Green Intelligent Systems and Applications

Research Article
Volume 4(1), 2024, 29–40
https://doi.org/10.53623/gisa.v4i1.342

Internet of Things and Web-App-Based Data Accessibility and Management System for Chromameter Sensor Database

Faisal Samsuri[1,2], Joni Welman Simatupang[3,4]*

[1]Graduate Institute of Automation and Control, College of Engineering, National Taiwan University of Science and Technology, Taipei 10607, Taiwan
[2]Electro-Optical Systems Laboratory, National Taiwan University of Science and Technology, Taipei 10607, Taiwan
[3]Electrical Engineering Study Program, President University, Cikarang, West Java 17530, Indonesia
[4]Fiber Advanced Lightwave Communication and Optical Network Research Group (FALCON-RG), President University, Cikarang, West Java 17530, Indonesia

*Correspondence: joniwsmtp@president.ac.id

SUBMITTED: 29 October 2023; REVISED: 24 February 2024; ACCEPTED: 28 February 2024

ABSTRACT: Information technology, an integral part of most industrial activities, is essential for supporting the rapid and substantial processes of the industrial sector. A key component of this technology is the Internet of Things (IoT), which is extensively integrated into these systems. At PT Sugity Creatives, an analysis revealed impractical methods in the production process, such as manual data recording and input, as well as the use of stickers on the rear side of product bumpers. These stickers can be detached from the main body (car) for verification purposes. To improve these processes, a data accessibility and management system were incorporated into a Raspberry Pi-based chromameter sensor prototype. This integrated system is designed to collect and store data in a database and uniquely identify data using QR Codes. The system takes an average of 7.97 seconds to store data and generate a QR Code per entry. This includes a module processing time of 7.25 seconds per data point and a rapid transmission rate of 0.72 seconds, covering data recording and QR Code transmission from the chromameter prototype, with data sizes ranging between 700 to 750 bytes.

KEYWORDS: Internet of Things; web apps; data management; database; QR code

1. Introduction

According to Nikoloski [1], the implementation of information technology in industrial activities offers at least three concrete benefits: (1) significantly reducing operational costs, (2) enhancing the quality of products, production processes, and customer service, and (3) fostering innovative product development to compete in more potential markets. A commonly used technology in industrial processes today is the Internet of Things, often abbreviated as 'IoT' or “IOT” (which can be used interchangeably). Terminologically, IoT is a concept in information technology that enables the connection of various entities such as electronic devices, sensors, actuators, factories, or even entire cities to an interconnected network or the internet [2–4]. This allows for the realization of a smart system, characterized by independence and automation, minimizing the need for human intervention [5–8].

However, some companies still do not fully utilize information technology, particularly IoT, in their daily operations. An example is PT. Sugity Creatives, located in the MM2100 Industrial Area, Cikarang Barat, Bekasi. Observations of their manufacturing process revealed that data logging is performed manually, with entries written on paper and then input into computers. Additionally, product identification involves applying stickers to the rear side of products. This means that for identification purposes, products must be detached from the main body, a process that is relatively impractical and cumbersome. According to Michael Stangl's publication, "Development of a Web-Based Monitoring System for a Distributed and Modern Production" [9], web app functionality can extend beyond documents and articles to control and monitoring systems. This expansion is supported by advancements in tools such as Javascript, CSS3, and HTML5. Furthermore, web-based monitoring systems can adapt to various devices, not just desktops but also mobile phones, thanks to the Responsive Web Design (RWD) paradigm in web programming.

Inspired by this concept, a different implementation was used in the project titled 'Student Attendance System Prototype With IoT-Based on Fingerprint and Temperature Sensors in New Normal Era of COVID-19 Pandemic' by J. W. Simatupang and R. A. Q. Shihab [10]. Therefore, the prototype IoT and web-app-based data accessibility and management system developed is expected to improve industrial processes, particularly by accelerating the speed of recording and accessing data needed for product identification post-manufacturing.

2. Materials and Methods

The design of the data accessibility and management system is segmented into several components: a database for recording and storing data from the sensor prototype via the Raspberry Pi micro-computer; a web app for user-side data review; and a communication module facilitating interaction between the Raspberry Pi, serving as the client, and the PC, functioning as the server. The fundamental algorithm of this concept is illustrated in Figure 1.

2.1. Database.

A database is an organized collection of data, meticulously structured to transform raw data into useful information [11]. It serves as a foundational technology supporting data management processes. Within the PC, the database is stored on a server and is accompanied by a set of programs known as Structured Query Language (SQL), used for inputting and manipulating data. A database can also be described as a compilation of various tables. For instance, if a user intends to create a database for a monitoring system incorporating two different sensors, it would be efficient to establish a single database containing two distinct tables, each identified by a specific name. This setup is illustrated in Figure 2.

Referring to Figure 1 above, the software system is a web-app-based structure integrated with the initially available sensor prototype. The construction of the data accessibility and management system, as indicated by the dashed line in the figure, begins with parsing and arranging the data through the communication module. This module facilitates communication on the client side, acquiring data via the prototype sensor. The sensor, configured to operate in conjunction with the Raspberry Pi micro-computer, is capable of transmitting data to the PC, designated as the data center, using the database system. In this research work, a database titled 'raspi_database' was established to store data from the sensor system prototype, utilizing the table 'as7341_sensor_data'. Concurrently, the table named

'dht11_sensor_data' served as a test platform for the system, employing the Arduino Uno Module with the DHT11 Temperature and Humidity Sensor before its integration with the Raspberry Pi featuring the AS7341 Chromameter Sensor.

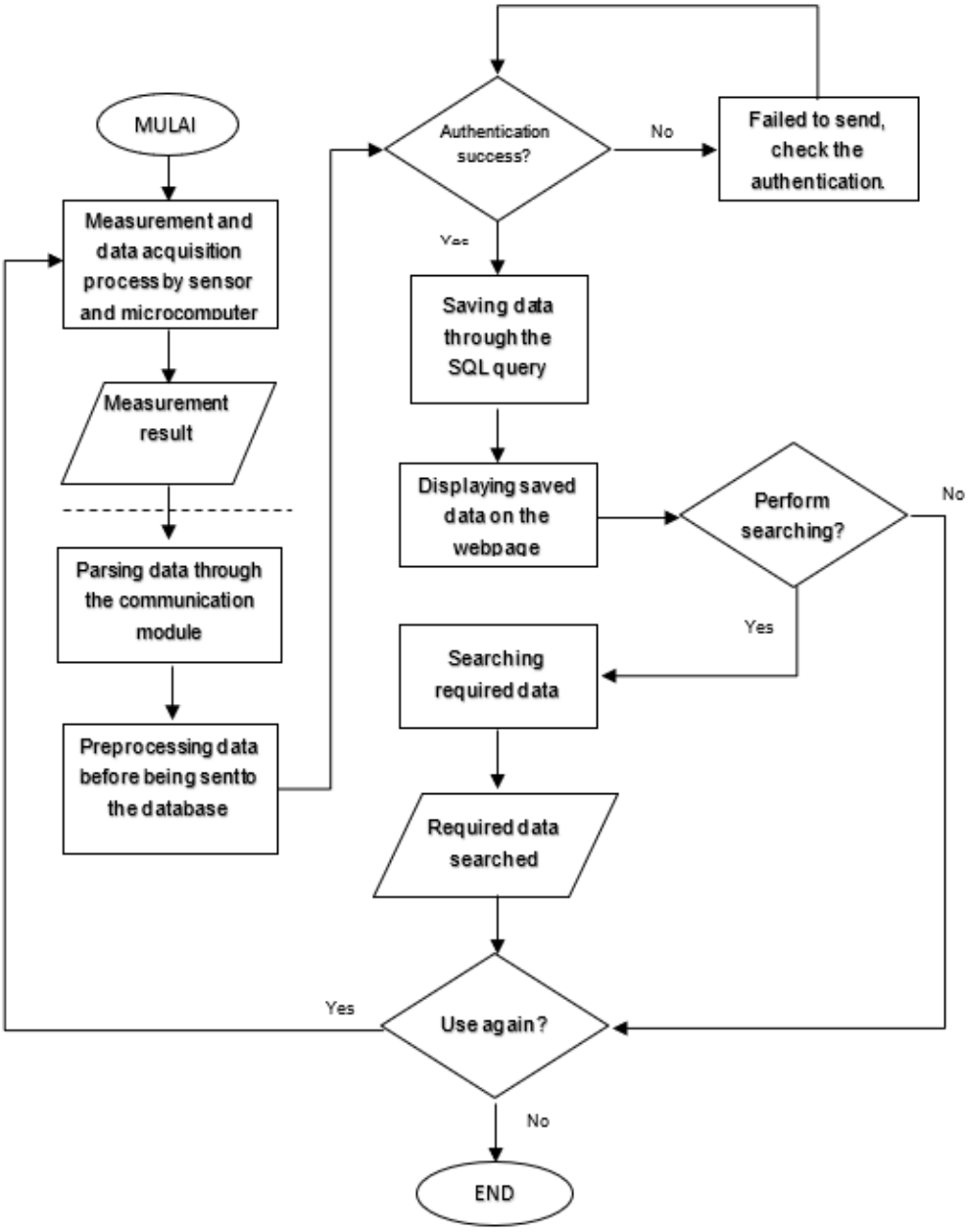


Figure 1. The planned algorithm of the system.

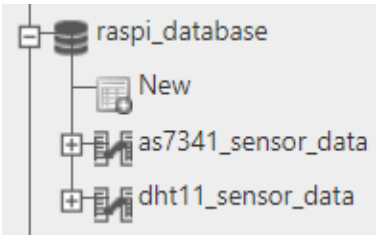


Figure 2. Database created in PHPMyAdmin.

2.2. Web-app: front-end developing.

The front-end serves as the interface section, enabling users to access control functions and display data requested from the server [12]. The code for building the front end can typically be viewed and edited within a web browser, allowing for real-time review and temporary modifications. However, the code reverts to its original form upon page reload or refresh. The languages used in constructing the front end of a web app include HTML, which structures the layout and elements; CSS, which designs the visual aesthetics; and JavaScript, which enables dynamic responses such as data searches, QR Code generation, camera usage, and saving specific files indexed in HTML elements. The basic algorithm underlying the front-end's working principle is depicted in Figure 3.

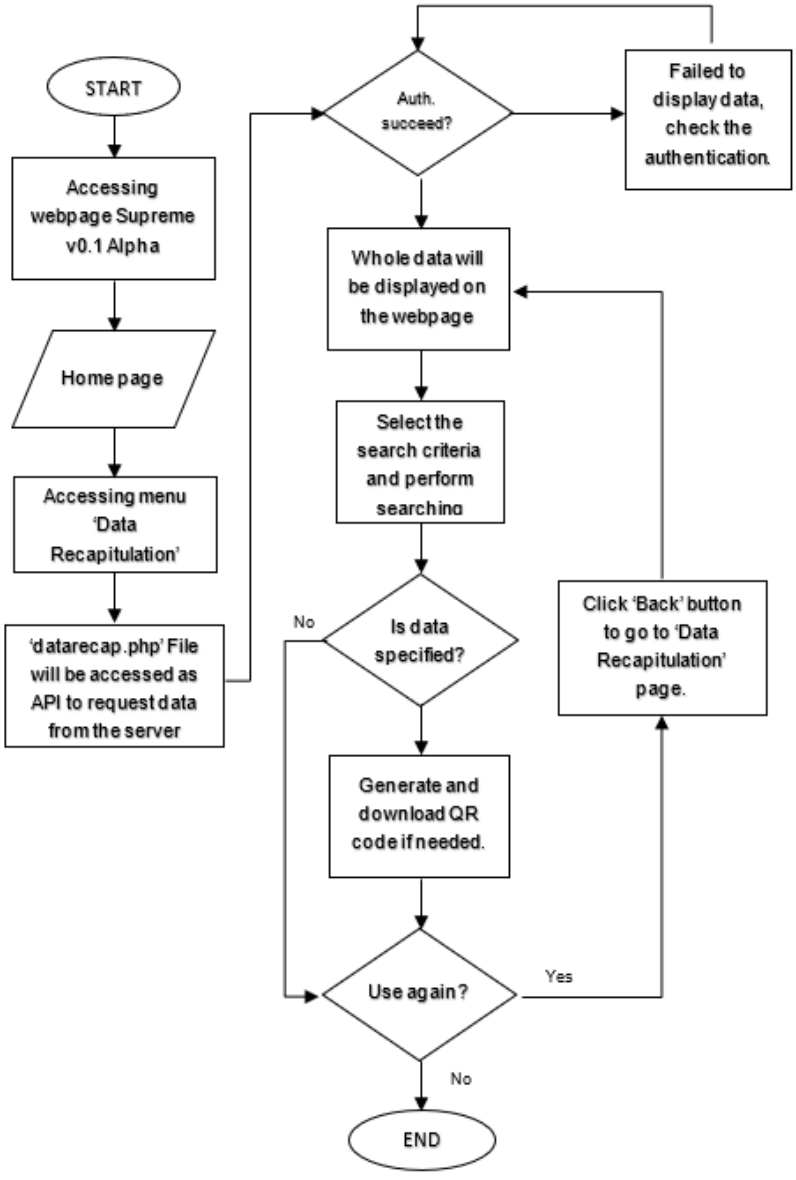


Figure 3. Front-end working principle algorithm.

2.3. Web-app: back-end developing.

The back-end functions as the 'behind-the-scenes' component of the system. It is developed using server-side scripting, a type of programming language focused on the functionality, logic, control, and behavior of a system [13]. Unlike front-end programming languages, back-end

languages are not visible to the web browser; they remain concealed from the user and are accessible only to the server. Common back-end languages include PHP, Java, Python, and Node.js, all of which process requests and commands initiated from the front end. For the development of the Supreme v0.1 Alpha web app's back-end, PHP was chosen. This decision was influenced by PHP's status as a standard language in XAMPP for constructing web servers. PHP is also widely favored among programmers due to its extensive libraries, which greatly facilitate web app development. An example of such utility is a library that enables direct API connection to the MySQL server through a built-in function named 'mysqli'. The fundamental algorithm of the web app's back-end operation is depicted in Figure 4.

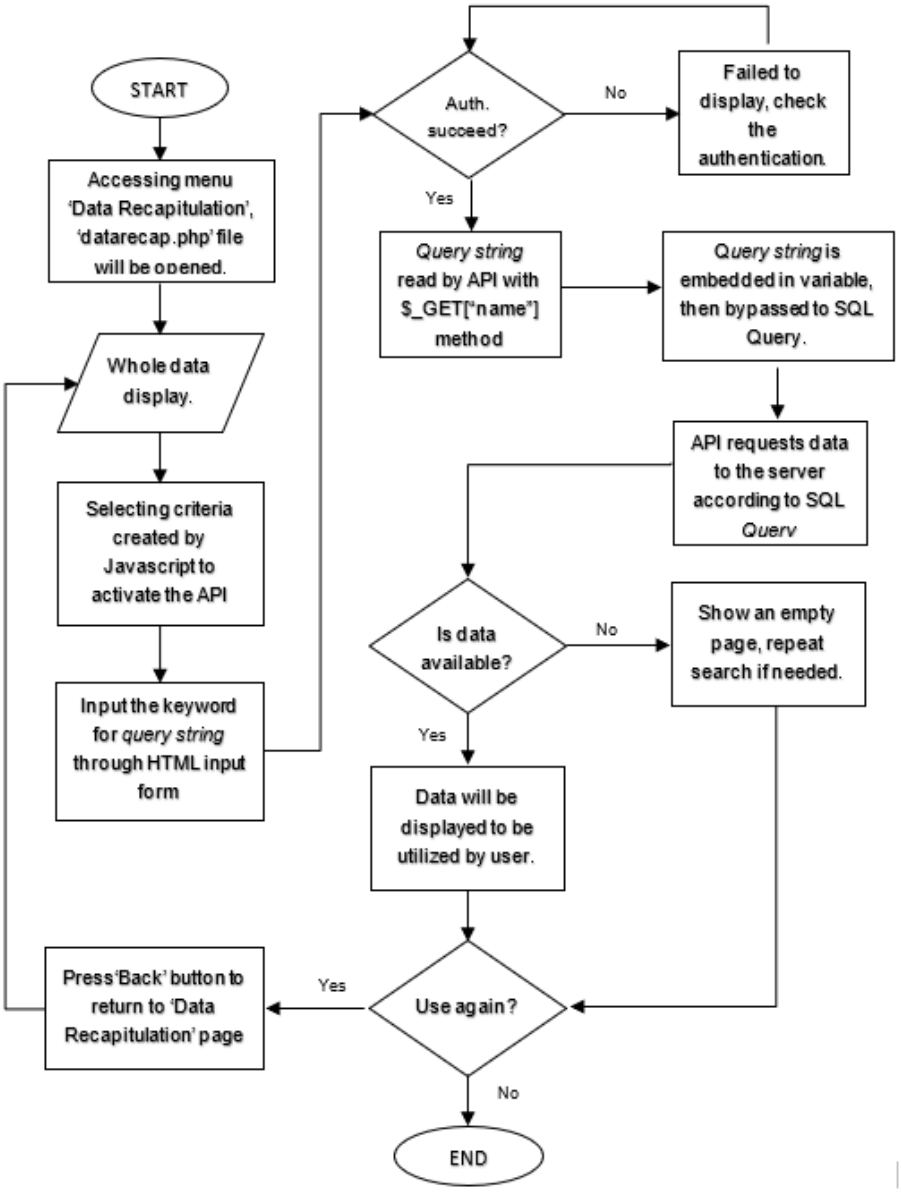


Figure 4. Back-end working principle algorithm.

2.4. Chromameter sensor system prototype.

The chromameter sensor system prototype was developed independently from this researcg work [14]. It comprises two primary components: a Raspberry Pi-4 micro-computer, serving as the main control unit, and an AS7341, which functions as the principal chromameter sensor for measuring color gradients. This prototype is designed to automate the gradient testing

process for car spare parts at the manufacturing site, particularly for car bumpers. The physical design of the prototype is illustrated in Figure 5 and Figure 6, showing the front and top views, respectively.

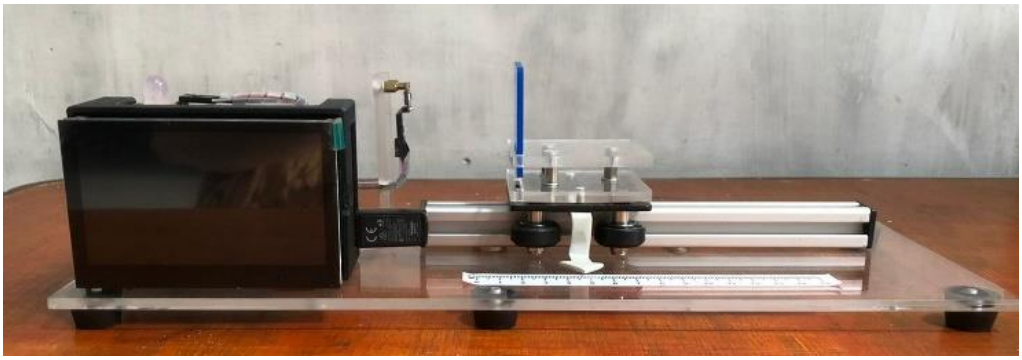


Figure 5. Chromameter sensor prototype (front-view).

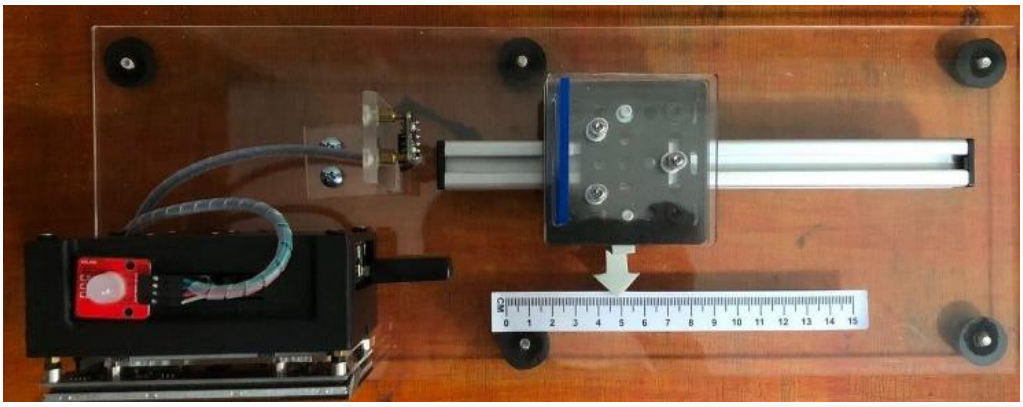


Figure 6. Chromameter sensor prototype (top-view).

Figures 5 and 6 detail the components of the prototype. Item 1 is the LCD attached to the Raspberry Pi micro-computer, which serves as the main control unit for displaying measurement results. Item 2 is the holder for the sample object being measured. Item 3, the LED indicator, features three colors, each representing a specific message. Item 4 is the AS7341 sensor. Item 5, the ruler, measures the distance between the object and the sensor. This prototype is designed to measure the color gradient of a specific color on spare parts. The results are processed by a Python module tailored for the sensor, and automation is achieved through an algorithm written in Bash language. The module segregates the measurement results into discrete numbers, storing them across 10 channels, with each channel corresponding to a specific criterion. These channel numbers are then recorded in a '.csv' file as the final step.

The prototype has several areas for development, such as storing measurement results in local memory. However, considering security and memory management, modern integrated automation systems employ databases and servers for large-scale data management and control. Furthermore, according to a case study, practical product identification is necessary. Therefore, all essential information is affixed to a special sticker placed on the back of the car bumper. For guarantee claims and service purposes, the sticker is reverified by detaching the bumper from the car. Consequently, further advancements, like developing a database and web server that can connect to the prototype via the IoT concept, are proposed to enhance functionality. This project also includes a web page for product identification and data review,

where each piece of information is embedded in a QR Code, instantly generated following the measurement.

2.5. Communication module and QR code.

A communication module is a script developed in a specific programming language, facilitating interaction between the server and the client. In this prototype, the communication module is programmed using both PHP and Python. The module programmed in PHP functions primarily to record data in the database. It is also referred to as an API, an acronym for Application Programming Interface. An API, in this context, is a software tool used by the client (the prototype) for establishing a connection to the server [15]. The web app employs two types of APIs: one that transmits data to the server, and another that retrieves data from the server. The data-transmitting API utilizes the POST HTTP Method, while the data-receiving API employs the GET HTTP Method [16, 17]. Figure 7 below illustrates the operation of these APIs.

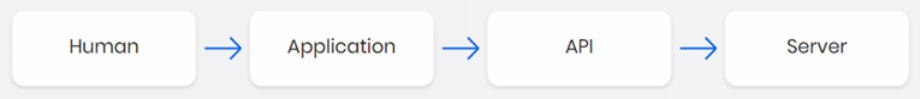


Figure 7. An API working principle.

The QR Code is generated instantly once the measurement concludes, using a generator. This client-side module or generator produces the QR Code and subsequently transmits it to the server computer. The generator establishes contact with the listener via the computer's IP address. In response, the listener identifies the client's IP address and initiates a 'handshaking' process. Once the connection is established, the server-side module, or listener, retrieves the message from the generator and downloads the QR Code to the computer. Figure 8 below shows the QR Code listener in its idle state.

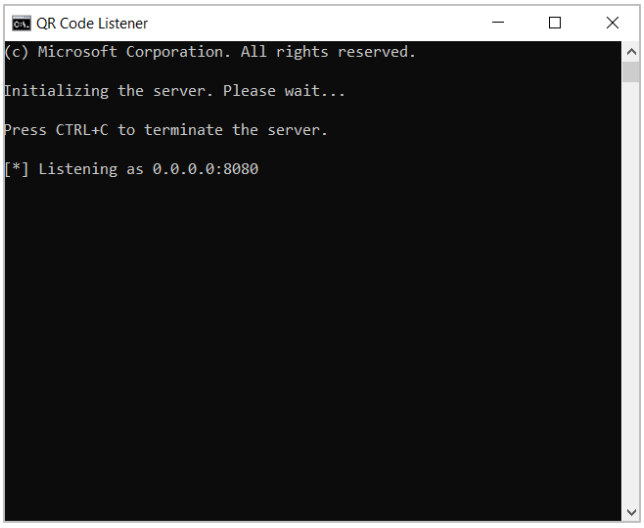


Figure 8. QR code listener.

3. Results and Discussion

Immediately following the completion of the gradient test, the measurement results are stored in a '.csv' file. An example of these results is presented in Table 1. The latest content is transmitted to the database soon after the completion of the measurement. This transmission

process is displayed on the Raspberry Pi screen. The resulting display is depicted in Figure 9. As illustrated in the figure 9, the server responded with HTTP Code 200, signaling that communication between the client and the server was successful. Subsequently, the data was stored in the database, as depicted in Figure 10. Each data point corresponds to a specific product or spare part. Following the integration of the web server with the sensor prototype, the system underwent 300 test cycles for data sampling. This was to assess the durability of the communication module on both the client and server sides. The tests aimed to ensure there were no repeated data transmissions, that data was transmitted accurately, and that the transmission speed was relatively fast compared to the company's previous method.

Table 1. Measurement result example.

Date & Time	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8	Data 9	Data 10
06/17/2022 15:44	4890	3121	2272	1283	4874	3105	2266	1277	17308	599
06/17/2022 15:45	4891	3122	2273	1284	4875	3106	2267	1278	17300	600
06/17/2022 15:46	4892	3122	2274	1284	4876	3106	2268	1278	17300	600
06/17/2022 15:47	4893	3125	2277	1289	4877	3105	2268	1273	17302	601
06/17/2022 15:48	4894	3125	2276	1289	4878	3110	2272	1275	17304	602

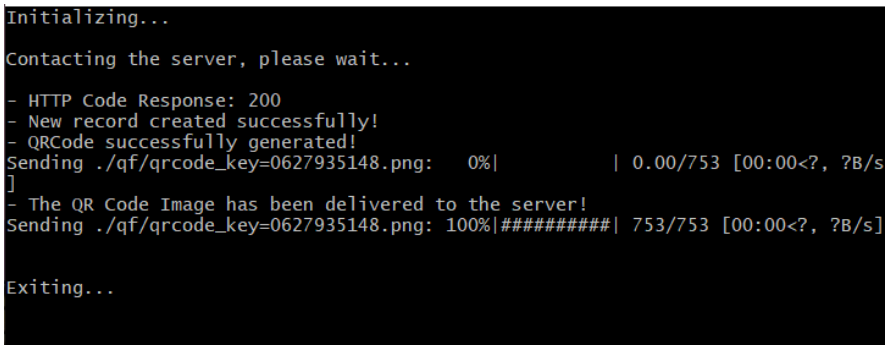


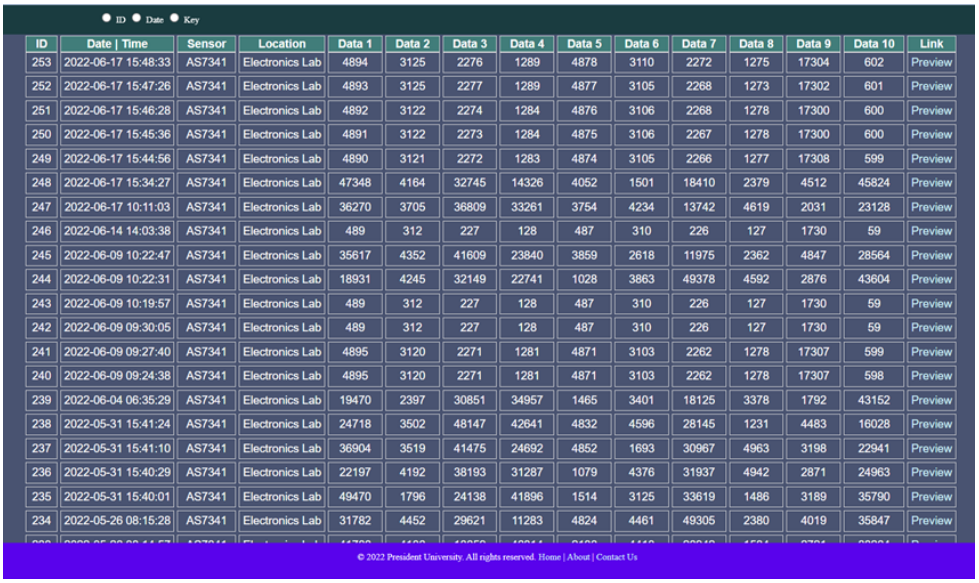
Figure 9. Transmitting process to the database.

4890	3121	2272	1283	4874	3105	2266	1277	17308	599	2022-06-17 15:44:56
4891	3122	2273	1284	4875	3106	2267	1278	17300	600	2022-06-17 15:45:36
4892	3122	2274	1284	4876	3106	2268	1278	17300	600	2022-06-17 15:46:28
4893	3125	2277	1289	4877	3105	2268	1273	17302	601	2022-06-17 15:47:26
4894	3125	2276	1289	4878	3110	2272	1275	17304	602	2022-06-17 15:48:33

Figure 10. Recorded data in the database.

Field observations indicated that the operator typically required around 15 minutes to complete the entire process. This included conducting the gradient test or measurement, manually recording the results on a data recapitulation sheet, typing the data into a computer, and affixing the data on a sticker to be applied to the rear side of the bumper. In contrast, the total time required for data recapitulation by the data accessibility and management system averaged about 7.97 seconds per data point. This time included 7.25 seconds for the module's processing and 0.72 seconds for QR code generation and transmission to the server computer.

Data review can be performed via the web app, eliminating the need for direct access to the database server. The database server is accessed solely by the database administrator for administrative tasks, such as editing and deleting data, as shown in Figure 11.

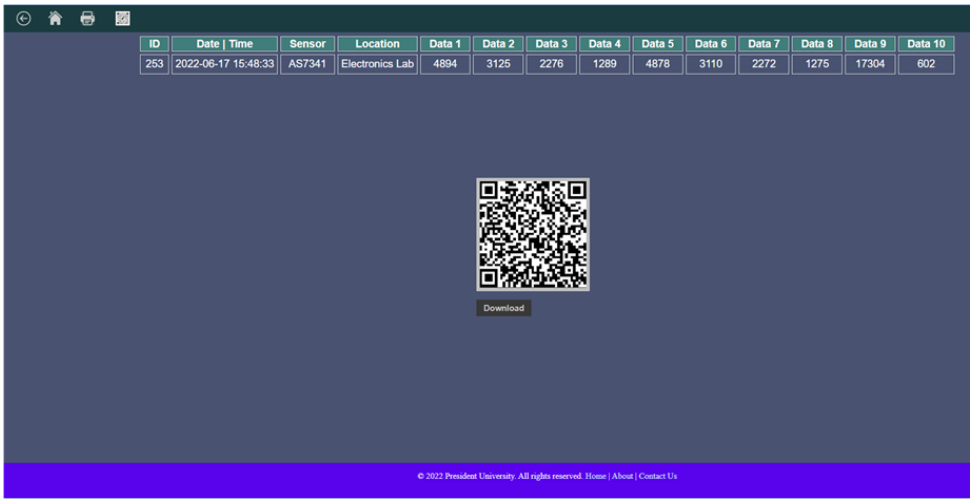


ID Date Key

ID	Date | Time	Sensor	Location	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8	Data 9	Data 10	Link
253	2022-06-17 15:48:33	AS7341	Electronics Lab	4894	3125	2276	1289	4878	3110	2272	1275	17304	602	Preview
252	2022-06-17 15:47:26	AS7341	Electronics Lab	4893	3125	2277	1289	4877	3105	2268	1273	17302	601	Preview
251	2022-06-17 15:46:28	AS7341	Electronics Lab	4892	3122	2274	1284	4876	3106	2268	1278	17300	600	Preview
250	2022-06-17 15:45:36	AS7341	Electronics Lab	4891	3122	2273	1284	4875	3106	2267	1278	17300	600	Preview
249	2022-06-17 15:44:56	AS7341	Electronics Lab	4890	3121	2272	1283	4874	3105	2266	1277	17308	599	Preview
248	2022-06-17 15:34:27	AS7341	Electronics Lab	47348	4164	32745	14326	4052	1501	18410	2379	4512	45824	Preview
247	2022-06-17 10:11:03	AS7341	Electronics Lab	36270	3705	36809	33261	3754	4234	13742	4619	2031	23128	Preview
246	2022-06-14 14:03:38	AS7341	Electronics Lab	489	312	227	128	487	310	226	127	1730	59	Preview
245	2022-06-09 10:22:47	AS7341	Electronics Lab	35617	4352	41609	23840	3859	2618	11975	2362	4847	28564	Preview
244	2022-06-09 10:22:31	AS7341	Electronics Lab	18931	4245	32149	22741	1028	3863	49378	4592	2876	43604	Preview
243	2022-06-09 10:19:57	AS7341	Electronics Lab	489	312	227	128	487	310	226	127	1730	59	Preview
242	2022-06-09 09:30:05	AS7341	Electronics Lab	489	312	227	128	487	310	226	127	1730	59	Preview
241	2022-06-09 09:27:40	AS7341	Electronics Lab	4895	3120	2271	1281	4871	3103	2262	1278	17307	599	Preview
240	2022-06-09 09:24:38	AS7341	Electronics Lab	4895	3120	2271	1281	4871	3103	2262	1278	17307	598	Preview
239	2022-06-04 06:35:29	AS7341	Electronics Lab	19470	2397	30851	34957	1465	3401	18125	3378	1792	43152	Preview
238	2022-05-31 15:41:24	AS7341	Electronics Lab	24718	3502	48147	42641	4832	4596	28145	1231	4483	16028	Preview
237	2022-05-31 15:41:10	AS7341	Electronics Lab	36904	3519	41475	24692	4852	1693	30967	4963	3198	22941	Preview
236	2022-05-31 15:40:29	AS7341	Electronics Lab	22197	4192	38193	31287	1079	4376	31937	4942	2871	24963	Preview
235	2022-05-31 15:40:01	AS7341	Electronics Lab	49470	1796	24138	41896	1514	3125	33619	1486	3189	35790	Preview
234	2022-05-26 08:15:28	AS7341	Electronics Lab	31782	4452	29621	11283	4824	4461	49305	2380	4019	35847	Preview

Figure 11. Data review via the web page

The system also includes a search menu with criteria-based options for specific reviews. These criteria are standard in database management and include options like data recapitulation through the ‘Date’ button, product identification number via the ‘ID’ button, or a unique identification key through the ‘Key’ button. In addition to the automatic QR Code generation post-measurement, as depicted in Figure 9, there is also an option for manual QR Code generation. This can be done by selecting specific data representing a particular product through the provided link, as shown in Figure 11, or via the search menu using the ‘ID’ criterion. Following this selection, the QR Code generator page will appear, as illustrated in Figure 12. The QR Code is generated upon clicking the ‘Generate’ menu, which is the fourth option located at the upper left side of the page.



ID	Date | Time	Sensor	Location	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8	Data 9	Data 10
253	2022-06-17 15:48:33	AS7341	Electronics Lab	4894	3125	2276	1289	4878	3110	2272	1275	17304	602

Download

Figure 12. Manual QR Code generation

After generating the QR Code, it can be verified using the QR Scanner feature available in the web app. This scanner, adapted and modified from the open-source web ScanApp.org, is primarily developed using Javascript. The result of the QR scan will display a web address that links to specific data corresponding to a product previously measured by the sensor system prototype. The web app developed for this project is accessible via smartphones to facilitate the scanning process. This feature can be accessed through the 'Identification' menu on the homepage, as illustrated in Figure 13.

The observed results indicate that the system operates effectively and efficiently. The integration of IoT and web technologies with the sensor prototype has significantly reduced the time required for data recapitulation. Utilizing QR Codes for product identification is also practical, and applying them to the front side of the bumper does not compromise the product's aesthetics while facilitating scanning. The QR Codes can be printed at a minimum size of 15mm x 15mm. Therefore, considering the time effectiveness, process efficiency, and automation mechanism, the system is suitable for data accessibility and management needs. However, currently set up in a localhost environment, future advancements could include web hosting to enable comprehensive online access of the system with public DNS, allowing universal over-the-air access.

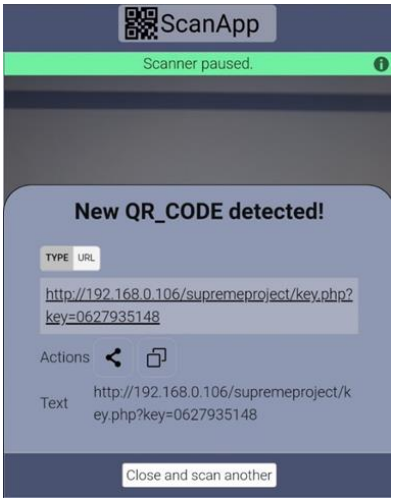


Figure 13. QR scanning on mobile web

4. Conclusions

The data accessibility and management system, when integrated into the sensor prototype, functions correctly. The communication modules, both on the server and client sides, operate effectively, enabling the recording and storage of data on the server and facilitating data display through the web app. This can be done either through direct web access or via links embedded in QR Codes. According to tests conducted on the system, it required approximately 7.97 seconds per data entry, based on an average taken over 300 tests. This processing time is notably quicker than the method previously employed by the manufacturer.

Acknowledgments

The authors acknowledge and thank the Research Institute and Community Service (RICS) of President University for financially supporting this research work with the incentive scheme for scientific publications, both national or international journals and conference proceedings.

Competing Interest

There is no competing interest between the authors.

References

[1] Nikoloski, K. (2014). The Role of Information Technology in the Business Sector. *International Journal of Science and Research, 3*, 303–309. https://ijsr.net/archive/v3i12/U1VCMTQzMjA=.pdf

[2] Simatupang, J.W.; Lubis, A.M.; Vincent. (2022). IoT-Based Smart Parking Management System Using ESP32 Microcontroller. 2022 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), 305–310. https://doi.org/10.23919/EECSI56542.2022.9946608.

[3] Sucipta, I.; Simatupang, J.W.; Kaswandi, C.; Purnama, I. (2021). Prototipe Pemantauan Tetes Cairan Infus Berbasis Iot Terkoneksi Perangkat Android. *Jurnal Teknik Elektro Universitas Mercu Buana*, *12*, 113–119. http://doi.org/10.22441/jte.2021.v12i3.003

[4] Rospawan, A.; Simatupang, J.W.; Purnama, I., (2022). IOT Application For Conveyor Motor Load Current And Temperature Monitoring Device for Factory Acceptance Test in Industrial Application. *Jurnal ELTIKOM: Jurnal Teknik Elektro, Teknologi Informasi Dan Komputer*, *6*, 152–156. https://doi.org/10.31961/eltikom.v6i2.553.

[5] Ali, Z.H.; Ali, H.A.; Badawy, M.M. (2015). Internet of Things (IoT): Definitions, Challenges, and Recent Research Directions. *International Journal of Computer Application*, *128*, 37–47. https://doi.org/10.5120/ijca2015906430.

[6] Kumar, S.; Tiwari, P.; Zymbler, M. (2019). Internet of Things is a revolutionary approach for future technology enhancement: a review. *Journal of Big Data*, *6*, 111. https://doi.org/10.1186/s40537-019-0268-2.

[7] Misra, S.; Roy, C.; Mukherjee, A. (2021). Introduction to Industrial Internet of Things and Industry 4.0, 1st ed.; CRC Press: Boca Raton, USA. https://doi.org/10.1201/9781003020905.

[8] Vermesan, O.; Friess, P. (2014). Internet of Things—From Research and Innovation to Market Deployment. River Publishers: Aalborg, Denmark. https://doi.org/10.1201/9781003338628.

[9] Stangl, M.; Pielmeier, J.; Berger, C.; Braunreuther, S.; Reinhart, G. (2016). Development of a Web-Based Monitoring System for a Distributed and Modern Production. *Procedia CIRP*, *52*, 222–227. https://doi.org/10.1016/j.procir.2016.07.073.

[10] Simatupang, J.W.; Shihab, R.A.Q. (2022). Student Attendance System Prototype With IoT-Based on Fingerprint and Temperature Sensors in New Normal Era of COVID-19 Pandemic. *Jurnal IPTEK*, 26, 99–106. https://doi.org/10.31284/j.iptek.2022.v26i2.3000.

[11] Derclaye, E. (2005). What is a Database?. *The Journal of World Intellectual Property*, *5*, 981–1011. https://doi.org/10.1111/j.1747-1796.2002.tb00189.x.

[12] Oluwatosin, H.S. (2014). Client-Server Model. *IOSR Journal of Computer Engineering (IOSR-JCE), 16*, 67-71. http://doi.org/10.9790/0661-16195771

[13] Iskandar, T.F.; Lubis, M.; Kusumasari, T.F.; Lubis, A.R. (2019). Comparison Between Client-Side and Server-Side Rendering in Web Development. *IOP Conference Series: Material Science and Engineering*, *801*, 012136. https://doi.org/10.1088/1757-899X/801/1/012136.

[14] Anam, Fajar C.; Samsuri, F.; Simatupang, J.W. (2023). Prototipe Chromameter untuk Deteksi Bumper Berbasis Raspberry Pi-4 dan Sensor AS7341. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, *11*, 677–690. https://doi.org/10.26760/elkomika.v11i3.677.

[15] Laksito, A.D. (2019). API Gateway Menggunakan SlimPHP pada Aplikasi Kantin Amikom. *IPTEK-KOM*, *21*, 31–42.

[16] Hogie, K.; Criscuolo, E.; Parise, R. (2005). Using Standard Internet Protocols and Applications in Space. *Computer Sciences Corp*, *47*, 603–650. http://doi.org/10.1016/j.comnet.2004.08.005

[17] Guilin, Z.; Wang, Z.; Jin, S. (2013). Research And Application Of Digital Transmission Of Differential Resistance Sensor. *Chinese Control and Decision Conference (CCDC)*, *25*, 2046–2051. http://doi.org/10.1109/CCDC.2013.6561272