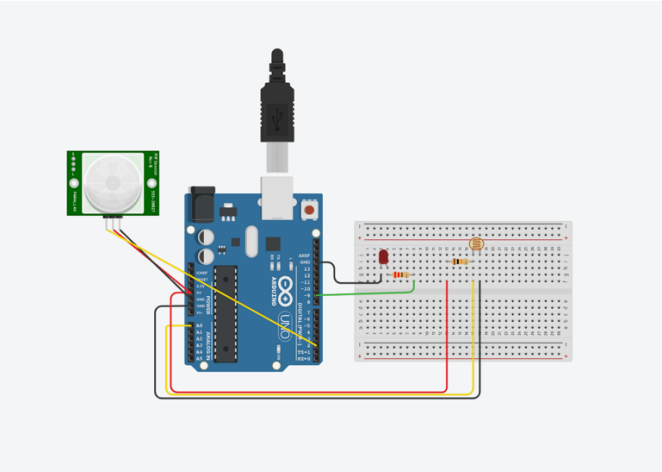
- **Python for data generation:** Python scripts generate sensor data for both motion and ambient light levels.

- **Node-RED for data flow and cloud integration**: The data is processed and sent to the cloud via Node-RED, where the system's logic is implemented.

- **AWS for cloud infrastructure**: AWS services are utilized to handle data storage, remote monitoring, and scaling capabilities.

- **JavaScript for frontend**: A simple web interface is built using JavaScript to visualize the data and provide user control over the system.

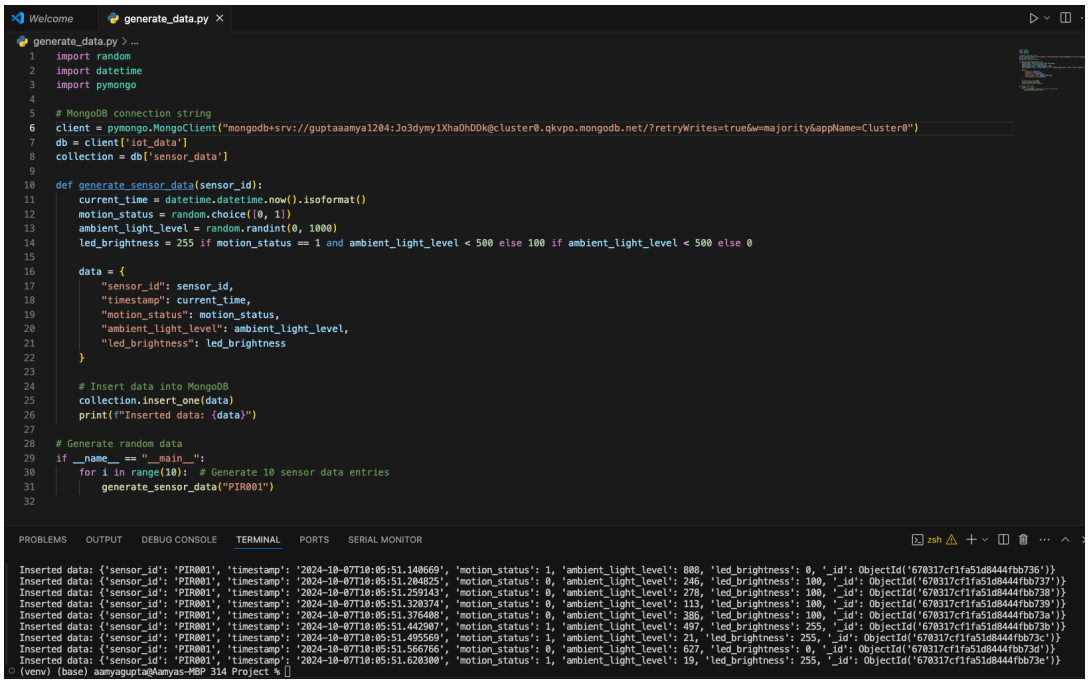
Tinker Cad



```
Motion Detected: 1
Motion detected and low light: LED Full brightness
LDR Value: 78
Motion Detected: 1
Motion detected and low light: LED Full brightness
LDR Value: 78
Motion Detected: 1
Motion detected and low light: LED Full brightness
LDR Value: 78
Motion Detected: 1
Motion detected and low light: LED Full brightness
LDR Value: 78
Motion Detected: 1
Motion detected and low light: LED Full brightness
LDR Value: 78
Motion Detected: 1
Motion detected and low light: LED Full brightness
LDR Value: 78
Motion Detected: 1
No motion, but low light: LED Dimmed
LDR Value: 78
Motion Detected: 0
```

Part1 – Data Generation

A Python script simulates real-time sensor data for both motion detection and ambient light levels. The script generates random data based on pre-defined parameters for motion status and light levels.

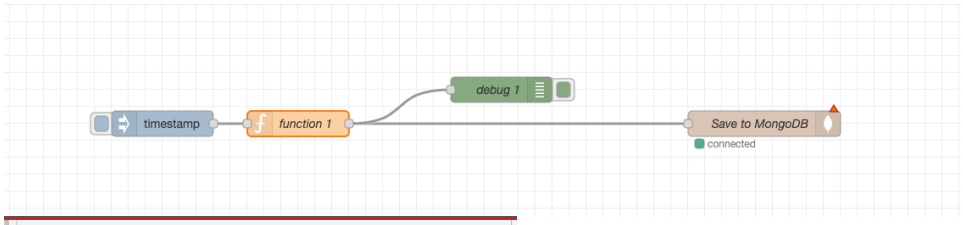


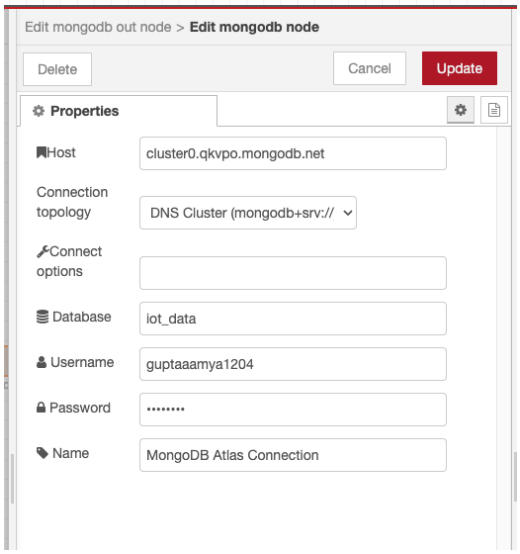
Part 2 – Data Processing Using Node-RED

Node-RED processes the sensor data from the Python script, applying logic to control the lighting:

- Full brightness if motion is detected and light levels are low.
- Dim the LED if no motion but light levels are low.

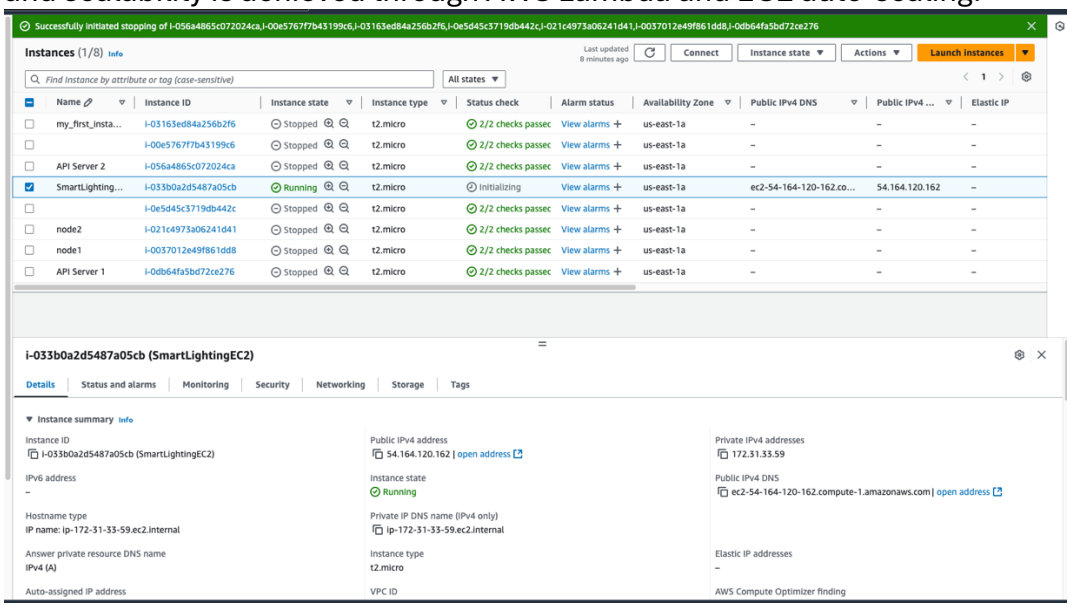
- Turn off the LED if there's sufficient ambient light.
  Processed data is sent to AWS IoT and MongoDB for storage.

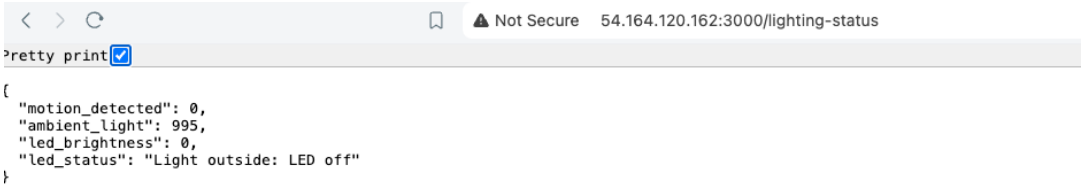




Part 3 – Cloud Integration with AWS

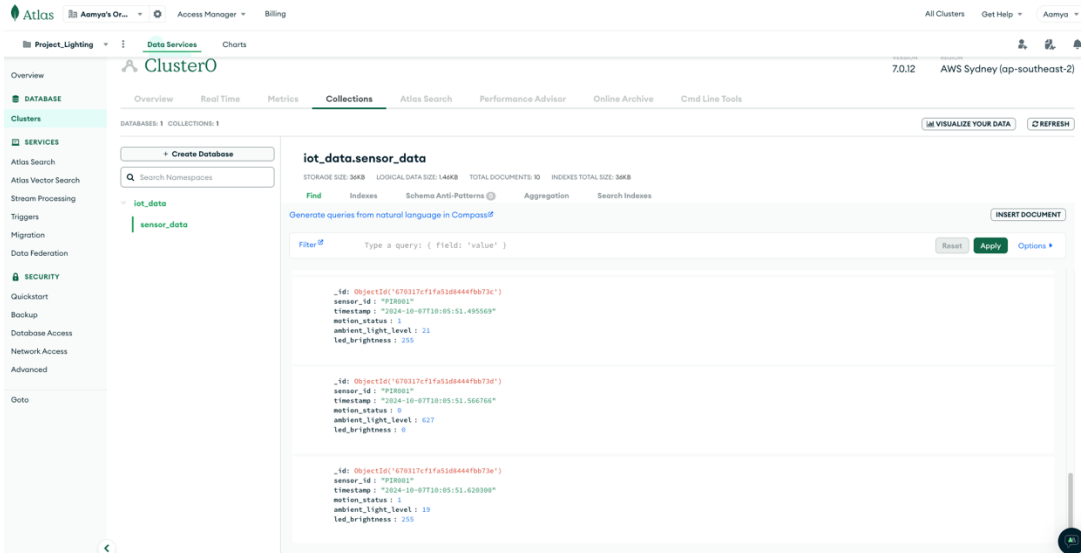
The system uses AWS IoT for cloud integration and scalability. Data is processed on an AWS EC2 instance and stored in MongoDB Atlas. AWS provides real-time monitoring, and scalability is achieved through AWS Lambda and EC2 auto-scaling.





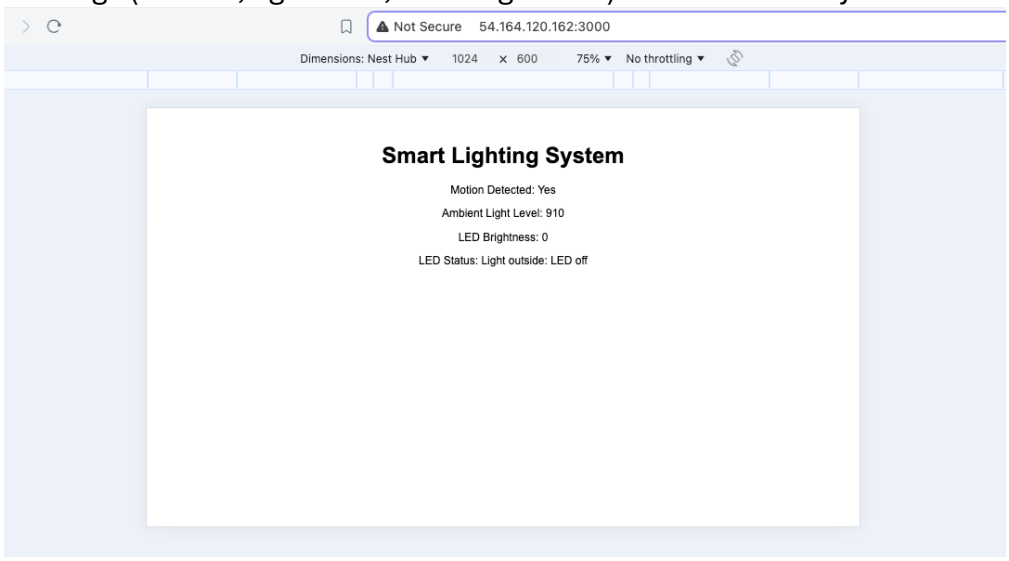
Part 4 – MongoDB for Data Storage

MongoDB Atlas stores all sensor data, including motion_status, ambient_light_level, and led_brightness. Data is retrieved from Node-RED and used for long-term analysis, ensuring scalability as more sensors are added.



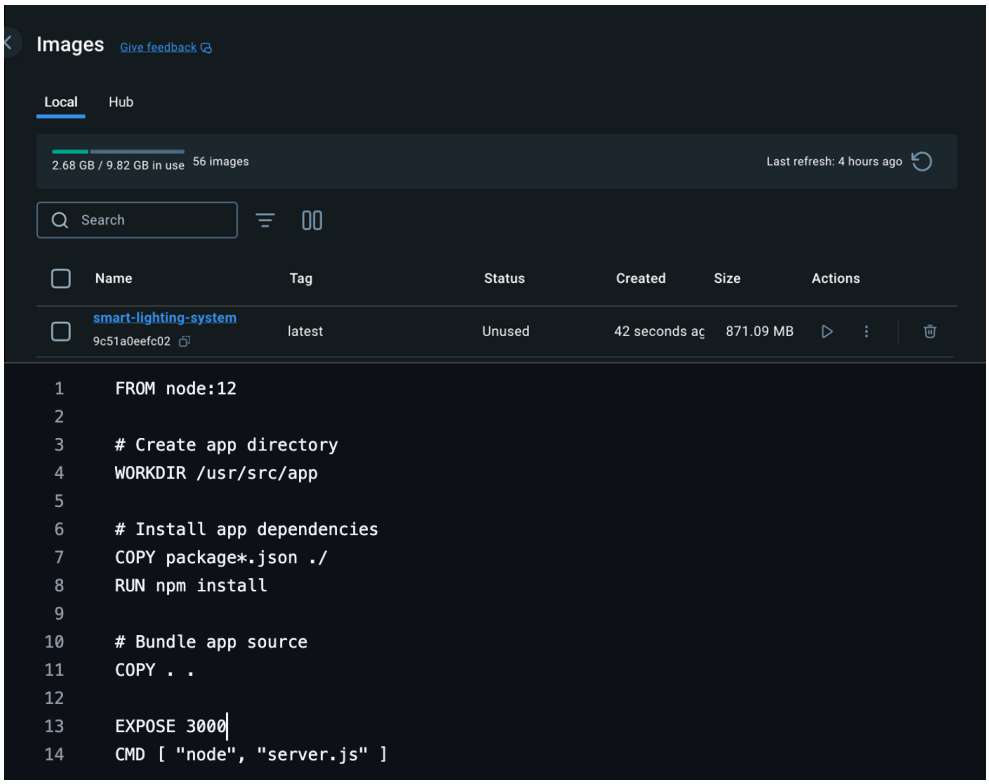
Part 5 – Frontend Interface Using JavaScript

A JavaScript-based frontend displays real-time sensor data and allows user control over the system. It fetches data from AWS and MongoDB, displaying current sensor readings (motion, light level, LED brightness) in a user-friendly interface.



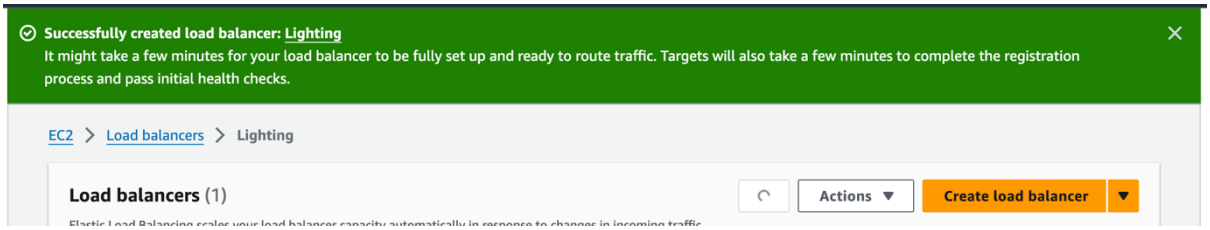
Part 6 – Containerization:

This is how I have used containerization to run my instances and code in one go and easily.



Part 7 – Load Balancer:

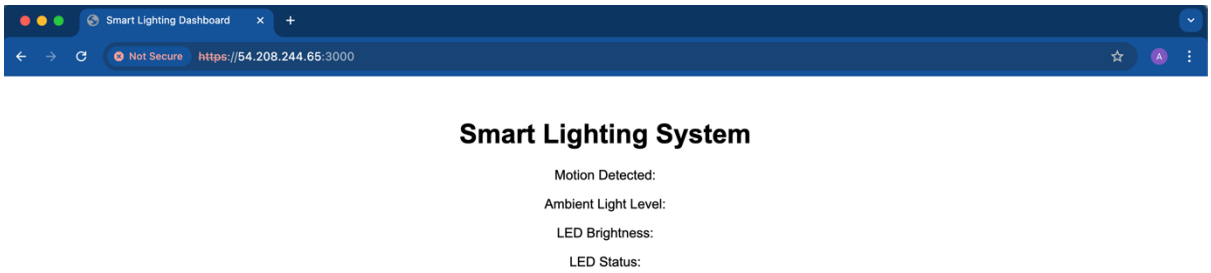
Incorporating load balancing is important for distributing incoming requests evenly across multiple instances to avoid overloading a single instance. Therefore I have created a load balancer named Lighting for the project.



Part 8 – Security

In the server code, I am now using **Helmet** for security reasons. This way, I can secure HTTP headers, which helps protect the application. Helmet enforces secure practices such as **Strict-Transport-Security (HSTS)**, ensuring all communications are made over HTTPS, further enhancing the overall security of the system. As you can see in the image below, we have https here. It is showing not secure because we created the

certificates ourselves so it is not considered authentic.



Part 9 – Microservices:

Through this project, I am utilizing different **microservices** together to ensure scalability and flexibility. Each service, such as sensor data collection, lighting control, and data storage, operates independently, allowing for seamless communication and coordination between them. This approach not only enhances system modularity but also enables easier maintenance and the ability to scale individual services as needed, making the overall architecture more resilient and efficient.

Scalability and Safety Considerations

Scalability:

- **Cloud-Based Architecture** using AWS IoT ensures the system can scale from small homes to large buildings.
- **MongoDB Atlas** allows for easy data storage expansion, handling more sensors without performance loss.
- **AWS Lambda and EC2** auto-scaling manage increased workloads seamlessly, ensuring system responsiveness.
- **Future Scalability** enables integration with more smart devices, expanding the system's capabilities.

Safety/Protection:

- **Data Encryption** ensures secure transmission of sensor data to the cloud.
- **Authentication Mechanism** restricts access to the system's web interface, protecting user controls.
- **AWS IoT Security**: TLS encryption secures device-to-cloud communication.
- **MongoDB Atlas** offers secure, cloud-based data storage, ensuring data integrity and protection.

Resources:

This GitHub Repository has all the code that I have utilised for the project.
https://github.com/aamyaaaa/A-Smart-Home-Lighting-System./tree/main

Conclusion

The Smart Home Lighting System developed in this project showcases an energy-efficient, scalable, and secure solution for residential and commercial use. By automating lighting based on motion detection and ambient light levels, the system reduces unnecessary power consumption and enhances user comfort. The use of cloud technologies like AWS IoT and MongoDB Atlas enables real-time monitoring, easy scalability, and secure data handling. The system is designed to grow from small-scale residential applications to large commercial environments, making it a versatile solution for the future of smart buildings. With its integrated sensors, cloud-based architecture, and a user-friendly frontend interface, this project offers a promising approach to creating sustainable and smart environments.