International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2643-640X
Vol. 6 Issue 6, June - 2022, Pages: 7-14

Design of Website Interface for Smart Greenhouse Monitoring and Controlling System Using NoSQL Database

Sumardi[1], Aghus Sofwan[2], Febrio Rizky Adhipratama[3]

Department of Electrical Engineering, Diponegoro University, Semarang, Indonesia
sumardi@elektro.undip.ac.id[1], aghus.sofwan@gmail.com[2], rizkyadhipratamactricalE@students.undip.ac.id[3]

Abstract: *The rapid development of technology affects many sectors, especially in the field of agricultural systems, one of which is the greenhouse farming system. Greenhouse is a closed planting system that can prevent plants from several factors that can cause damage to growth. Greenhouses interact with the surrounding environment and create a microclimate in the greenhouse. Changes in the microclimate are very influential for greenhouse cultivation plants. The focus in this research is to design a website interface based on MVC architecture (Model, View, and Controller) using a NoSQL database. The website interface is used to monitor and control the smart greenhouse microenvironment based on Internet of Things (IoT). The test results show that the website interface for monitoring and controlling the smart greenhouse environment is able to handle HTTP requests from a maximum of 15 concurrent users.*

Keywords: *Smart Greenhouse, Internet of Things (IoT), NoSQL, MVC Architecture*

1. Introduction

The rapid development of technology affects many sectors, especially in the agricultural sector, one of which is the greenhouse farming system. Greenhouse is a closed planting system that can prevent plants from several factors that can cause damage to growth[1]. Greenhouses are widely used in agricultural systems because they can produce optimal growth because plants are not influenced by external climatic conditions and climatic conditions in the greenhouse can be adjusted as needed.

Greenhouse interacts with the surrounding environment and create a microclimate in the greenhouse. The closed structure of the greenhouse causes the rate of air exchange in the greenhouse to be very small compared to the surrounding environment so that the air in the greenhouse tends to be hotter than the surroundings. Solar radiation that enters the greenhouse will be trapped in the greenhouse so that the temperature in the greenhouse will also be higher than the surrounding environment, resulting in a greenhouse effect[2]. High temperatures can damage the metabolic system in plants so that plants become less productive.

To overcome these problems, a website interface design for monitoring and controlling the micro environment greenhouse was carried out. The main contribution of this paper is to design a website interface on a smart greenhouse system using NoSQL as a database. The design of the smart greenhouse monitoring and control website interface is expected to assist farmers in observing and controlling the greenhouse microclimate remotely.

2. Related Work

Most researches on designing interface for Internet of Things (IoT) was conducted. Research on the design Real-time Water Quality Monitoring and Notification System for Aquaculture by Jomsuda Duangwongsa used Realtime Database as the database[3]. Another research for design IoT interface is the research entitled Integration in The Physical World in IoT Using Android Mobile Application by Thiyagarajan[4]. Research about designing Web Based Environment Monitoring System Using IoT by Pooja Ghule used Thinkspeak cloud storage based on MQTT protocols[5]. Another research designed Smart Greenhouse Management System based on NB-IoT and Smartphone[6]. That research used MySQL as relational database to stored data sensor.

3. Design of NoSQL Database

The database on the website interface of the smart greenhouse monitoring and control system uses NoSQL with a structure in the form of key and value pairs. The database server used is the Firebase realtime database server. The database is used to store sensor reading data from each IoT node and actuator control flags. The grouping of data from each IoT node is grouped based on the parent name of the IoT node name, as shown in Fig. 1. The parent name in database represents the node name.

International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2643-640X
Vol. 6 Issue 6, June - 2022, Pages: 7-14

Fig. 1. *Each Parent's Name of The IoT Node*

Each parent's node has four children, they are Actuator, ActuatorStatus, DataSensor, and Properties. Child Actuator stored actuator's flag data. This flag value will be used as a reference by the microcontroller to control the actuator manually. Child ActuatorStatus is used to store the value of the condition of the actuator whether the actuator is ON or OFF. The data representation of the child actuator status is shown in Fig. 2. Child Properties is used to store node name.

Fig. 2. *The Data Representation of The Child ActuatorStatus*

Child sensor data is used to store sensor readings. Data is grouped by sensor reading date. The grouping of sensor reading data is shown in Fig. 3.

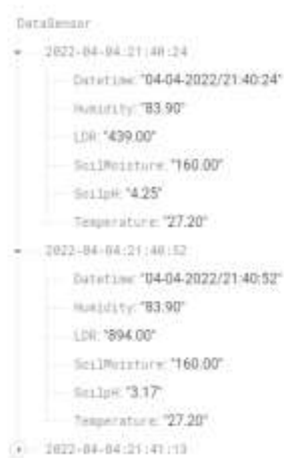

Fig. 3. *Grouping of Sensor Reading Data by Datetime*

4. The Unified Modelling Language (UML) of Website Design

In general, website software for monitoring and control systems has the following functions:

1. Authentication and Authorization

   Authentication functions to verify the identity of users who are trying to enter the system. The authorization function plays a role in checking the access rights of a user in accessing several features in the system.

2. Communication

   The monitoring and control system website interface communicates with the sensing domain via the internet network using several standard HTTP methods to perform:

a. Sending actuator flag data to database to control actuator.
b. Downloading sensor reading data from the database.

3. Sensor Measurement Data Management

The website interface processes and displays sensor reading data from each node which is stored in the database in tabular, numerical, or graphic form.

Fig. 4. *Use Case Diagram of System*

To perform remote monitoring and controlling smart greenhouse, website interface has three actor, consist of administrator, verified user, and unverified user. Each actor has every access right for using the feature. The use case diagram from this system is shown in Fig 4. Each use case is described in Table 1.

Table 1: Use Case Description

Use Case	Actor	Description
Manage User	Administrator	In this use case, the administrator can see the user account listing and can delete the user account
Login	Administrator, verified user, and unverified user	The login use case aims to authenticate users and authorize user access rights.
Register	Administrator, verified user, and unverified user	Use case register aims to register new users into the system.
Log Out	Administrator, verified user, and unverified user	Use case log out aims to delete all sessions on the system.
Accessing Profile	Administrator, verified user, and unverified user	The use case to change the profile is a use case that aims to update the user's password.
Accessing Home	Administrator, verified user, and unverified user	The use case to change the profile is a use case that aims to update the user's password.
View sensor data history	Administrator, and verified user	The overall history of the data in the database is presented in graphical form. Only admins and verified users can view the data history.
View trending data chart	Administrator, and verified user	The trending graph presents the last 30 data read by the sensor in graphical form as well as the average of the last 30 data in numerical form.

Controlling actuator	Administrator, and verified user	The data in this use case is a state to turn the actuator on and off manually.

1. Administrator (Admin)
   Admin has access rights which include all existing access rights on the system. The admin has the right to observe and control the smart greenhouse, as well as manage users registered in the system.
2. Verified User
   Verified user is a user who has verified his email address when logging in for the first time after registration. Verified users can perform all monitoring and control functions on the system.
3. Unverified User
   Unverified users are users who have not verified their email address when logging in for the first time after registration. The email verification process aims to validate that the email address used is genuine. To prevent actions that have the potential to damage the system, as well as attacks from bots, unverified users can only access the home page on the system.

5. Results and Discussions

In this section, we provide the testing results of the system of smart greenhouse. The smart device is located at agreenhouse with a half-cylinder shape, as shown in Fig. 5. Thegreenhouse dimensions are width, length, and height with values of 2m, 4.5m, and 2.5m, respectively.

Fig. 5. *Inside the Smart Greenhouse*

In carrying out its function as a provider of smart greenhouse internet of things interfaces, the website interface has 8 main features, including login, register, profile, home, chart history, interface, data table, and user management. The login page is used for user authentication and authorization. The login page display is shown in Fig. 6.

Fig. 6. *Login Page Display*

After authenticating the login, the user will be directed to the home page, as shown in Fig. 7. On the home page, users can view the latest sensor reading data information and an average graph of the entire history of sensor reading data. Users can also switch to select smart greenhouse nodes.

Fig. 7. *Home Page Display*

The data graphic page display provides an interface in the form of a graph of reading the latest 30 sensor data. The graphical interface will display variable readings of air humidity and air temperature in the greenhouse, soil moisture, soil pH, and sunlight intensity. This page can only be accessed by admins and verified users. The display of the data graphic page view is shown in Fig. 8.

Fig. 8. *Data Graphics Page Display*

The entire history of sensor reading data is displayed in tabular form. The sensor data reading history table is displayed on the data table page as shown in Fig. 9.

Fig. 9. *Data Tables Page Display*

To control the actuators in the smart greenhouse, the website interface provides an interface page. The interface page is used to send actuator flags from the user to the remote smart greenhouse database. The interface page is shown in Fig. 10.

Fig. 10. *Interface Page Display*

Another feature is user account management. User account management includes displaying user account data and manipulating deleting user accounts.

Fig. 11. *User Management Page Display*

The user management view provides a user interface to view the list of users in Firebase Authentication and perform manipulations to remove users from Firebase Authentication, as shown in Fig. 11.

Website interface testing is divided into 2 tests, namely URL routing testing and website performance testing in handling concurrent users. This website interface testing aims to the the response of an HTTP method from each URL and to calculate the level of website performace in handling request from concurent users.

5.1 Website URL Routing Testing

Route testing aims to test the response of an application's URL routing address. The test results will provide an HTTP response from the URL under test. The expectation from this test is that all urls will give a redirect response (302) except the login (/) and register (/register) page URLs will give an OK (200) response. This is because all URLs (except login and register) will be protected by the authentication process and the authorization of user access rights so that they will be redirected to the login page (/) if the authentication process has not been carried out.

Table 2: URL Routes Testing Results

URL	Method	Description	Result
/	GET	HTTP response expectation is OK	200
/register	GET	HTTP response expectation is OK	200
/logOut	GET	HTTP response expectation is Found and Redirect	302
/profile	GET	HTTP response expectation is Found and Redirect	302
/user-list	GET	HTTP response expectation is Found and Redirect	302
/data-graphic	GET	HTTP response expectation is Found and Redirect	302
/graphic/data	GET	HTTP response expectation is Found and Redirect	302
/home	GET	HTTP response expectation is Found and Redirect	302
/home/data	GET	HTTP response is Found and Redirect	302
/interface	GET	HTTP response expectation is Found and Redirect	302
/actuator-states	GET	HTTP response expectation is Found and Redirect	302
/data	GET	HTTP response expectation is Found and Redirect	302
/data-table	GET	HTTP response expectation is Found and Redirect	302

The results of the URL routing test with the GET method obtained the results of 13 tested URL routes that met the expectations in the test case. As shown in Table 2, the login (/) and register (/register) routes successfully returned 200 responses. Routes (/logOut), (/profile), (/user-list), (/data-graphic), (/graphic/data), (/home), (/home/data), (/interface), (/actuator-states), (/data), and (/data-table) successfully returned a 302 response.

5.2 Website Performance Testing

Website performance testing aims to measure the level of website performance in handling HTTP method requests from users. Testing is done using JMeter. Jmeter is an application that is used to test the load of a website. Testing is done by varying the number of threads (users). The results of website performance testing are summarized in Table 3.

Table 3: Website Performance Testing Results

No	Number of *User*	Number of Request Sample	Percentage Failed Request Sample	Percentage Success Request Sample	Average Response Time (ms)
1	1	33	24,24%	75,86%	1453
2	5	165	35,15%	64,85%	2345
3	10	330	33,03%	66,97%	6015
4	15	495	97,37%	2,63%	1445

Based on the results of website performance testing, the website interface of the smart greenhouse monitoring and control system shows that the system has the maximum ability to handle HTTP requests from 15 users simultaneously. This is evidenced in the fourth iteration with the number of users 15, the percentage value of the sample request success is close to 0%, worth 2.63%.

6. Conclusions

A website interface for smart greenhouse monitoring and controlling system can built IoT-based website reliably. In testing the website URL routing with the GET method, no errors were found in the website routes design program. All URLs, except root (/) and register, managed to give a 302 response or found and redirect as expected on testing and a 200 response on URLs (/) and (/register). The smart greenhouse monitoring and control system website has the maximum ability to handle HTTP requests from 15 users simultaneously. Based on the results of testing the website interface functionality and URL routing, the website interface subsystem of the smart greenhouse monitoring and control system can provide a reliable IoT interface. the average error percentage is 6.23%.

7. References

[1] M. Dorais, A. P. Papadopoulos, and A. Gosselin, "Greenhouse Tomato Fruit Quality," in *Horticultural Reviews*, Oxford, UK: John Wiley & Sons, Inc., 2010, pp. 239–319.

[2] P. P. Reddy, "Greenhouse Technology," in *Sustainable Crop Protection under Protected Cultivation*, Singapore: Springer Singapore, 2016, pp. 13–22.

[3] S. U. Jomsuda Duangwongsa, Thepparath Ungsethaphand, Pradchayaporn Akaboot, Supakit Khamjai, "Real-time Water Quality Monitoring and Notification System for Aquaculture," *6th Int. Conf. Digit. Arts, Media Technol. 4th ECTI North. Sect. Conf. Electr. Electron. Comput. Telecommun. Eng.*, 2021, doi: https://doi.org/10.1109/ECTIDAMTNCON51128.2021.9425744.

[4] M. Thiyagarajan and C. Raveendra, "Integration in the physical world in IoT using android mobile application," in *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, Oct. 2015, pp. 790–795, doi: 10.1109/ICGCIoT.2015.7380570.

[5] P. Ghule and M. Kambli, "Web Based Environment Monitoring System Using IOT," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Apr. 2019, pp. 1338–1342, doi: 10.1109/ICOEI.2019.8862721.

[6] F. Zhang, X. Wan, T. Zheng, J. Cui, X. Li, and Y. Yang, "Smart Greenhouse Management System based on NB-IoT and Smartphone," in *2020 17th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, Nov. 2020, pp. 36–41, doi: 10.1109/JCSSE49651.2020.9268351.

International Journal of Engineering and Information Systems (IJEAIS)
ISSN: 2643-640X
Vol. 6 Issue 6, June - 2022, Pages: 7-14

Authors

Sumardi, born on November 11th, 1968 in Sukoharjo, Central Java. He graduated with a bachelor from Diponegoro University, Semarang, majoring in Electrical Engineering, Faculty of Engineering, in 1994. He got a master's degree in 1998, majoring in Instrumentation and Control Engineering Physics Program, Bandung Institute of Technology. He had followed some training including PLC training organized by OMRON as PLC producers and digital signal processing training organized by the Bandung Institute of Technology. Current business is as a lecturer and researcher at the Electrical Engineering Department, Faculty of Engineering, Diponegoro University, Semarang, Indonesia.

Aghus Sofwan, lives in Semarang, Indonesia. He received a Bachelor's degree in Electrical Engineering from Diponegoro University, Semarang, Indonesia in 1995. He received a Master's degree in Electrical Engineering from Gadjah Mada University, Yogyakarta, Indonesia in 2002. He accomplished his PhD, in 2016. He works as a lecturer at the Electrical Engineering Department, Faculty of Engineering Diponegoro University, Semarang, Indonesia from 1997 until the present. His research interest in internet of things and artificial intelligence.

Febrio Rizky Adhipratama born on Februari 21st, 2000 in Semarang, Central Java, Indonesia. He is a final year bachelor student at Diponegoro University, Semarang, Indonesia, majoring in Electrical Engineering, Faculty of Engineering. His research interest in Internet of Things and Web Development.