

Design and Construction of a Swimming Pool Reservation System Based on Next.js and REST API

Faturrahman^{1*}, Diah Aryani²

^{1,2}Esa Unggul University, Jakarta, Indonesia

*Corresponding author: fatur020302@student.esaunggul.ac.id

Article Information:

Received: 23 July 2025

Revised: 12 Februari 2026

Accepted: 01 July 2026

DOI: 10.33050/icit.v12i2.4008

ABSTRACT

This research project aims to improve the manual reservation method still used at the Raihan Islamic Swimming Pool in Tangerang by creating a web-based pool reservation and monitoring system using Next.js technology and REST API. Issues such as slow service, inaccurate record keeping, and a lack of information transparency often reduce customer satisfaction and operational effectiveness. This research adopted an Extreme Programming approach in the system development process, collecting data through literature reviews, interviews, and observations. The results of the implementation based on black box testing indicate that this system is able to provide direct access to information and automate the booking and payment process through Midtrans, so that the entire reservation process can be carried out more efficiently, quickly, and integrated without significant obstacles. The use of PWA, updateable news components, and integration with Google login also contribute to an improved user experience and convenience. Although it needs development in several areas, such as automatic cancellation features and reporting, this system is believed to be very useful for improving efficiency, control, and service quality.

Keywords—Online Reservation, Web Service, REST API, Next.js, Midtrans

1. INTRODUCTION

To increase operational efficiency, prevent scheduling conflicts, and improve user experience, IT applications in the form of reservation systems allow users to book services online [1]. Azizi and colleagues [2] revealed that implementing an online reservation system can drastically reduce administrative burdens, waiting times, and the need for manual documents in public service sectors such as hospitals. The success of digital reservation systems in improving operational efficiency and administrative services in various fields has been proven. Thanks to the development of internet technology, online booking platforms have become an important element in providing digital services. Such platforms are essential to improve operational effectiveness and user satisfaction in various fields, including hospitality, logistics, transportation, and public sports facilities [3], [4]. It is expected that the implementation of the Next.js framework with Server-Side Rendering (SSR) and Client-Side Rendering (CSR) strategies will improve application performance and user experience, due to faster initial load times and better interaction [5], [6].

In the context of technological advancements, connecting REST APIs with the latest JavaScript frameworks such as Next.js provides a customizable and scalable structure for creating responsive and interactive booking systems [7]. Furthermore, booking systems that implement a data-driven approach enable service providers to directly track customer preferences, supporting

more adaptive service personalization [8]. Therefore, booking systems are crucial in the digital age, improving efficiency, flexibility, and service quality while automating administrative processes.

Building efficient, inclusive, and user-friendly systems to address challenges related to performance, sustainability, and limited access to technology is a current challenge in developing online reservation systems. Complex reservation systems can be overwhelming for users with limited device and network access [6]. While effective simulations and robust data-driven decision-making systems are needed, implementing an overbooking model at a container port can improve service performance [3]. From a consumer behavior perspective, online reviews, usability, and service quality significantly influence customers' decisions to use a reservation system. This suggests that a system, even if technically superior, may fail to attract users if it is not designed with the perceived quality of the user experience in mind [9].

While online booking systems built with Next.js are expected to drastically improve the performance and accessibility of web applications [5], [6], their implementation is rare in environments with limited technological support. Research conducted by Talath et al. [4] highlighted the need for a simple and user-friendly booking system for sports fields for individuals inexperienced with technology. However, this study did not analyze the technical performance of the system under limited network conditions; the focus was on the user interface design and modular architecture. Furthermore, while Srivastava et al. [6] investigated key features of Next.js such as server-side rendering, pre-rendering, and dynamic routing, their study did not address the practical application of Next.js in small contexts such as community sports facilities or schools. Thorough studies on the efficiency of Next.js in situations with limited bandwidth, modest hardware, or limited technical capabilities are scarce as previous research has generally highlighted the framework's application in large commercial scenarios. Therefore, additional research is needed to assess the effectiveness, usability, and performance of reservation systems built with Next.js in a local community with limited resources.

Based on information from the aforementioned sources and data obtained through direct interviews with staff and visitors in April 2025, the manual reservation system at the Raihan Islamic Swimming Pool in Tangerang presented several issues. These included recording errors, repeated bookings, and delays in response due to the manual schedule checking process. Visitors were required to make reservations and cancellations directly, while staff had to manually record everything in a book. This decreased service efficiency led to customer dissatisfaction. Visitors chose to leave due to delays in communication and a lack of transparency. Therefore, to improve productivity and user experience, a web-based reservation system using Next.js and the REST API was needed.

2. LITERATURE REVIEW

Table 1. Literature Review

No.	Research Title	Research Title	Research Title	Research Title
1.	Pemanfaatan Nextjs Dan Mongodb Dalam Sistem Informasi Web Manajemen Data Beras Pada Ud Sri Utami[10]	Using the Waterfall method, system development with Next.js (frontend and SSR) and MongoDB (NoSQL database).	This study shows that Next.js is useful for developing web-based systems for transaction and sales management. It provides high performance, responsiveness, and good data handling for e-commerce applications.	The system does not yet have report visualization features (graphs or charts) and real-time notifications which could be very useful for admins or consumers.

2.	Implementasi Rest Api <i>Web service</i> Dalam Membangun Aplikasi <i>Multipatform</i> Untuk Usaha Jasa [11]	Using the Waterfall method, the system implements REST API to connect all platforms.	This study concludes that REST API Web Service is an efficient solution for multipatform applications because it allows real-time data access from various devices, increases service efficiency, and speeds up the search and ordering process with flexible and responsive data management.	The research doesn't address authentication or authorization, which are commonly required in RESTful APIs. User interactions are still based on traditional request-response, even though features like real-time status updates could be more optimal using WebSockets.
3.	Pengembangan Sistem <i>Booking Online</i> Pada Cokro Barbershop Berbasis <i>Website</i> [8]	Using the Waterfall Method, using the PHP and MySQL programming languages	This study demonstrates how a website-based online booking system can increase the efficiency of barbershop services, making them more convenient for clients and simpler for company owners to administer, through features including online booking, digital queue management, and automated reporting.	There are no additional features like automatic notifications, booking confirmations, payment systems, or employee management. The app is also not yet available as a mobile app (Android/iOS), which would improve user experience.
4.	Analisa Dan Perancangan Sistem Informasi <i>Booking</i> Antrian Pelayanan Pada Klinik Medika Sainika Berbasis <i>Website</i> [12]	Using the Waterfall Model approach, the Black Box Testing method is used to ensure that the functionality meets the requirements.	A website-based online queue booking system increases the efficiency and quality of healthcare services by lowering wait times, streamlining operations and raising customer satisfaction through features like online reservations, automatic scheduling and queue notifications.	The system only covers registration and booking; it does not yet support electronic payment features that would enhance patient convenience. Furthermore, the system is website-based and is not yet available as a mobile app, which would allow patients to access it more flexibly on mobile devices.
5.	Pengembangan Sistem Informasi Pemesanan Barang Berbasis <i>Web</i> [9]	Using the Agile Software Development approach	This study demonstrates how features such as online ordering, real-time stock monitoring and automated transactions can improve operational efficiency, reduce errors in stock and transaction	While there are checkout and reporting features, the system does not yet include online payment methods or payment gateway integration, which are

			management, and enhance the effectiveness and transparency of web-based booking systems.	essential in a modern ordering system.
--	--	--	--	--

The literature review reveals several challenges in developing online booking systems. Modern frameworks like Next.js, which excel in performance, efficiency, and scalability, have not been widely used in previous research. Important features such as digital payments, booking confirmations, automated notifications, and support for mobile apps or progressive web apps (PWAs) have also not been widely implemented. Furthermore, the use of REST APIs is not effective for cross-platform development or security. To create an efficient, secure, and user-friendly system, this study recommends developing an online booking system based on Next.js with a REST API and advanced features such as real-time availability checks, single sign-on, social media login, and payment gateway integration.

3. METHODOLOGY

3.1 Data Collection Methods

This study utilized various data collection methods to obtain comprehensive and in-depth information regarding the problems encountered in the swimming pool reservation system. Data collected through observation and interviews will provide a practical overview of the field conditions, while a literature review will provide a solid theoretical foundation for system design.

1. Direct observations were conducted at the Raihan Islamic Swimming Pool for three days, from April 7 to 10, 2025. The primary focus of the observations was the activities of staff in recording reservations and transactions, as well as the manual reservation procedures for visitors. The weaknesses of the manual method were revealed through the collected data, indicating the need for a more efficient digital system.
2. Interviews were conducted with Muslim women who desired more privacy while using the pool facilities, as well as with pool staff and an elementary school teacher representing group reservations. The purpose of these interviews was to gain an understanding of their expectations, barriers, and experiences related to the reservation system and financial reporting, with the hope of better targeting and meeting user needs during the implementation process.
3. A literature review was conducted to strengthen the theoretical basis for system development. The literature analyzed included the Next framework. js, Extreme Programming development methodology, REST API concepts, and previous studies related to the implementation of web-based reservation systems. This study offers guidance in determining the most appropriate technology strategy for the system's needs.

3.2 System Development Methods

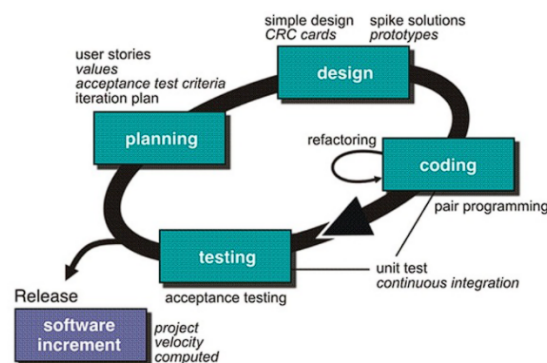


Figure 1. Extreme Programming Method

The system development method used in designing the website-based swimming pool booking and monitoring application in this study refers to the principles of extreme programming (XP). The following are the system development stages applied.

1. Planning: The planning stage is the initial phase in system development, identifying and analyzing user and system requirements [13]. This research collaborated with the Raihan Islamic Swimming Pool in Tangerang to gain a deeper understanding of the ongoing booking and monitoring process at Raihan Swimming Pool
2. Design: The design stage involves creating a UML (Unified Modeling Language) model to detail the system structure and interactions [14]. In the design stage, this study used UML structural diagrams consisting of Use Case Diagrams, Activity Diagrams, and Class Diagrams. This design provides a clear visual overview of the application architecture before entering the implementation phase.
3. Coding: The implementation stage involves converting the prepared design into computer-readable programming code [15]. In this stage, the researchers utilized a combination of JavaScript, Next.js, HTML, and CSS, along with other programming languages, to ensure the integrity and performance of the final product.
4. Testing: The testing stage on the Raihan Swimming Pool booking and monitoring system is carried out to ensure that the system operates according to requirements and is free from bugs or errors that can affect application performance [16]. In this study, the testing stage was carried out using the black-box testing method.

4. RESULTS AND DISCUSSION

4.1 Data Collection Methods

The results of the planning phase, which included system requirements analysis conducted through observation and interviews with staff and visitors to the Raihan Islamic Swimming Pool in Tangerang, revealed that staff experienced challenges such as being overwhelmed by the large number of manual bookings, which were prone to recording errors, duplicate bookings, and the difficulty of monitoring schedules and finances remotely. Meanwhile, visitors struggled to make reservations because they had to come in person and lacked real-time schedule information. Based on these challenges, a website-based swimming pool booking and monitoring application was developed. This system will assist both visitors and pool managers in improving the quality and effectiveness of the booking and monitoring processes.

Table 2. PIECES Analysis

No.	Aspects	Existing System	Proposed System
1.	Performance	Manual performance can cause errors and delays in the recording and booking process.	Manual performance can cause errors and delays in the recording and booking process.
2.	Information	Process simplification and automation through web-based applications improve overall performance.	Process simplification and automation through web-based applications improve overall performance.
3.	Economics	Pool information, schedules, and transactions are prone to human error in manual recordkeeping.	Pool information, schedules, and transactions are prone to human error in manual recordkeeping.
4.	Control	The implementation of web services allows for better data integration, as well as providing accurate and up-to-date information.	The implementation of web services allows for better data integration, as well as providing accurate and up-to-date information.
5.	Efficiency	Manual processes tend to be more expensive and time-consuming,	Manual processes tend to be more expensive and time-consuming,

		because they still use paper and other stationery.	because they still use paper and other stationery.
6.	Service	Time and cost savings through automation of booking and payment processes	Time and cost savings through automation of booking and payment processes

User Stories

This system is designed to meet the needs of two primary user groups: admins and customers. This is based on the results of user needs identification. For admins, the system needs to provide a standalone online booking function that simplifies the booking process and reduces the need for direct communication. The system should also have automatic reservation recording to prevent human error. Admins also need features to manage customer data in an organized manner, as well as a scheduling calendar that displays pool availability in real time.

For customers, the system should provide them with the ability to directly access the schedule, make their own reservations, and receive automatic confirmation after booking. To increase convenience and confidence in the service provided, additional features are needed, such as the option to choose from a variety of services, the ability to make online payments, and clear information about packages and pricing.

4.2 Design

The result of the design phase is a proposed business process translated into Unified Modeling Language (UML). This study used three types of UML diagrams: Use Case Diagrams, Activity Diagrams, and Class Diagrams. This design will simplify and accelerate overall system development. Furthermore, this design is expected to minimize errors during system development, as the development process can reference the design documents produced during this phase. The following are some of the results of the system's UML design:

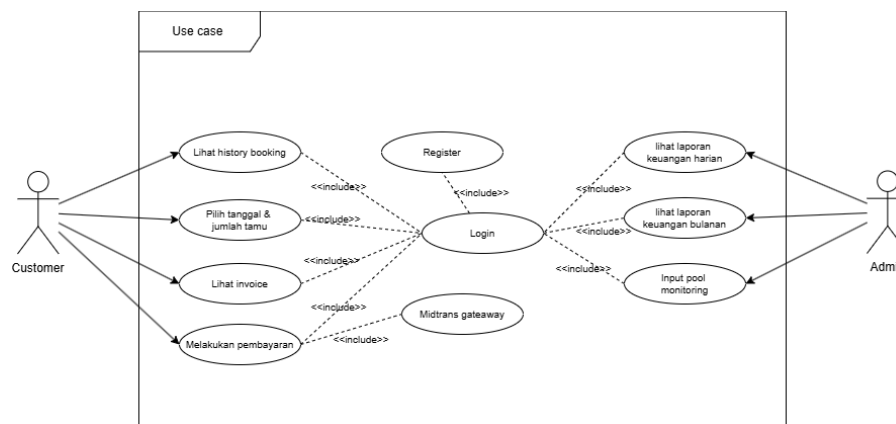


Figure 2. Proposed Use Case Diagram

As shown in Figure 2, the customer and admin are the two main actors in this interaction. While customers don't need to log in to view the booking page, contact page, and dashboard, they must register if they don't already have an account to reserve a pool. Customers can make a pool reservation and proceed to the payment page to complete the transaction after successfully logging in. It's important to note that the admin's dashboard is different from the customer's, and the admin must log in before viewing the transaction.

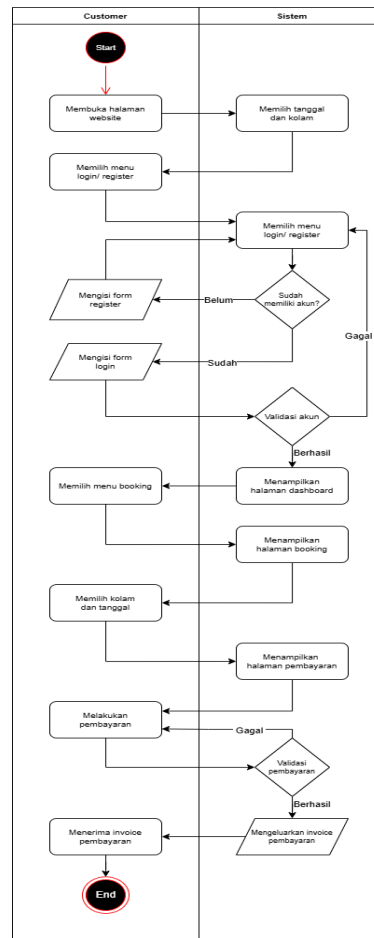


Figure 3. Proposed Activity Diagram

From figure 3 customers can book swimming pools directly using the recommended business system through a website integrated with the REST API. The first step in this process is to log in to an existing account or register if you don't already have one. Once successfully logged in, customers can select a pool and date, then use the Midtrans API interface to complete the payment online. Upon successful payment, the system will immediately generate an invoice as proof of booking and payment.

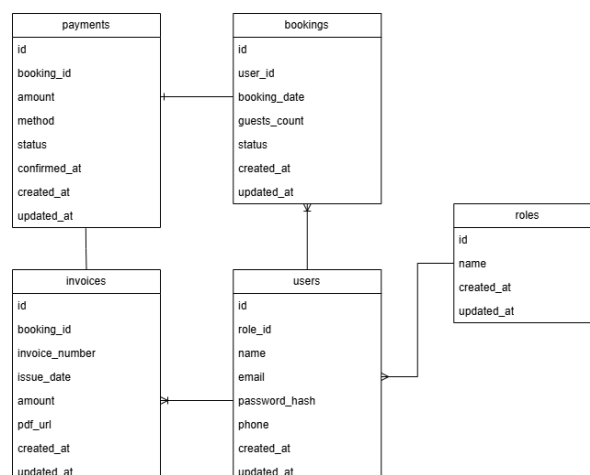


Figure 4. Proposed Class Diagram

Figure 4 shows a proposed class diagram depicting the structure and relationships between classes in the swimming pool booking and monitoring management system. This diagram consists of several main classes, such as Users, Roles, Bookings, Payments, and Invoices. Overall, this diagram provides a comprehensive overview of how data and functions relate to account holders and the activities they can perform within the system

4.3 Coding

The coding implementation process involves three stages: the frontend (the website's appearance), the backend (the behind-the-scenes system that processes databases and systems), and the database (a collection of data or information stored systematically). The booking and monitoring system on the Raihan Swimming Pool website was built using the Next.js framework, a modern React framework for developing full-stack web applications. The user interface (frontend) uses TypeScript with JSX syntax and Tailwind CSS styling support. Meanwhile, the backend is developed serverless using Next.js's built-in API routes, also written in TypeScript, facilitating frontend and backend integration without the need for a separate backend framework like Express.js. MySQL is used for database storage. This development adopts the Progressive Web App (PWA) concept, enabling the application to operate across multiple platforms and devices, such as computers, tablets, and smartphones. It also utilizes a REST API for Midtrans payments, news retrieval, and customer login with Google accounts.

4.3.1 Iteration Plan

The development process for the booking and monitoring system on the Raihan Swimming Pool website was implemented using the Extreme Programming (XP) approach. This development was divided into several iterations, with each iteration prioritizing the gradual development of key features.

The first iteration focused on the system's core design, including database development, user authentication (login and registration), and user interface development. The second iteration focused on creating critical functions, including the pool booking process and managing user and administrator data. The third iteration aimed to develop the payment function and integrate the payment system through Midtrans. The final iteration was the system refinement phase, which included black-box testing and the addition of supporting functions. The XP iterative methodology allows for gradual system development, providing the ability to adapt to changing needs and producing a system that better suits user needs.

4.3.2 Implementing Google Account Login

This system utilizes the OAuth 2.0 method to access Google accounts. An automatically generated URL is used to redirect users to the Google consent page. After the user successfully logs in and grants permission, Google redirects them to the backend to process the token and retrieve user information. This eliminates the need for manual account creation and facilitates more secure authentication.

4.3.3 Dynamic Data Fetching Implementation

This system automatically presents the latest sports news, including swimming and aquathlon, through a connection to the GNews API. The NewsPage component's fetch method automatically fetches data, ensuring that information is always up-to-date without the need for manual updates. This feature makes the application more informative and relevant to current sports issues

4.3.4 Payment Implementation Using Midtrans

Transaction processing is carried out via the REST API by sending a POST request from the frontend to a backend endpoint connected to the Midtrans Snap API using the official SDK. The backend will respond with a transaction token, which the frontend uses to display the payment

pop-up. This flow reflects the full REST API integration between the frontend, backend, and Midtrans' payment services.

4.3.5 User Interface Implementation Results

The following is a display of the user interface (UI) that has been customized to meet the needs.

1. Login Page

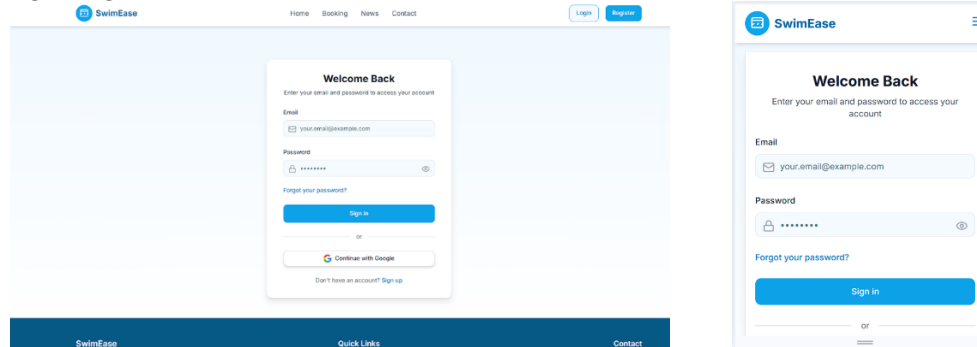


Figure 5. Website Login Page View in Desktop and Mobile Mode

Displays a login page containing a form with email and password fields, and also allows login using a Google account

2. News Portal Page

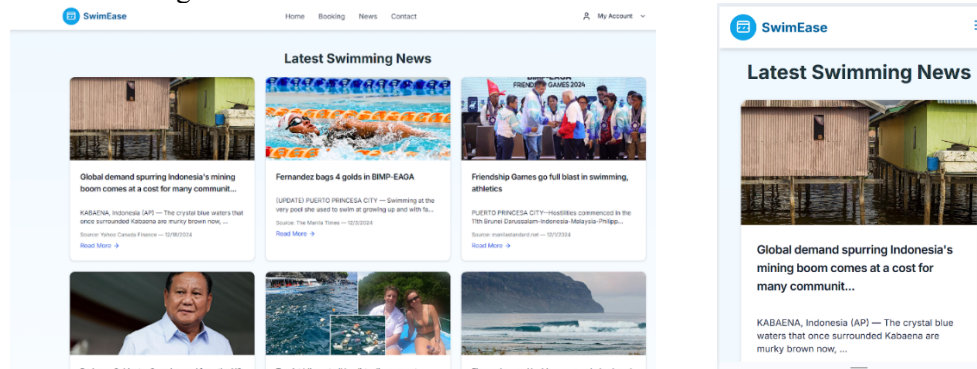


Figure 6. Website News Portal Page View in Desktop and Mobile Mode

Displays a page about news related to the world of water sports, or can contain the latest news about upcoming activities held by the swimming pool.

3. Payment Page

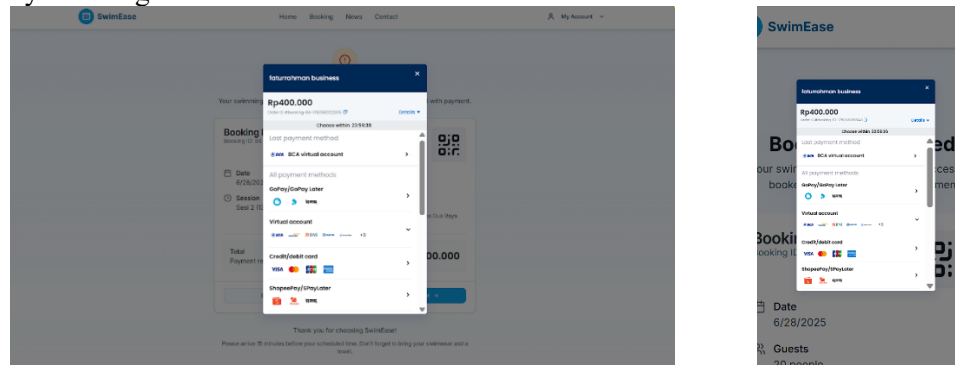


Figure 7. Website Booking Page View in Desktop and Mobile Mode

Displays the payment page using a third-party API, Midtrans, to complete payments using various payment methods

4.4 Testing

This testing phase utilizes black-box testing. This testing phase of the swimming pool booking and monitoring application system is conducted to ensure the system functions as required and is free of bugs or errors that could impact application performance. The following table shows the results of the black-box testing conducted.

Table 3. Black-Box Testing

No.	Activity	Test Scenario	Expected Results	Test Results
Customer				
1.	Login	1. Fill in customer data 2. Press the login button	Successfully logged into the dashboard page, and there were no problems logging in.	Valid
2.	Register	1. Fill in customer data 2. Press the register button	Successfully entered the login page, and there were no problems when registering.	Valid
3.	Booking	1. Add the booking date and session to be included in the checkout. 2. Press the "continue to payment" button.	There was no problem selecting the booking date and session, pressing the "continue to payment" button, and successfully going to the "confirmation" page.	Valid
4.	Payment	1. Press the "Continue to payment" button 2. Integration with MidTrans 3. Make payment on the MidTrans page	Displays virtual account and price details, no problems occur when making payments with Midtrans API integration and successfully displays payment invoices.	Valid
5.	Invoice	1. Click the "view invoice" button on the "my bookings" page. 2. Click the "view invoice" button.	Displays virtual account and price details, no problems occur when making payments with Midtrans API integration and successfully displays payment invoices.	Valid
6.	Logout	1. Press the logout button	The account was successfully logged out of the system, and there were no problems when logging out.	Valid
Admin				
1.	Login	1. Fill in the admin data 2. Press the login button	Successfully logged into the dashboard page, and there were no problems logging in.	Valid
2.	Viewing transaction data (Monitoring)	1. View transaction data on the dashboard	Displays transaction data and there are no problems when viewing transaction data.	Valid
3.	View the booking list	1. Select the booking list menu	Displays complete booking list data, and there are no problems	Valid

			when displaying booking list data.	
4.	Editing a booking on the booking list	1. Select the booking list menu 2. Press the "edit" button 3. Edit the booking list details 4. Press the "save changes" button 5. Press the "close" button	Successfully changed the customer booking detail data	Valid

5. CONCLUSION

The Raihan Swimming Pool booking and monitoring system has successfully overcome manual booking challenges such as queues, recording errors, and limited access to information by leveraging web services and REST APIs. This solution allows users to view information directly, make reservations and payments independently, and utilizes Next.js, Midtrans, Google login, and PWA technologies. Furthermore, this solution provides pool managers with access to an automated financial dashboard, facilitating monitoring and decision-making. Overall, this system has improved operational efficiency and service quality.

While the current booking and monitoring system has improved service convenience and efficiency, it still has several shortcomings. First, booking cancellations and refunds must be handled manually by administrators because an automated system is not yet available. Second, integration with third parties can be problematic because REST API documentation is not yet publicly accessible in a standard format like Swagger. Third, although the system uses Google authentication, it lacks advanced security features like JSON Web Tokens (JWT) to protect API endpoints. Fourth, there are no data visualization tools that can assist managers in making strategic decisions, such as interactive graphs depicting usage data and financial reports. Furthermore, objective performance evaluation is still needed in future research, as system performance measurements, particularly those related to Server-Side Rendering (SSR) and Client-Side Rendering (CSR), have not been quantitatively measured using metrics like First Contentful Paint (FCP) or Largest Contentful Paint (LCP).

Based on the research results, the system can be improved to enhance service quality by adding real-time notification features and the ability to automatically cancel and refund. To support better strategic decision-making, an analytical dashboard for data visualization is also needed. From a technical perspective, the use of OpenAPI (Swagger) is necessary to align REST API documentation for better integration and improve security through the implementation of technologies such as JSON Web Token (JWT). Finally, to obtain more objective results, a quantitative assessment of system performance related to the implementation of SSR and CSR is needed using metrics such as FCP and LCP.

REFERENCES

- [1] N. P. Alalwan, A. A., Dwivedi, Y. K., & Rana, "gital transformation and smart systems: A systematic literature review and future research agenda," *Technol. Forecast. Soc. Change*, 2021, [Online]. Available: <https://doi.org/10.1016/j.techfore.2021.120692>
- [2] G. Azizi, S. M., Khatony, A., & Moradi, "The effectiveness of hospital online booking systems in improving patient satisfaction: A quasi-experimental study," *BMC Med. Inform. Decis. Mak.*, 2020, [Online]. Available: <https://doi.org/10.1186/s12911-020-01165-6>
- [3] M. Wasesa, F. I. Ramadhan, A. Nita, P. F. Belgiawan, and L. Mayangsari, "Impact of overbooking reservation mechanism on container terminal's operational performance and greenhouse gas emissions," *Asian J. Shipp. Logist.*, vol. 37, no. 2, pp. 140–148, 2021, doi:

- 10.1016/j.ajsl.2021.01.002.
- [4] J. Talath, A. Guru Vaishnavi, C. Rojashree, and B. Kavya, "Web Based Sports Arena Booking Hub," *Int. J. Multidiscip. Res.*, vol. 7, no. 2, pp. 1–21, 2025, doi: 10.36948/ijfmr.2025.v07i02.42591.
 - [5] S. Pati and Y. Zaki, *Evaluating the Efficacy of Next.js: A Comparative Analysis with React.js on Performance, SEO, and Global Network Equity*, vol. 1, no. 1. Association for Computing Machinery, 2025. doi: 10.1145/3701716.3715565.
 - [6] S. Srivastava, H. Shukla, N. Landge, A. Srivastava, and D. Jindal, "A Comprehensive Review of Next.js Technology: Advancements, Features, and Applications," *SSRN Electron. J.*, 2024, doi: 10.2139/ssrn.4831070.
 - [7] Y. Muharam, M. D. Sugiri, T. Informatika, U. B. Bandung, T. Informatika, and U. B. Bandung, "IMPLEMENTASI ALGORITMA FIRST COME FIRST SERVED PADA APLIKASI PEMESANAN LAPANGAN FUTSAL BERBASIS WEBSITE PADA BISOC FUTSAL," vol. 08, pp. 41–45, 2021.
 - [8] T. I. J. Kusumawati, "Pengembangan Sistem Booking Online Pada Cokro Barbershop Berbasis Website," *JoMMiT J. Multi Media dan IT*, vol. 7, no. 1, pp. 026–030, 2023, doi: 10.46961/jommit.v7i1.737.
 - [9] A. Suprianto, "Pengembangan Sistem Informasi Pemesanan Barang Berbasis Web," pp. 131–139, 2024.
 - [10] M. Sanjaya and P. R. N. Saputra, "Pemanfaatan Nextjs Dan Mongoddb Dalam Sistem Informasi Web Manajemen Data Beras Pada Ud Sri Utami," *Inf. Syst. Educ. Prof. J. Inf. Syst.*, vol. 8, no. 1, p. 25, 2023, doi: 10.51211/isbi.v8i1.2414.
 - [11] R. Choirudin and A. Adil, "Implementasi Rest Api Web Service dalam Membangun Aplikasi Multiplatform untuk Usaha Jasa," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 18, no. 2, pp. 284–293, 2019, doi: 10.30812/matrik.v18i2.407.
 - [12] H. Susilo, N. Abdillah, M. Ikhsan, and H. Diana Morika, "Analisa Dan Perancangan Sistem Informasi Booking Antrian Pelayan Pada Klinik Medika Saintika Berbasis Website," *J. Kesehat. Med. Saintika*, vol. 14, no. Nomor 1, pp. 344–352, 2023.
 - [13] B. Dwi and P. Informatika, "Pengembangan Sistem Informasi Pemesanan Catering Berbasis Website Dengan Metode Extreme Programming," *Teknologipintar.org*, vol. 3, no. 9, pp. 2023–2024, 2023.
 - [14] D. W. T. Putra and R. Andriani, "Unified Modelling Language (UML) dalam Perancangan Sistem Informasi Permohonan Pembayaran Restitusi SPPD," *J. TeknolIf*, vol. 7, no. 1, p. 32, 2019, doi: 10.21063/jtif.2019.v7.1.32-39.
 - [15] B. O. Lubis, "Penerapan Global Extreme Programming Pada Sistem Informasi Workshop, Seminar Dan Pelatihan Di Lembaga Edukasi," *Informatika*, vol. 3, no. September, pp. 234–245, 2016.
 - [16] I. D. Maulana, Y. A. Susetyo, P. Studi, T. Informatika, F. T. Informasi, and K. S. Wacana, "Implementasi Fetch API dalam pengembangan Backend Website Daftar Film dengan Next . JS Abstrak," vol. 6, no. 1, pp. 187–196, 2025.