SENIOR PROJECT SD2-2022

Smart Lighting in the Campus: The Web Application and Data Analytics

Design Specifications

Submitted to

School of Information, Computer and Communication Technology
Sirindhorn International Institute of Technology
Thammasat University

November 2022

by

Paphana Yiwsiw 6222780379
Thanakit Lerttomolsakul 6222780569
Sirada Chaisawat 6222780668
Leeyakorn Cheewakriengkrai 6222770933

Advisor: Dr. Somrudee Deepaisarn

Abstract

Nowadays, the urbanization trend in major metropolitan areas is increasing based on various implementations of intelligent systems and technology. This project developed an automatic system for creating a sufficient lighting environment at Thammasat University, Rangsit Campus, Thailand. The system aims to optimize energy consumption while providing sufficient illumination and extending the current smart street light infrastructure of the campus towards an intelligent and sustainable campus.

The development of this project consists of two sections: the web application and the data analytics. The web application developed an easy-to-use interface for monitoring and controlling installed devices and API solutions for simple real-time environmental and light data collection. The data analytics feature is implemented using feature extraction techniques to create a prediction model for optimizing dimming value. This proof of concept system show that an automatic lighting system can be implemented using existing infrastructure on the campus.

Acknowledgements

We would like to express our deepest gratitude to Dr. Somrudee Deepaisarn, the project advisor, for her guidance, support, patience, and encouragement. Without her invaluable advice, knowledge, and expertise, this project could not have been finished. It has been a great pleasure for us to work with her.

We would like to thank Suphachok Buaruk and Chanon Tantiwattanapaibul for their support and suggestions for this project.

We would like to thank Prof. Dr. Virach Sornlertlamvanich and his team for providing data and information and letting us be involved in the project about the smart street light system at Thammasat University, Rangsit Campus, Thailand.

We would also like to thank Dr. Boontawee Suntisrivaraporn and the SparkBeyond team for providing us with tutorials, guidance, and recommendation for the implementation of the SparkBeyond platform in this project.

Finally, we would like to acknowledge our friends and family for their tremendous support throughout the project, both physically and emotionally.

The financial support provided by the Thammasat University Research fund under the Thailand Science Research and Innovation Fundamental Fund, Contract No. TUFF19/2564 and TUFF24/2565, for the project of "AI Ready City Networking in RUN", based on the RUN Digital Cluster collaboration scheme.

Contents

List of Figures

Abbreviations

The technical terms listed below will be expressed by their abbreviations in this report. Other abbreviations will be defined at their first occurrence.

API	Application Programming Interface
CMS	Central Management System (Smart Street Light External Control Platform)
CSV	Comma-separated Values
HTTP	Hypertext Transfer Protocol
MAE	Mean Absolute Error
QoL	Quality of Life
REST	Representational State Transfer
RMSE	Root Mean Square Error
SQL	Structured Query Language

Chapter 1

Project Concept

1.1 Summary

This project implements and develops an easy-to-use control and maintenance platform for a smart street light system installed inside Thammasat University, Rangsit campus, Thailand. This project is separated into two parts: the web application and the data analytics. In the web application, we provided the dashboard for visualizing data from the environmental sensor and various devices, the interface for monitoring and controlling devices, and API for connecting with CMS API, which is the interface for controlling the smart street light on the campus. For data analytics, we build prediction models using the feature extraction technique and machine learning to optimize light dimming values based on the campus environment.

1.2 Motivation

In this project, we aim at the automatic system to adjust the proper lighting environment for the campus. Smart street lighting devices along six main roads in Thammasat University, Rangsit campus are scheduled to turn on and off at the specified time daily, allowing safety and convenience for the public. However, static scheduling does not guarantee the optimized lighting solution for the campus, and weather conditions always change by the influence of the environment, which is uncontrollable and mostly unpredictable. Lighting devices may need to turn on or off in response to those weather conditions, which is not the current specified period, thus generating unnecessary electrical usage or reducing road safety. Several cities have implemented smart street lighting systems. For example, Amsterdam Smart City (ASC) in Amsterdam, the Netherlands, developed an automatic system used with smart lighting devices to adjust luminance according to the surrounding environment in the area to serve different weather conditions and control the pedestrian flow [1]. Another example can be seen in the smart street light project in Barcelona, Spain, which installed an LED lighting system. They developed an API to communicate with the management system [2] and integrated it with data collection of meteorological data [3].

When considering the availability of devices including 167 smart lighting devices with each device connected by one of the three gateways, and measures meteorological and environmental surroundings by using the environmental sensor, we can collect the data to predict environmental conditions in the future using an AI-assisted machine learning framework (SparkBeyond) and send commands to control the light dimming value of lighting devices for adjusting it accordingly. By developing the web application to monitor and control the system and integrating data analytics from the environmental sensor data, we expand the following potential to users, which are university maintenance personnel, and benefit students and residents. Moreover, it may reduce unnecessary electrical usage and cost, enhance road and public safety, and improve the Quality of Life (QoL) of residents at Thammasat University, Rangsit Campus [4].

1.3 Users and Benefits

Two main groups of users and stakeholders will receive benefits from this project. The first group is students, staff, and residents who live around Thammasat University, Rangsit Campus. They will benefit from this project because there will be sufficient lighting in the area, which will improve their security and QoL.

The other group is the staff who monitor and control the light system of the campus. Most of them are from the building and ground services department. The easy-to-use web application interface will help them monitor and control the light system and respond to maintenance easier and faster. They can also help the executives in energy saving and cost reduction. They will have insight into power usage based on weather conditions and help them plan the power usage plan in the future.

1.4 Typical Usage

The typical scenario to use this web application is for monitoring the system, controlling the lighting devices in the area automatically, and maximizing energy efficiency. To maximize efficiency, the installed environmental sensor first collects the data from the surroundings. The collected data consists of meteorological data such as temperature, humidity, illuminance, Ultra Violet A, Ultra Violet B, wind velocity, wind direction, air pressure, etc. Then, the prediction model is constructed using the SparkBeyond platform by using feature extraction techniques based on the collected data. The model is later used to predict the future illuminance value and find the optimal light dimming value. This process ensures there is sufficient lighting in the area at all times, instead of the default setting that is only turned on and off based on the threshold value. The application then executes the command to lighting devices around the university to adjust the light dimming value accordingly.

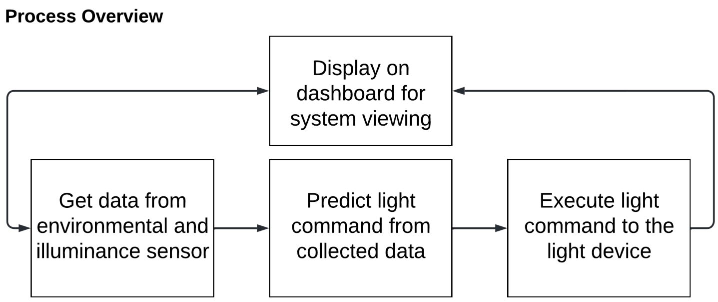


Figure 1.1: The overview of the system process

After all, the web application is provided for maintenance staff for system viewing by showing data on the dashboard in numerical and graphical forms to show the current environmental condition in the area. It can also display the connection status of installed devices and notify staff to repair and make the maintenance correctly Moreover, the web application can be used to control lighting devices in the university area, and report the power usage and lighting control back to the system. The overall process of the web application and the system is illustrated as shown in figure 1.1.

1.5 Main Challenges

The main challenge of web application development is the system instability related to the electrical grid connected to the devices on the campus. Blackouts and electricity drops happen frequently during severe weather conditions and cause devices to malfunction and data loss. Therefore, the web application may not work as intended during those downtimes.

Meanwhile, the challenge of data analytics is the usage of the SparkBeyond platform to create a prediction model. Since this platform is new to all project members, thus we need to learn how to use it for our project. To ensure the prediction model gets the most accurate result and reflects the real-world situation as much as possible, the study of the environmental sensor data must be carried out thoroughly.

Chapter 2

Requirements Specifications

2.1 System Description

The smart street light system at Thammasat University, Rangsit Campus, is installed to provide sufficient luminance to students and residents in the area with control of the system accessible via the internet. The web application is created to help maintenance staff to monitor and control the system easier. As mentioned in the previous section, lighting devices are turned on and off at the specified time which may not provide sufficient luminance and waste electrical energy. The data analytics will help to improve the efficiency of energy consumption and promote sufficient lighting by predicting the future illuminance in the area based on the current environmental condition and calculating the light dimming value to adjust it accordingly. This will later create an automatic light control system that adjusts the light for the future and provide more efficient lighting in the area. Furthermore, the web application provides a dashboard to show the environmental data, energy usage, the lighting control report, and system status for easier maintenance and monitoring.

2.1.1 Perspective

The interaction between the users and the web application is illustrated in figure 2.1

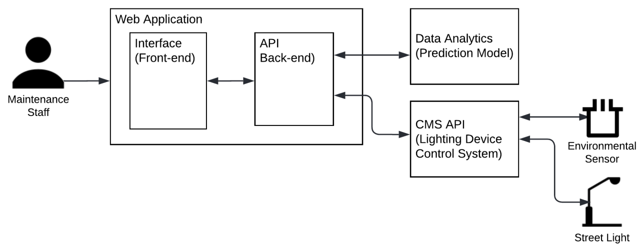


Figure 2.1: Perspective diagram of the system

From figure 2.1, maintenance staff can interact with the system via the web application to monitor and control the smart street light system on the campus. The data analytics platform is connected to the web application in the background to predict the future environmental condition when the new measurement is obtained, as well as communicated to CMS API which is the interface between our system and the control platform.

2.1.2 Functions

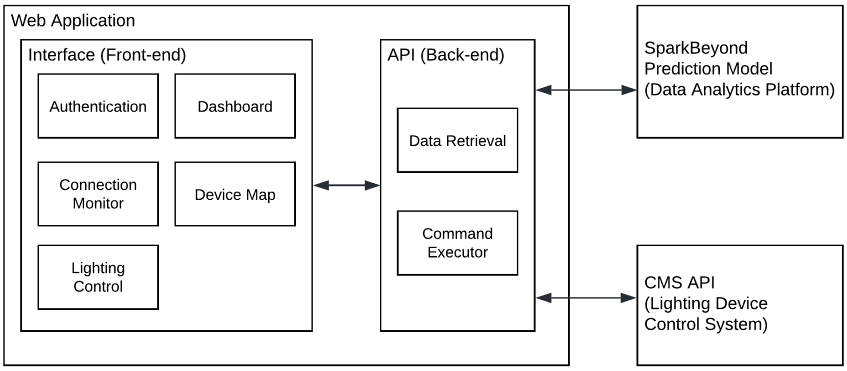


Figure 2.2: The diagram of all functions in the system

The system can provide the web application and data analytics as shown in figure 2.2. Provided functions of the system are listed as follows.

- The frontend web application interface act as an interface between the system and users to provide the following functionalities:
  - Authentication for users to log in to the system
  - System dashboard for displaying graphs of collected environmental data and power usage and control data from installed devices and sensors.
  - Device connection monitor for monitoring connection status of lighting devices.
  - Device map view for showing the location of installed devices.
  - Lighting control panel for adjusting light dimming value of each device.
- Backend API act as a mediator between front-end application, CMS API control system, and SparkBeyond platform.
- Data analytics using the SparkBeyond platform to predict the future illuminance from current environment data using the trained prediction model.

2.2 Requirements

The requirements of the system are listed in the following subsections.

2.2.1 Web Application

WA1 The web application should be clear, easy to use, and easy to navigate.

WA2 The web application must be responsive to any screen sizes.

WA3 The web application must be deployed on the cloud platform.

2.2.2 Data Visualization

DV1 The dashboard must be able to display data from the environmental sensor and illuminance sensors as numerical values and graphs.

DV2 The dashboard must be able to display data from the lighting control and power usage report as numerical values and graphs.

DV3 The dashboard should be able to display data based on the installed area filter.

2.2.3 Device Control

DC1 The lighting device should be controllable using the web application.

DC2 The status of installed devices should be displayed on the web application.

DC3 The map of installed devices should be shown on the web application.

DC4 The disconnection log of devices should be displayed on the web application.

2.2.4 API

AP1 The API should follows the REST architecture and available to use over HTTP.

AP2 The API must be able to send the collected data from installed sensors.

AP3 The API must be able to send the power usage and lighting control report.

AP4 The API must be able to send the current connection status of installed devices.

2.2.5 Data Analytics

DA1 The prediction model should be able to predict future illuminance values using collected data from the environmental sensor.

DA2 The model should be able to calculate the suitable light dimming value for lighting devices based on the environment in the area.

Chapter 3

Design Specification

3.1 System Architecture

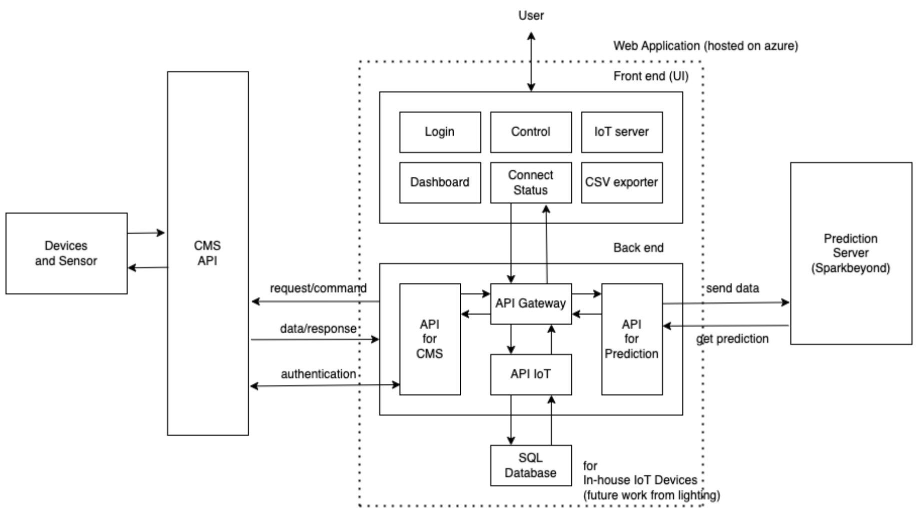


Figure 3.1: The architectural overview of the web application

System Architecture Design is designed based on user requirements. To develop a web application, It is necessary to establish the front-end user interface and the back-end API gateway of the web application. The API gateway at the back end is the medium to communicate between the front-end application, the external control platform, a prediction server, and an SQL database. To display environmental data on the application, the front end sends an HTTP request to the API gateway. Then, the API Gateway for CMS processes the request and sends it to the CMS API that collects data from the devices. After that, the CMS API sends environmental data back to the front end. The environmental data is also sent to the prediction server API. The API passes the data to the prediction server and gets a prediction value from it. Then, the prediction value is sent back to the CMS API and used the prediction value to dim the light of street light devices. In the future, we will have IoT sensors to monitor and collect meteorological and environmental data, so it is necessary to prepare the API for those IoT sensors to help in data collection and visualize data on the front end. The process and architecture of the web application are shown in figure 3.1.

The web application is built using Node.js, a JavaScript runtime environment [5], Express.js framework to handle routing of API and application [6], and Axios module [7] to handle request and response to the external platform, including CMS API, SQL database, and the prediction server. The user interface is built using Bootstrap CSS framework [8] to support responsive web application development for better user experience [9] and standard styling of the interface, jQuery to handle user events on each page, Leaflet.js to display an interactive map of installed device location [10], and Chart.js to display charts with various types of customization [11] on the dashboard and other pages. The web application is hosted on Microsoft Azure using Azure App Services because of their support in Node.js and other runtime environment [12] to make the application accessible on demand [13].

In the data analytics model, the prediction model is created on the SparkBeyond platform using a feature extraction technique. The API of this data analytics server must be created to obtain the prediction value from SparkBeyond and use it on the web application. The prediction model demo_smart_light (SparkBeyond) will be trained every night, and the analytic.py script is responsible for streaming data in the model and streaming predicted value out to a locally-saved file. When the web application sends the request to obtain the prediction value, the query.py script reads a locally-saved file and responds back with the predicted value and suitable light-dimming value within the requested time period. The architecture of the data analytic model and server is illustrated in figure 3.2. The data analytics API is created on Python 3.7 runtime environment, using a flask web framework, and hosted on Microsoft Azure using Azure Virtual Machine.

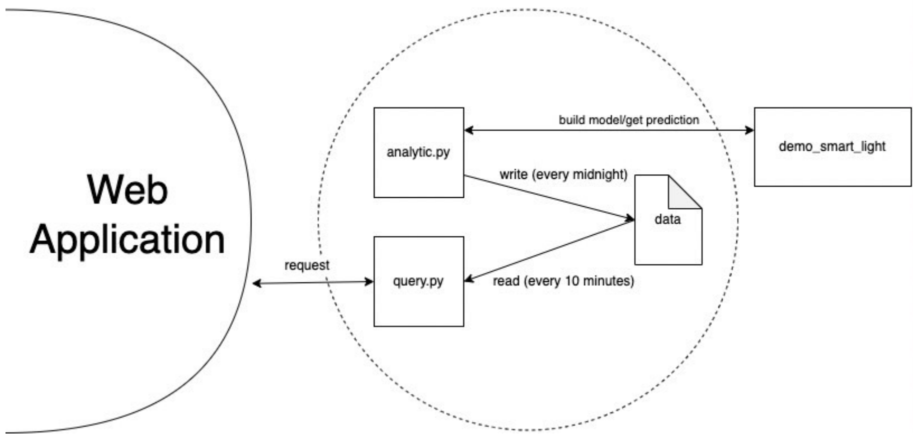


Figure 3.2: The architectural overview of the data analytics and prediction server

3.2 Detailed Design

The detailed design of this project can be divided into 2 sections: the web application and the data analytics.

3.2.1 Web Application Interface Design

The web application serves visualization of data and control of the smart street light devices. In the side navigation bar, pages are grouped into 5 sections based on their functions, which are overview, control, IoT sensors, system status, and about us. The overview section provides a dashboard that shows the collected data from environmental and illuminance sensors to help users to understand and analyze the data and current situation faster. The control section allows users to control the smart street light devices as individual devices or zones. The IoT sensors section provides the display of the collected data from the IoT sensor built in the future. The system status section displays the current connection status of devices in the system, including gateways, sensors, and street light devices, to help users in diagnosing the problem presented in the area such as electrical blackouts and power drops. It also shows the gateway disconnection log and irregular events that occurred to installed devices. The about us page contains the all information about the project.

Login Section

The user login page is shown in Figure 3.3. System administrators, in this case, called users, can log in to the web application or go to the download environmental data page from the link provided on this page.

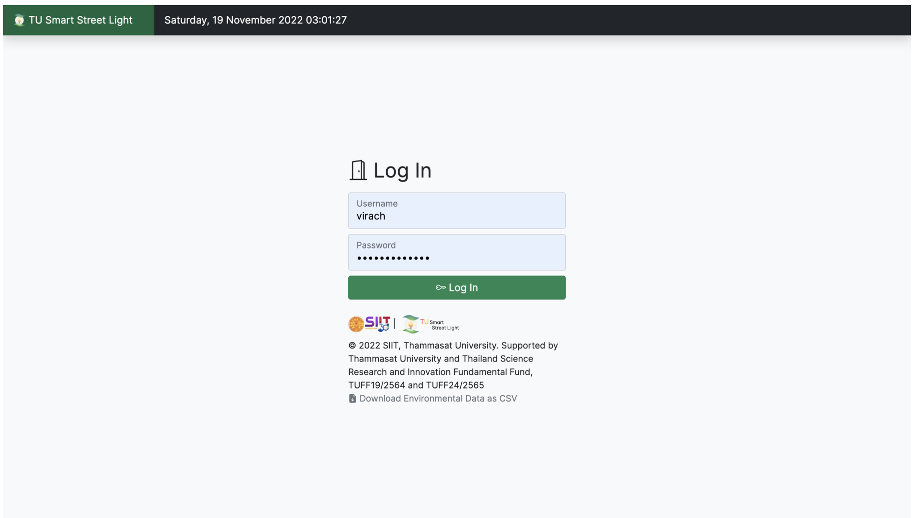


Figure 3.3: The login page of the web application

On the environmental data download page, public users and system administrators can download the data collected by the environmental sensor as shown in Figure 3.4. Those data are prepared in CSV format and grouped by the month they are collected.

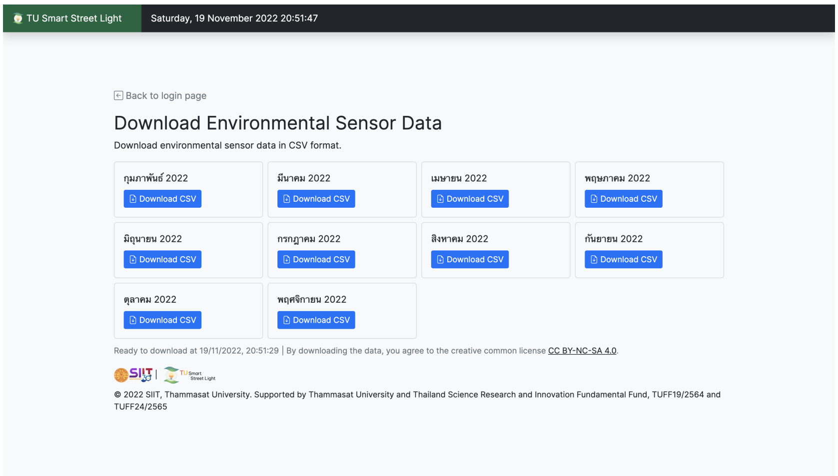


Figure 3.4: The environmental data CSV download page of the web application

Overview Section

Users will be sent to the system overview page after they logged in. The data collected by the environmental sensor are displayed in the dashboard as shown in figure 3.5. Those data consist of temperature, humidity, wind velocity, wind direction, illuminance, rain level, ultraviolet A, and ultraviolet B. They are presented in both numerical and graphical formats. The changes in data from the past 10 minutes are also mentioned under the heading of each data as increasing, decreasing, or no change.

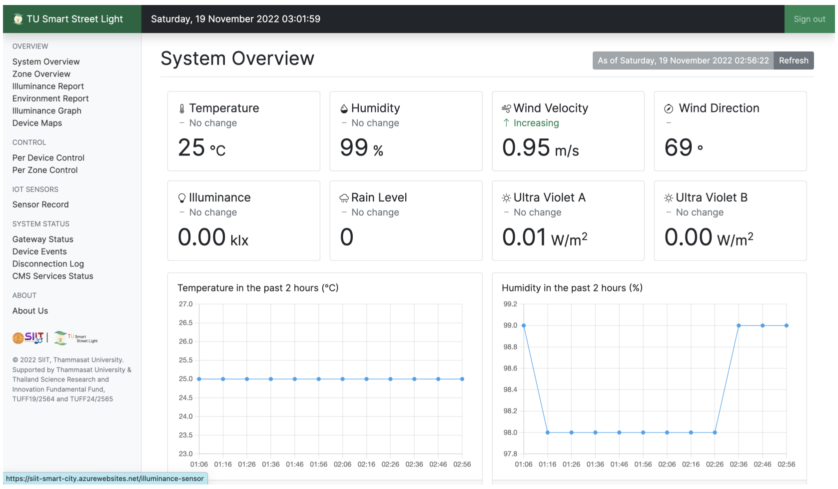


Figure 3.5: The system overview page of the web application

Users can get the data of each specific zone on the zone overview page. They can select the zone from the provided drop-down list. The data will also be displayed in numerical and graphical formats as shown in figure 3.6. Those data consist of average active energy, average active power, average V RMS, and average current power. The list of devices, including power usage data, is shown based on the selected zone also shown in the page.

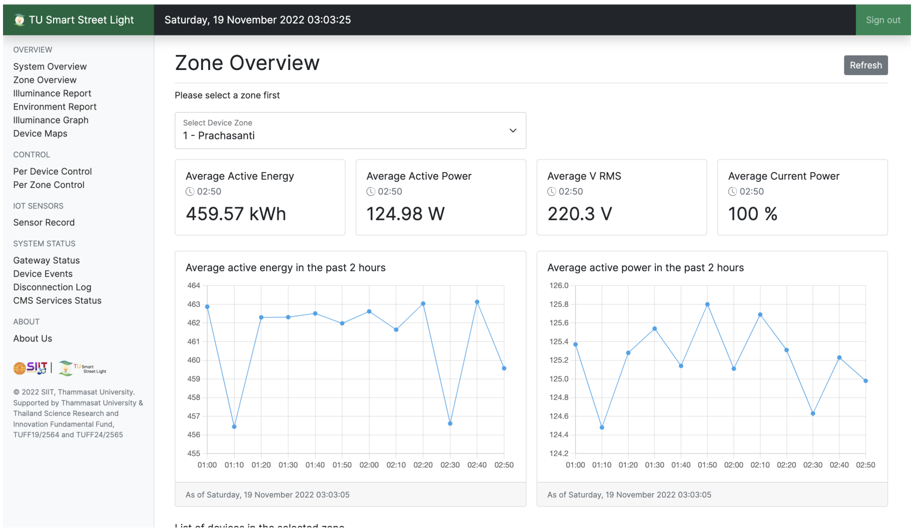


Figure 3.6: Graphs in the dashboard of the zone overview page

The data from the illuminance sensor on each smart street light device are displayed on the illuminance sensor report page as shown in figure 3.7. The dashboard on this page provides the name of the lighting device, collected data in the lux unit, the ID of the street light and sensor device, and the timestamp of the collected data.

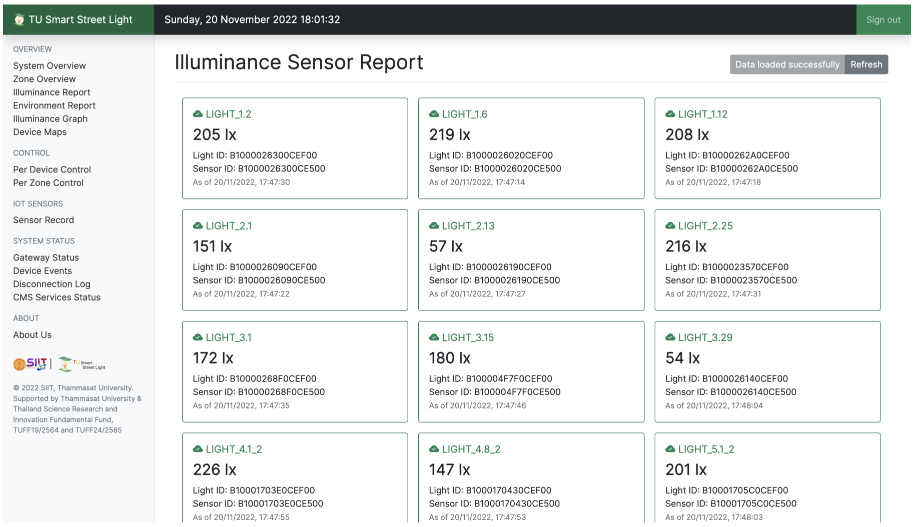


Figure 3.7: The illuminance sensor report page of the web application

Users can observe the change in illuminance value over time on the illuminance graph page as shown in Figure 3.8. The change of collected illuminance value in the past 3 hours of the selected zone is displayed in graphical format.

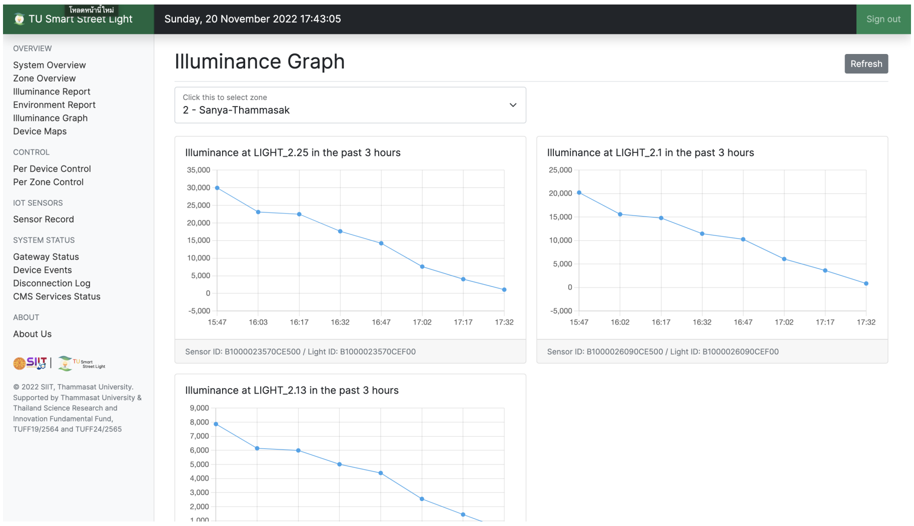


Figure 3.8: The illuminance sensor graph page of the web application

Users can search for a specific range of data collected from the environmental sensor on the Environment report page as shown in figure 3.9. They enter the start time and end time, then the data in that range will be displayed in the table.

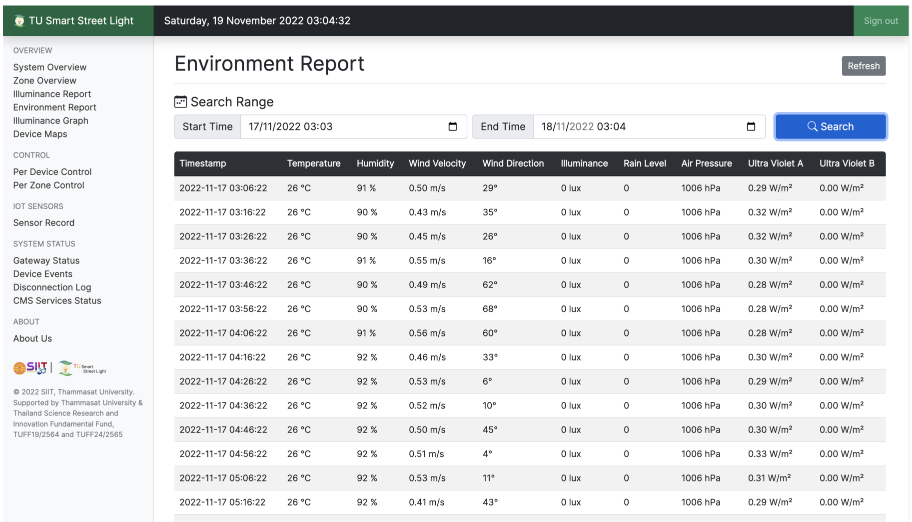


Figure 3.9: The environmental report page of the web application

On the device map page, all the smart street light device locations are displayed on the interactive map. The users can quickly find information about each device by clicking on the yellow pin. The information will pop up as shown in figure 3.10. The pop-up on the marker consists of the device name, device ID, zone name, gateway used, and connection status. All of the information about gateway devices, environmental sensors, and lighting devices are provided under the map as shown in figure 3.11.

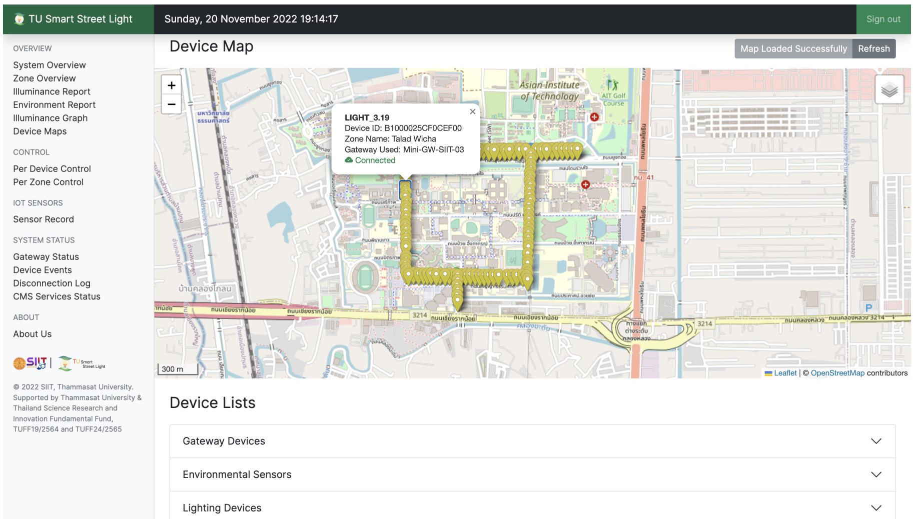


Figure 3.10: The device location interactive map of the web application

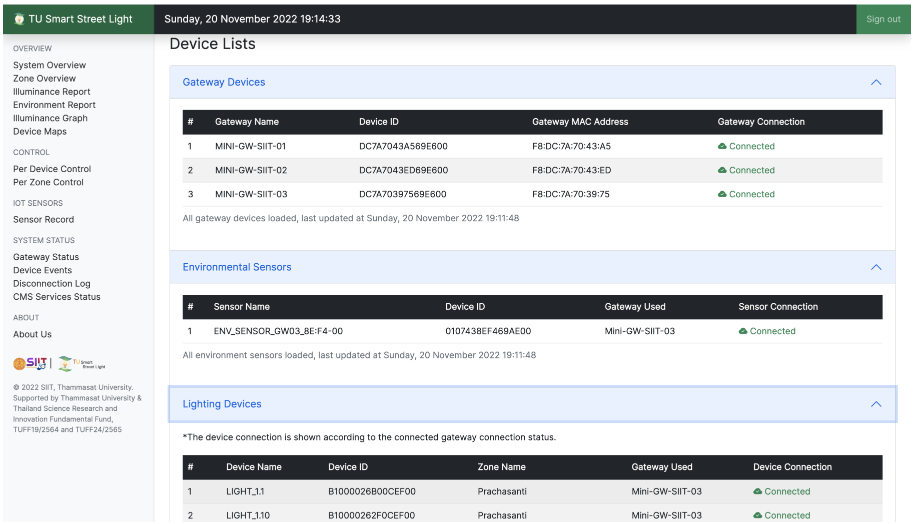


Figure 3.11: The device list in the map page of the web application

Control Section

In the control section of the web application, users can decide to control the smart street light devices as a device or a zone. The per-device control page allows users to adjust the smart street light device individually. The current device status and power usage data will be displayed after the users choose the zone and the device as shown in figure 3.12. The new light power can be set to a specific value between 0 to 100 or set to the predefined

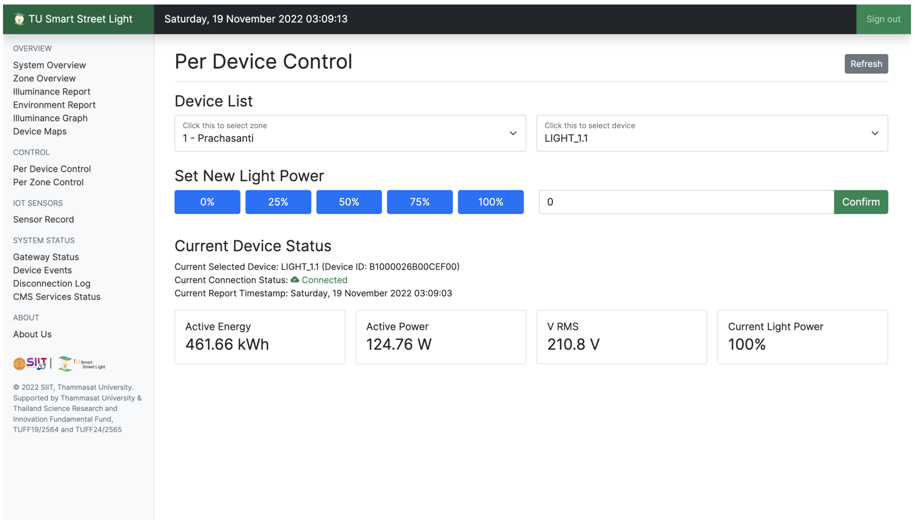


Figure 3.12: The individual device control page of the web application

power percentage. In the per zone control page, users can adjust the light power of a single zone by choosing the preferable zone and the predefined light power percentage or entering a specific value. The power usage data of devices in the selected zone will be displayed in the table as shown in figure 3.13.

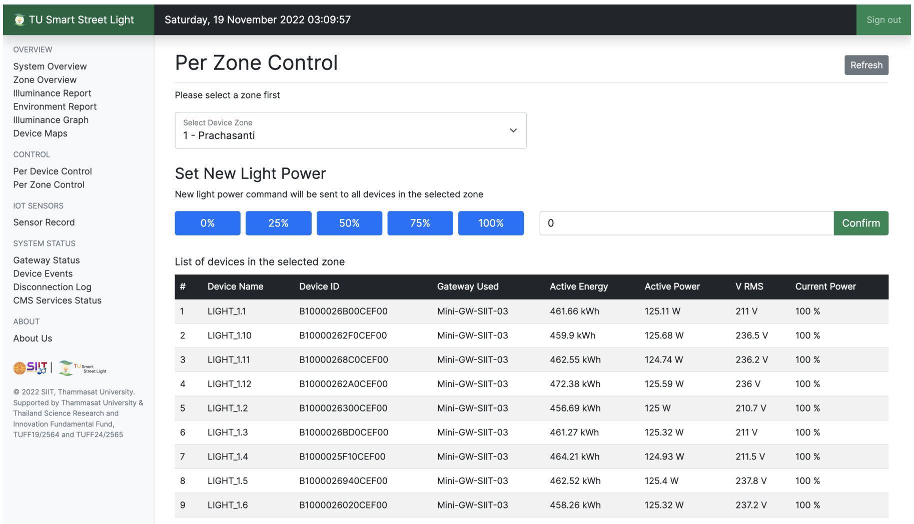


Figure 3.13: The individual zone control page of the web application

IoT Sensors Section

The IoT sensors data page is prepared for displaying the data collected from student-made environmental sensors in the future from the IoT sensor team in the smart street light project as shown in figure 3.14

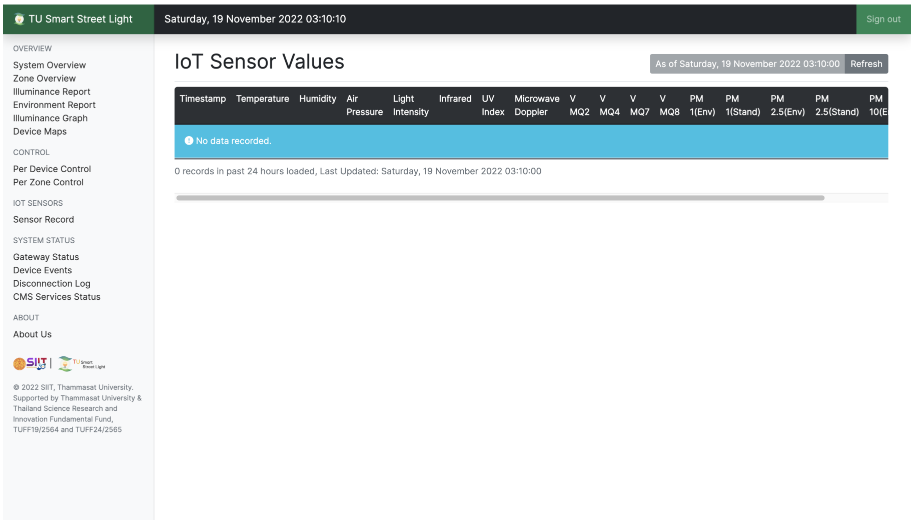


Figure 3.14: The IoT sensor value page of the web application

System Status Section

The system status page displays the status of the gateway and environmental sensor. This page displays the current status of gateways and the disconnected time. The disconnected time can help the system administrator in diagnosing the problem with the system as shown in figure 3.15.

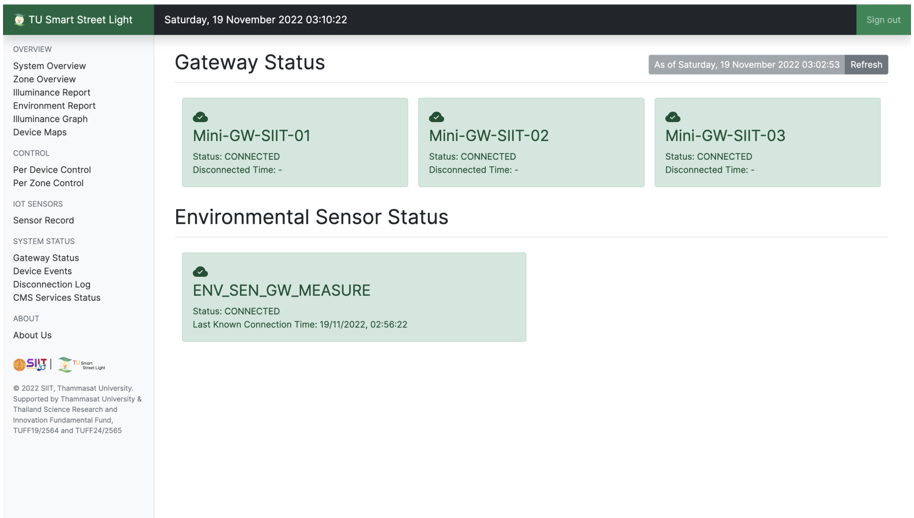


Figure 3.15: The gateway connection status page of the web application

The device event page displays the irregular event occurring to the devices. The system administrator can select to view the event according to the zone or gateway device. The list of events and related information will be shown in the table as illustrated in figure 3.16. The gateway disconnection log page displays the disconnect timestamp of the gateway. Device information of each gateway is also display above each log table as shown in figure 3.17.

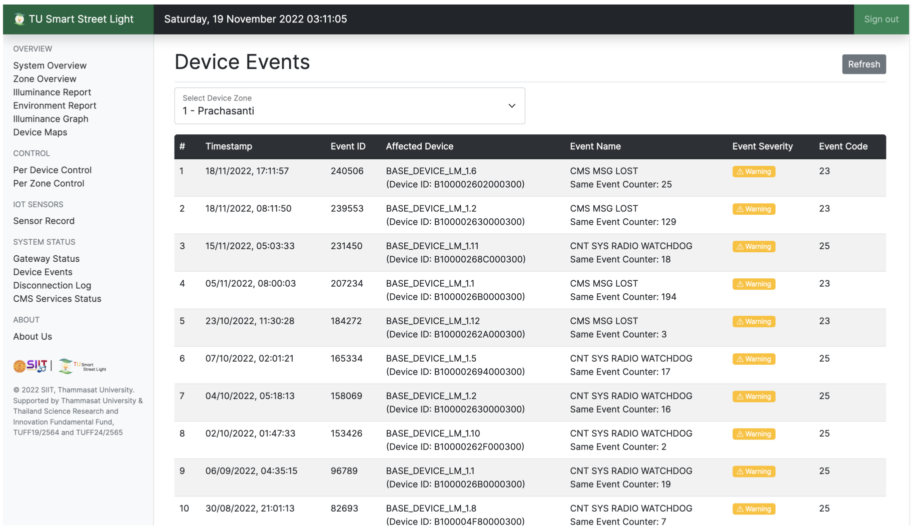


Figure 3.16: The device event page of the web application

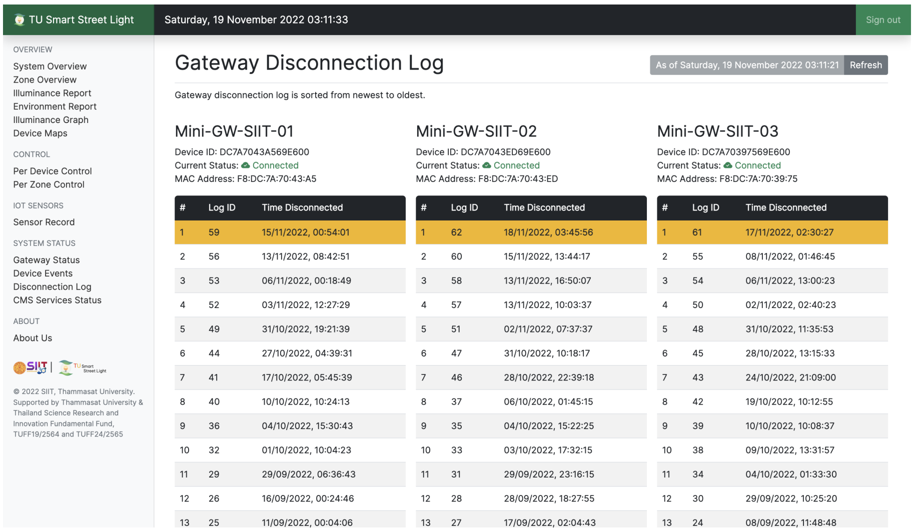


Figure 3.17: The gateway disconnection log page of the web application

The CMS service status page displays the status of the service from the CMS API to further help system administrators in diagnosing the problem in the system as shown in figure 3.18.

About Us Section

The about us page contains information about the project, such as the project name, logo, members, generated paper, copyright statement, and funding information shown in figure 3.19.

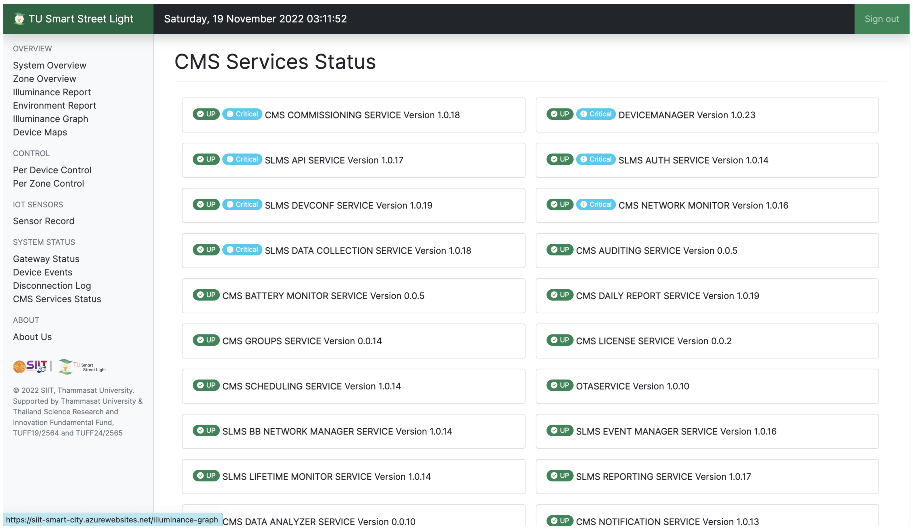


Figure 3.18: The CMS service status page of the web application

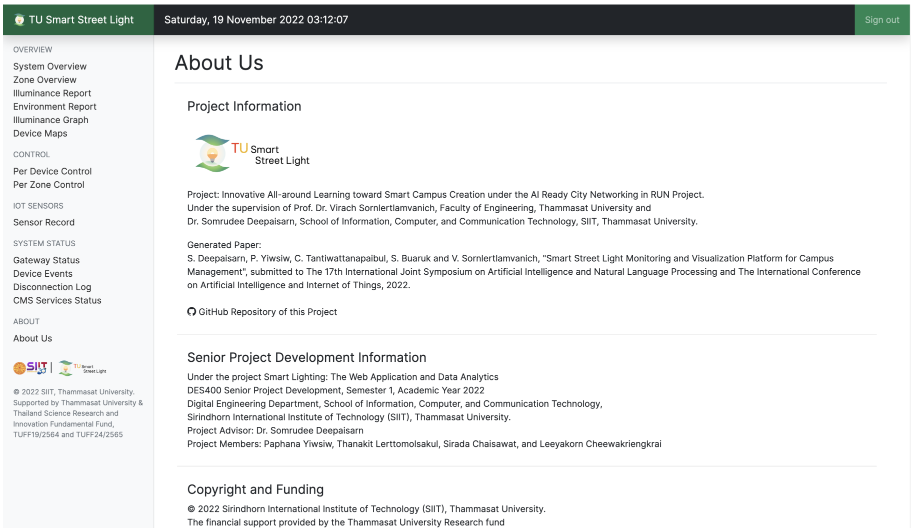


Figure 3.19: The about us page of the web application

3.2.2 Data Analytics Design

In the data analytics part, The Sparkbeyond platform is implemented to build a prediction model for predicting future illuminance value. The collected data from the environmental sensor are used as the imported dataset, and the SparkBeyond python SDK and flask web framework are used to build the prediction server and predict the suitable light-dimming values for smart light devices.

Preprocessing and Import Data

The environmental sensor dataset is imported from the created API in the previous section in JSON format. Those data are preprocessed by using Pandas data frame. Unrelated data, such as the gateway timestamp, and data that may cause the model to overfit, such as Ultraviolet A and Ultraviolet B, are excluded from the dataset. The training set data is from the last 3 months before the model is trained. The training dataset includes date and time, humidity, temperature, air pressure, illuminance, and wind velocity.

Baseline Pipeline Setting

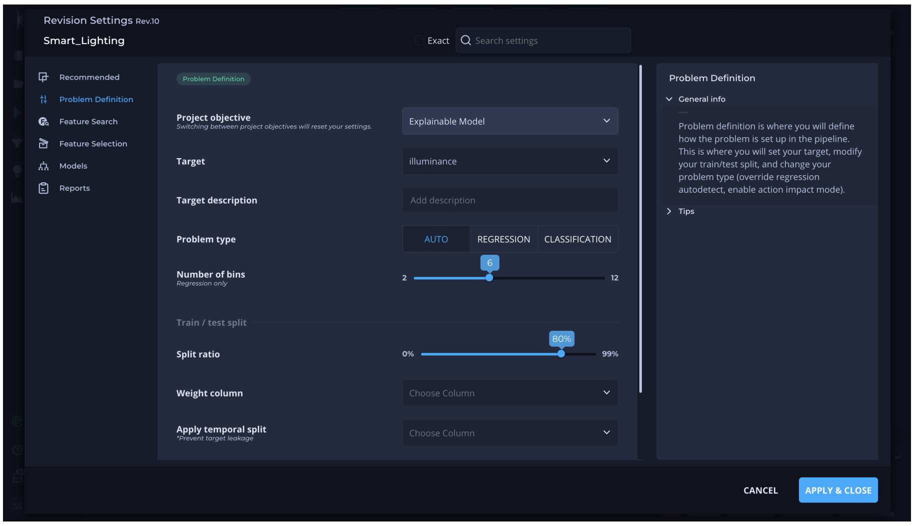


Figure 3.20: The baseline pipeline setting in the SparkBeyond platform

As illustrated in figure 3.20, Sparkbeyond provides the baseline pipeline setting to define the problem statement of this prediction model. The important setting includes the project objective, data split, target, the number of features, and model selection.

- Project Objective: This option is used to define the learning setting depending on the project objective. It can be set to be an insight or explainable model. For this project, the explainable model option is chosen in order to obtain a balance between model performance and feature interpretability.
- Data Split: The dataset is split into the training set and testing set. The split ratio between the training and testing set is set to 80% to 20%.
- Target: Target is the value that we want the model to predict. For this project, the target is set to be illuminance value.
- The number of features: Features extraction is done automatically in the Sparkbeyond platform. The platform will explore all the features, then select only 300 potential features according to the predefined feature count.

- Model Selection: The root mean square error or RMSE of the algorithm is used to compare and find the optimal algorithm for this prediction project. The selected algorithms that are considered in this project are SciKitLearnGradientBoosting, SciKitLearnXGBoost, SciKitLearnRandomForest, and SciKitLearnDecisionTree.

Context Implementation

To predict the future illuminance value, the time series context is connected to the data related to the timestamp of the main dataset as shown in figure 3.21. The time window values are compared by using correlation coefficient and learning time.

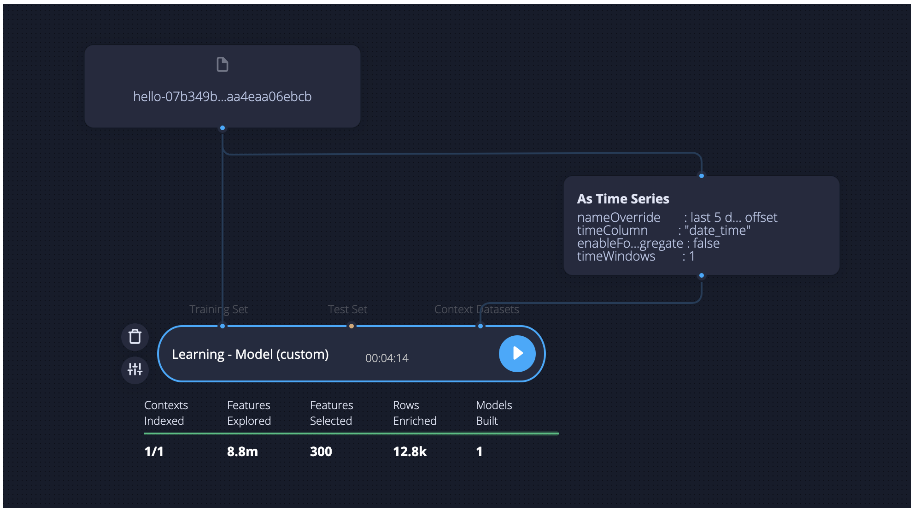


Figure 3.21: The pipeline diagram in the SparkBeyond platform

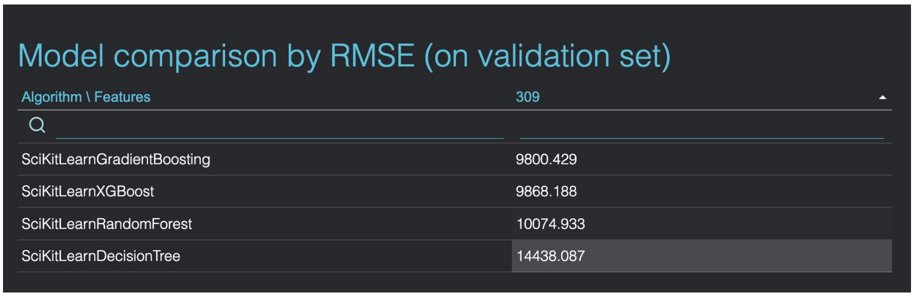
3.3 Test Results

3.3.1 Performance of Prediction Model's Algorithms

In figure 3.22 shows the result of the experiment between 4 algorithms, which are SciKitLearnGradientBoosting, SciKitLearnXGBoost, SciKitLearnRandomForest, and SciKitLearnDecisionTree. By comparing the RMSE of each algorithm, SciKitLearnGradientBoosting performs the best because the RMSE is equal to 9,800.429 which is the least of all 4 algorithms.

3.3.2 Performance of Different Time Windows

Corresponding features are acquired from the dataset based on the specified time windows. As shown in figure 3.23, the performance by using correlation coefficient and learn-



Model comparison by RMSE (on validation set)

Algorithm \ Features	309
SciKitLearnGradientBoosting	9800.429
SciKitLearnXGBoost	9868.188
SciKitLearnRandomForest	10074.933
SciKitLearnDecisionTree	14438.087

Figure 3.22: The comparison of models by RMSE in the SparkBeyond platform

ing time of each different time window is shown. According to the figure 3.23, the correlation coefficient are not significantly different from each other.

Time Windows	Correlation Coefficient	Learning Time
3 days	0.9683	4:53:00
4 days	0.9682	5:53:00
5 days	0.9679	5:50:00
6 days	0.9685	7:04:00
7 days	0.9691	6:38:00

Figure 3.23: The comparison of performance between different time windows

The figure 3.24 also illustrated the relationship between correlation coefficient and learning time which shows the increasing trend of learning time spent when the size of the time window increased. The time windows of 6 to 7 days are selected because of their highest correlation coefficients.

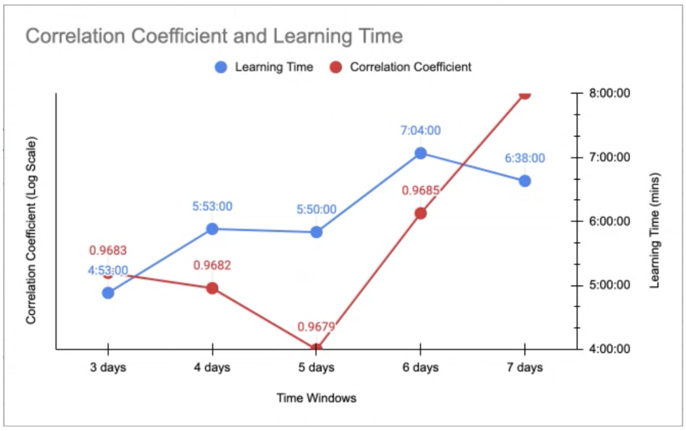


Figure 3.24: The graph between correlation coefficients and learning time across different time window

3.4 System Deployment

The web application can be accessed using the following public URL that generated by Microsoft Azure App Service, https://siit-smart-city.azurewebsites.net. The environmental data download page can be accessible by the public for further research and analytical task about the environmental and meteorological field, using the URL, https://siit-smart-city.azurewebsites.net/csv-download.

For the API created under this web application, it can be accessed using https://siit-smart-city.azurewebsites.net/api/ and follow by defined path in the API documentation, https://documenter.getpostman.com/view/24246318/2s8YekSv8o. The base code of the web application and its API is available on GitHub using the following link: https://github.com/waterthatfrozen/Smart-City/

Bibliography

[1] S. Madakam and R. Ramachandran, "Amsterdam smart city (ASC): fishing village to sustainable city," in *WIT Transactions on Ecology and the Environment*. WIT Press, Jul. 2016. [Online]. Available: https://doi.org/10.2495/sc160681

[2] E. Pejstrup and J. Egenolf, "Integrated infrastructures factsheets," Nov 2019. [Online]. Available: https://grow-smarter.eu/fileadmin/editor-upload/Smart/Action_Area_2-compressed.pdf

[3] G. M. Llode, "A concise review of fifteen intelligent street lighting systems which can pave the way for the emergence of smart cities in the caribbean," *i-manager's Journal on Future Engineering and Technology*, vol. 15, no. 4, p. 28, 2020. [Online]. Available: https://doi.org/10.26634/jfet.15.4.17313

[4] S. Deepaisarn, P. Yiwsiw, C. Tantiwattanapaibul, S. Buaruk, and V. Sornlertlamvanich, "Smart street light monitoring and visualization platform for campus management," in *2022 The 17th International Joint Symposium on Artificial Intelligence and Natural Language Processing and The 3rd International Conference on Artificial Intelligence and Internet of Things (iSAI-NLP-AIoT)*, 2022, pp. 1–5.

[5] N. Chhetri, "A comparative analysis of node.js (server-side javascript)," 2016. [Online]. Available: https://repository.stcloudstate.edu/csit_etds/5

[6] A. Mardan, "Using express.js to create node.js web apps," *Practical Node.js*, pp. 51–87, 2018.

[7] "Axios," https://axios-http.com/, 2022.

[8] "Get started with bootstrap," https://getbootstrap.com/docs/5.2/getting-started/introduction/, 2022.

[9] F. Almeida and J. Monteiro, "The role of responsive design in web development," *Webology*, vol. 14, no. 2, pp. 48–65, 2017. [Online]. Available: http://www.webology.org/2017/v14n2/a157.pdf

[10] V. Agafonkin, “Leaflet - an open-source javascript library for interactive maps,” https://leafletjs.com/, 2022.

[11] H. M. Millqvist and N. Bolin, “A comparision of performance and scalability of chart generation for javascript data visualisation libraries : A comparative experiment on chart.js, apexcharts, billboard, and toastui,” Ph.D. dissertation, University of Skövde, 2022.

[12] V. P. Desai, K. S. Oza, P. P. Shinde, and P. G. Naik, “Microsoft azure: Cloud platform for application service deployment,” *International Journal of Scientific Research in Multidisciplinary Studies*, vol. 7, no. 10, pp. 20–23, 2021. [Online]. Available: https://www.isroset.org/pdf_paper_view.php?paper_id=2581&4-ISROSET-IJSRMS-06711.pdf

[13] N. Dissanayake and K. Dias, “Web-based applications: Extending the general perspective of the service of web,” in *10th International Research Conference of KDU (KDU-IRC 2017) on Changing Dynamics in the Global Environment: Challenges and Opportunities*, 2017.