Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

June 2026. Vol. 5. No. 3; e-ISSN: 2808-4519

Development of a Web-Based Booking Portal System for a Hospital Information System

Moh. Zahri Valent Affandi[1*], Fauzan Ilyas Almeyda[2], Moch. Fikri Nazaruddin[3], Faisal Mutaqin[4]

[1,2,3,4]*UPN "Veteran" Jatim*

23081010159@student.upnjatim.ac.id[1*], 23081010188@student.upnjatim.ac.id[2], 23081010217@student.upnjatim.ac.id[3], faisalmuttaqin.if@upnjatim.ac.id[4]

Abstract

The advancement of information technology has accelerated digital transformation in the healthcare sector through the development of web-based healthcare services that are more accessible to the public. One implementation of this transformation is a Patient Portal that provides various digital healthcare services, including online appointment booking. However, complex navigation and poorly structured information presentation can reduce user convenience when accessing these services. This study aims to develop a web-based Patient Portal user interface to improve accessibility to digital healthcare services. The research employed the Prototyping method, which includes requirement analysis, interface design, system implementation, data integration, functional testing, and evaluation. The system was developed using React JS, Remix JS, TypeScript, and Mantine UI, and integrated with an Application Programming Interface (API) to provide dynamic data presentation. The development focused on the appointment booking feature, which consists of branch selection, clinic selection, doctor selection, payment method selection, schedule selection, and booking confirmation. The results indicate that the developed interface provides more structured information, simplifies user navigation, and delivers a responsive user experience across various devices. Black Box Testing results showed that all system functions operated according to user requirements. Therefore, the development of the web-based Patient Portal interface can improve access to digital healthcare services and support a more effective and efficient healthcare appointment booking process.

Keywords: *Appointment Booking; Digital Healthcare Services; Patient Portal; User Interface; Web Application*

1. Introduction

The development of information and communication technology has driven digital transformation across various sectors, including the healthcare sector. This transformation aims to improve service quality, administrative efficiency, and accessibility of healthcare services for the public. One form of digital transformation implementation in healthcare is the development of web-based healthcare service systems that enable users to access various services online. Through such systems, patients can obtain health information, register for services, select examination schedules, and manage their healthcare needs without having to visit healthcare facilities directly. The utilization of web-based technology is considered capable of improving service efficiency and expanding public access to digital healthcare services [1].
A booking portal is one implementation of digital healthcare services within a Hospital Information System (HIS) designed to connect patients with healthcare facilities through a web-based platform. This system provides various features that support healthcare service processes, such as patient account registration, healthcare facility information search, doctor appointment booking, laboratory examination services, vaccination services, and Medical Check-Up (MCU) services. Although these features have been able to support user needs, the success of a system is determined not only by the completeness of its functionality but also by the quality of its user interface. An unintuitive interface, complex navigation, and poorly structured information presentation can cause users to experience difficulties in completing their intended tasks. Such conditions may result in reduced user comfort and decreased effectiveness in utilizing digital healthcare services [2].
Based on these conditions, it is necessary to develop a system that not only focuses on the User Interface (UI) but also on the server-side and overall application architecture to provide a better, safer, and more efficient user experience. On the user interface side, development is carried out through the implementation of a simple, responsive, and easy-to-understand design. UI development focuses not only on the aesthetic aspects of the interface but also on ease of navigation, design consistency, information accessibility, and efficiency of system workflows. The implementation of good UI/UX principles has been proven to improve usability and user satisfaction in utilizing digital services [3].
In addition, development is also carried out on the server-side to improve system security, scalability, and maintainability. The implementation of authentication mechanisms using JSON Web Tokens (JWT) is applied to ensure secure user access to the available services. From the software architecture perspective, the system is developed using a modular approach through the separation of components based on their responsibilities, such as the use of a service layer to manage business logic, type definitions to maintain data

structure consistency, and the separation between presentation, service, and data management layers. This approach aims to improve code readability, simplify development and maintenance processes, and support future feature development. The web-based patient booking system is developed through user interface improvements, server-side optimization, and the implementation of a more structured and secure system architecture to enhance the quality of digital healthcare services. With these developments, it is expected that users will be able to access digital healthcare services more easily, quickly, securely, and comfortably, while the system achieves better maintainability, scalability, and security to support the continuous improvement of technology-based healthcare services.

2. Research Methods

This study uses the Prototyping method in the process of developing a web-based Patient Booking Portal user interface. The Prototyping method was selected because it allows the development process to be carried out gradually through the creation of an initial design that can be evaluated and adjusted according to user needs. This approach helps developers gain a clearer understanding of system requirements and minimizes design errors during the implementation stage [4]. The focus of this study is the development of a user interface that can improve access to digital healthcare services through more structured information presentation, easy-to-understand navigation, and a responsive interface across various devices.

The first stage of the study is requirements analysis, which is the process of identifying user needs and system requirements to be developed. At this stage, information is collected regarding the healthcare service workflows available within the Patient Booking Portal, such as the login process, healthcare facility selection, doctor appointment booking, laboratory examinations, vaccination services, and Medical Check-Up (MCU) services. The analysis is conducted through observation of the existing system, discussions with the development team, and literature studies related to user interface development and digital healthcare services. The results of this stage are used as the basis for determining the features and interface requirements to be developed.

The next stage is user interface design, which is carried out by developing the navigation structure, system workflow, and interface design for each page. The design process considers User Interface (UI) and User Experience (UX) principles, such as design consistency, ease of use, information readability, and interface responsiveness [5]. After the design process is completed, the implementation stage is carried out using React JS, Remix JS, TypeScript, and Mantine UI technologies on the user interface side, along with the implementation of a structured server-side architecture through service layers, type definitions, and API integration. The system is also equipped with a JSON Web Token (JWT)-based authentication mechanism to support secure user access to the available services. Furthermore, testing is conducted using the Black Box Testing method to ensure that each feature functions according to user requirements. Black Box Testing is used because it is capable of evaluating system functionality based on inputs and outputs without considering the structure of the program code used [6]. The testing results are used as evaluation material to determine the success level of the user interface development in supporting easier access to digital healthcare services.

The research stages carried out in the development of the web-based Patient Booking Portal user interface consist of:

Table 1: Research Phases

Number	Appearance (in Time New Roman)	
	Research Phases	**Description**
1	Literature Review	Collecting references on user interface, user experience, web applications, and digital health services
2	Needs Analysis	Identifying user needs and required features for the Patient Booking Portal
3	User Interface Design	Designing the page structure, navigation, and interface components for the Patient Booking Portal
4	User Interface Development	Implementing the design using React JS, Remix JS, TypeScript, and Mantine UI
5	Data Integration	Connecting the interface to the API to display data dynamically
6	Functional Testing	Testing each feature and the system's user flow
7	Evaluation	Analyze the results of the implementation and make improvements if any issues are identified

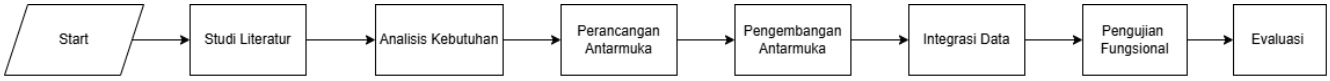

Fig. 1: Research Process

In addition to user interface requirements, the analysis stage also included the identification of system requirements on the server-side and application architecture. This stage aims to ensure that the developed system not only provides an easy-to-use interface but also delivers secure, structured, and maintainable services. The analysis covers user authentication requirements, patient data management, integration with Application Programming Interfaces (APIs), communication mechanisms between the user interface and the server, as well as user session management throughout the system usage process. The results of the analysis are used to determine a system architecture design that supports the separation of responsibilities among application components, enabling development and maintenance processes to be carried out more effectively.

Based on these requirements, server-side development was carried out by implementing a modular architecture using React JS, Remix JS, and TypeScript. The system was designed by separating business logic into a service layer (service files), defining data structures through type definitions, and separating user interface components from data processing activities. The use of TypeScript and type definitions aims to maintain the consistency of data structures received from APIs, minimize data processing errors, and improve system reliability throughout the development process. In addition, user authentication and authorization mechanisms were implemented using JSON Web

Tokens (JWT) to enhance secure access to the Patient Booking Portal services. JWT was selected as the authentication mechanism because it facilitates authentication processes without directly storing user sessions on the server, thereby improving system scalability and reducing access latency [7].
To support data management efficiency and provide a better user experience, the system utilizes a session management mechanism integrated with local data storage using Zustand Persist. This approach enables important user data, such as session information and specific payloads, to remain available as long as users maintain access rights without requiring repeated data reloading. In the data retrieval process, Remix JS utilizes the loader feature to fetch and provide data from the server before pages are rendered to users. This approach helps improve application performance, reduce page loading time, and ensure that the displayed data always reflects the latest condition on the server. Overall, the implementation of a modular architecture, JWT authentication, session management, type-safe programming, and state management using Zustand Persist supports the creation of a system that is more secure, structured, easier to develop, and possesses better maintainability and scalability.

3. Result and Discussion

3.1. Results of the Needs Analysis

The requirements analysis stage was conducted to identify user needs regarding the digital healthcare services available on the web-based Patient Booking Portal. The analysis was carried out through observations of the existing service workflows and user requirements in booking healthcare services online. Based on the analysis results, it was found that users require a system capable of providing healthcare service information clearly, easily accessible, and supported by a simple workflow so that it can be used by various groups of users. Ease of access to information is one of the important factors influencing user acceptance of digital healthcare services [8].

The following illustrates the booking portal workflow that has been designed to provide an efficient User Experience (UX):

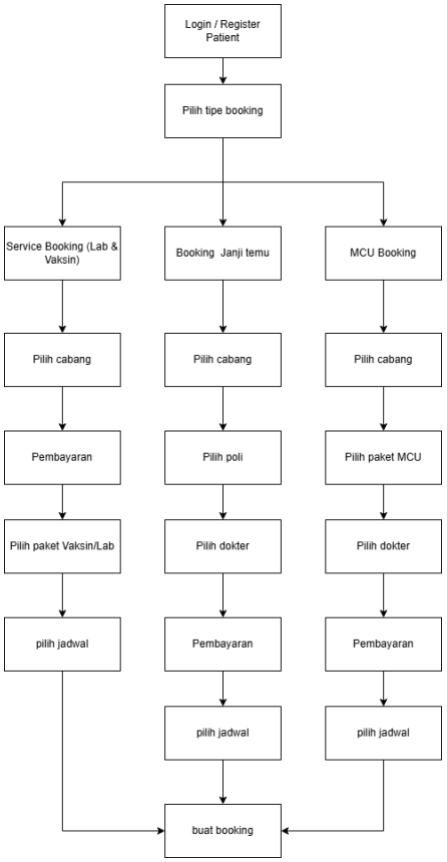

Fig. 2: Design of the Patient Booking Portal Workflow in the Hospital Information System

In addition to easy access to information, users also require a booking process that is structured and easy to understand. In the developed system, the patient booking process is designed so that users can complete all booking stages sequentially, starting from selecting a healthcare facility branch to the booking confirmation process. With a clear workflow, users can understand each step that must be completed without experiencing confusion while using the system. A simple navigation structure and a good user experience have been proven to improve the usability of web-based systems [9].
Another requirement is the system's ability to display data dynamically according to the availability of healthcare services. Information such as healthcare facility branches, examination package selection, service clinics, doctors, payment methods, and service schedules must always be updated so that users receive accurate information. Therefore, the interface is designed to connect with an Application Programming Interface (API), enabling data to be displayed in real time according to the information available in the backend system. The integration of digital technology in healthcare services plays an important role in improving information quality and the effectiveness of healthcare service delivery [10].

3.2. User Interface Design Results

Based on the identified requirements, a user interface design process was carried out by considering User Interface (UI) and User Experience (UX) principles. The design was conducted with the objective of creating an interface that is easy to use, provides clear navigation, and delivers a comfortable user experience.
The interface structure was designed using a user-centered approach that prioritizes simplicity and design consistency. Each page has a uniform component layout, enabling users to recognize system functions and workflows more easily. In addition, the use of card-based components helps present information in a more structured and understandable manner. The workflow that patients must follow requires them to complete the Login/Register process for system authentication purposes before accessing all pages of the booking portal.

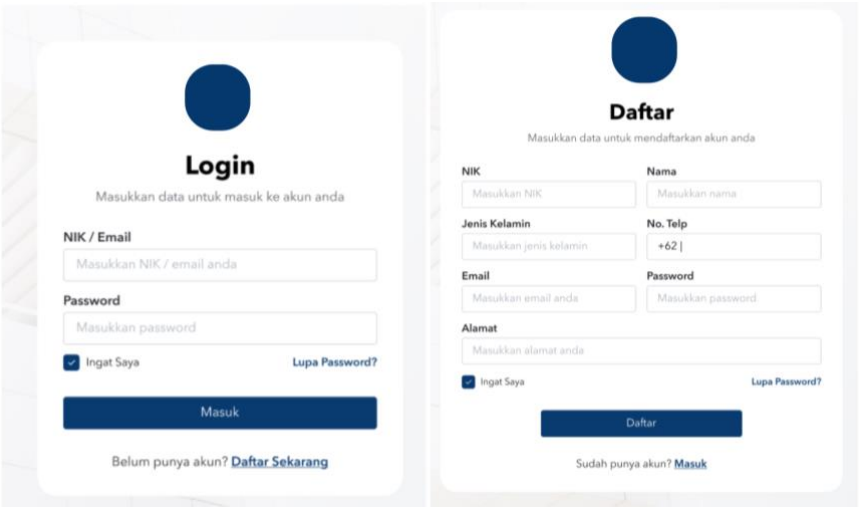

Fig. 3: Appointment Booking Interface Design

Before making a booking, patients will be directed to the main dashboard, which serves as a navigation point to the next booking page of their choice. The dashboard is designed with a layout that makes it easy for patients to understand the information on the website.

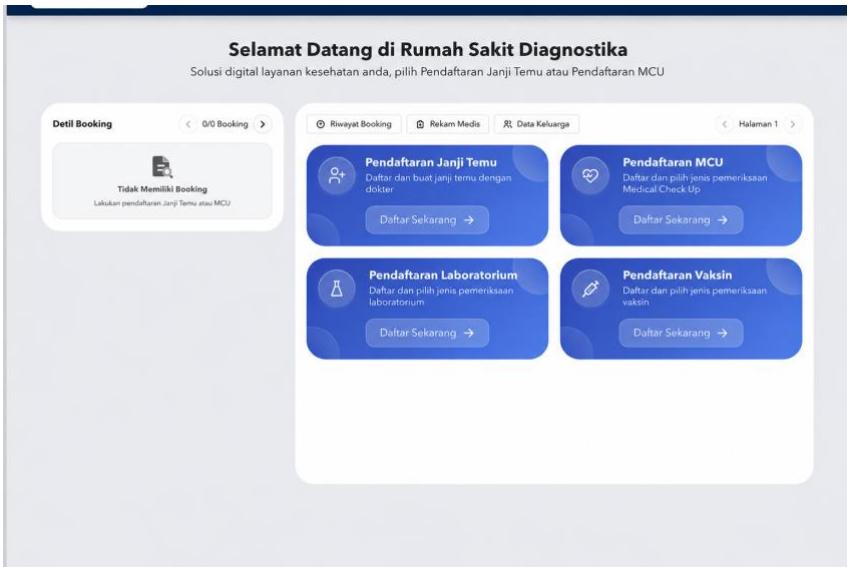

Fig. 4: Appointment Booking Interface Design

The process for booking appointments, services (lab tests/vaccinations), and medical checkups is designed in six main steps: selecting a healthcare facility branch, selecting a clinic, selecting a doctor, selecting a payment method, selecting a service schedule, and confirming the booking. For services (Lab/Vaccines), there is no selection of a doctor or clinic; instead, users select a service package (lab/vaccine package). Each step is designed sequentially to help users complete the booking process more easily.

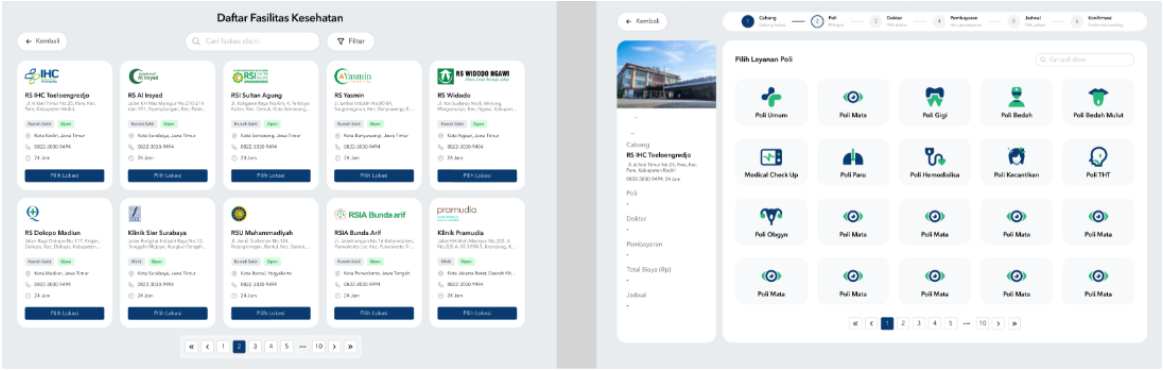

Fig. 5: Appointment Booking Interface Design

3.3. Implementation of the Patient Booking Portal Interface

Based on the development results, the Patient Booking Portal interface was successfully implemented using React JS, Remix JS, TypeScript, and Mantine UI. The development focused on simplifying the healthcare service booking process through booking features. The interface was designed by emphasizing ease of navigation, visual consistency, and responsiveness across various device sizes. All pages were developed using a reusable component approach, which facilitates both system development and maintenance processes.

The implemented patient booking workflow consists of several stages, namely healthcare facility branch selection, clinic selection, examination package selection, doctor selection, payment method selection, consultation schedule selection, and booking confirmation. Each stage is designed sequentially so that users can understand the booking process more easily and minimize errors when making healthcare service bookings.

3.3.1. Branch Selection Page

The branch selection page is the first step in the booking process. On this page, the system displays a list of available healthcare facilities along with supporting information that can help users choose their preferred service location.

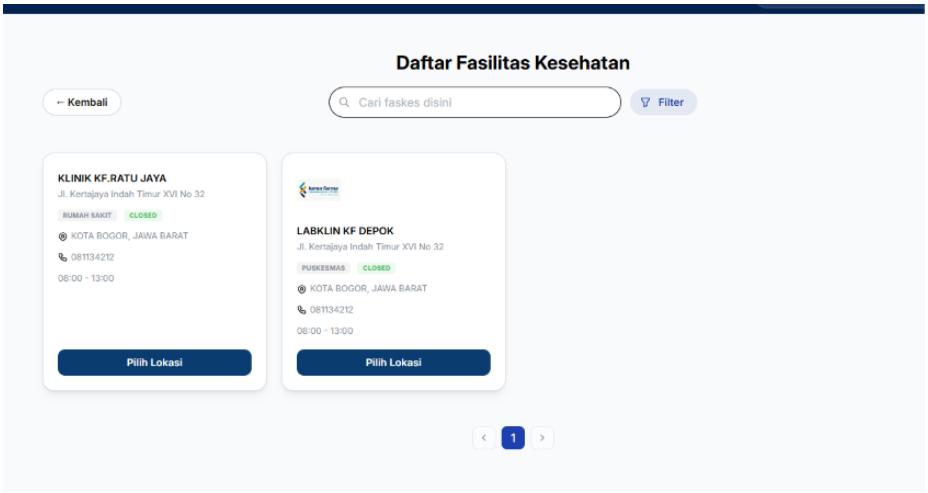

Fig. 6: Branch Selection Page

Based on the implementation, the use of a card-based interface makes it easier for users to view branch information in a more structured way. In addition, the information displayed can be dynamically updated through API integration, ensuring that users receive data that reflects the system's current status.

3.3.2. Clinic Selection Page

After selecting a branch, users are directed to the clinic selection page (except for lab service or vaccine appointments). This page displays the various clinic services available at the previously selected healthcare facility.

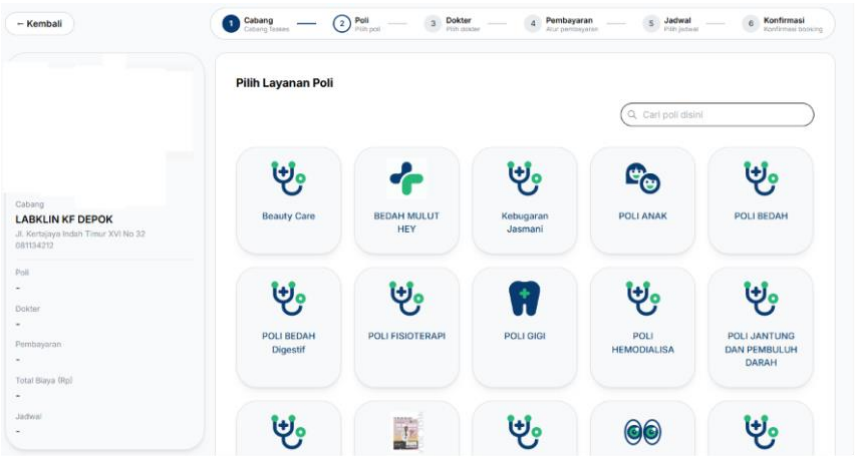

Fig. 7: Clinic Selection Page

The interface is designed to display a simple list of services so that users can select the services they need without having to navigate through many additional screens. The concise presentation of information helps speed up the process of selecting healthcare services.

3.3.3. Service Plan Selection Page

The service selection page is intended solely for vaccine or lab appointments and is used to display a list of available service packages based on the selected appointment type—either a vaccine appointment or a lab appointment—or the user's selection. This page will display a list of available packages, including the package names and available stock.

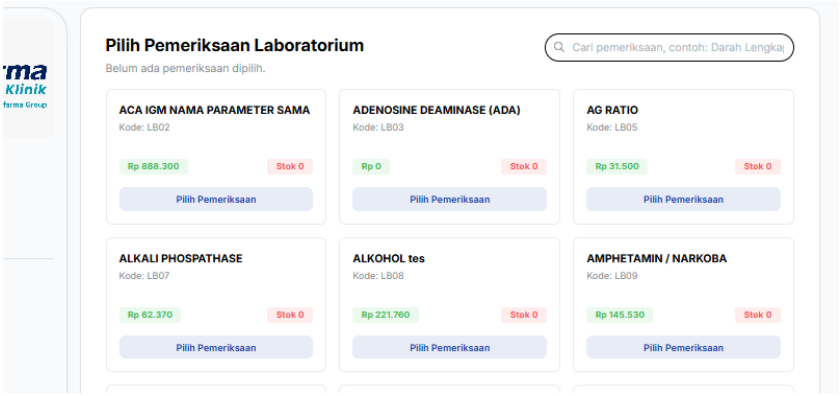

Fig. 8: Laboratory Test Package Selection Page

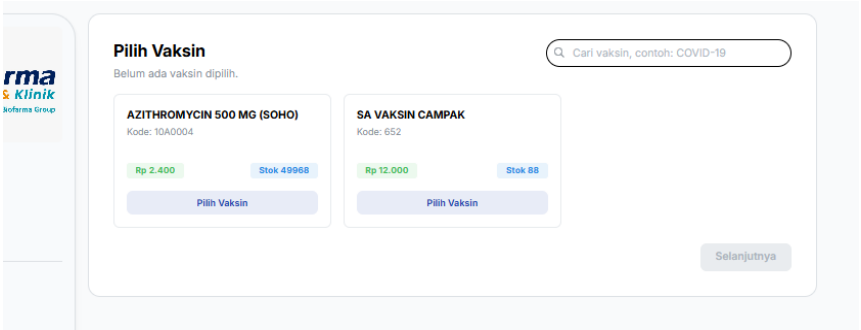

Fig. 9: Vaccine Screening Package Selection Page

The service selection page is designed to allow multiple selections from the displayed items. The use of a grid layout enhances the user experience, while the use of cards and reusable components improves the website's performance. Additionally, a search function is provided to make it easier to find items when there are a large number of them.

3.3.4. MCU (Medical Check-up) Package Selection Page

The Medical Checkup (MCU) package selection page functions the same way as the service selection page—it is used to select a checkup package. The page uses a card-based layout, displaying comprehensive details about the package's features, including the price and a button to select the checkup directly.

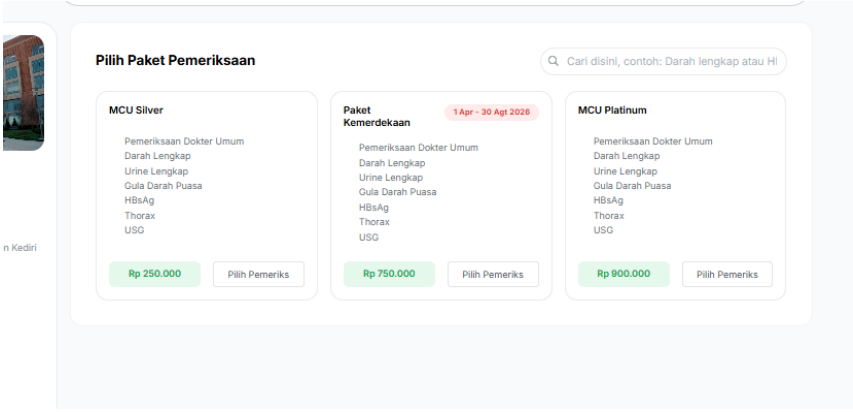

Fig. 10: MCU Package Selection Page

3.3.5. Doctor Selection Page

The doctor selection page (excluding lab and vaccine service bookings) is used to display a list of available doctors based on the clinic selected by the user. The information displayed includes the doctor's name and specialization.

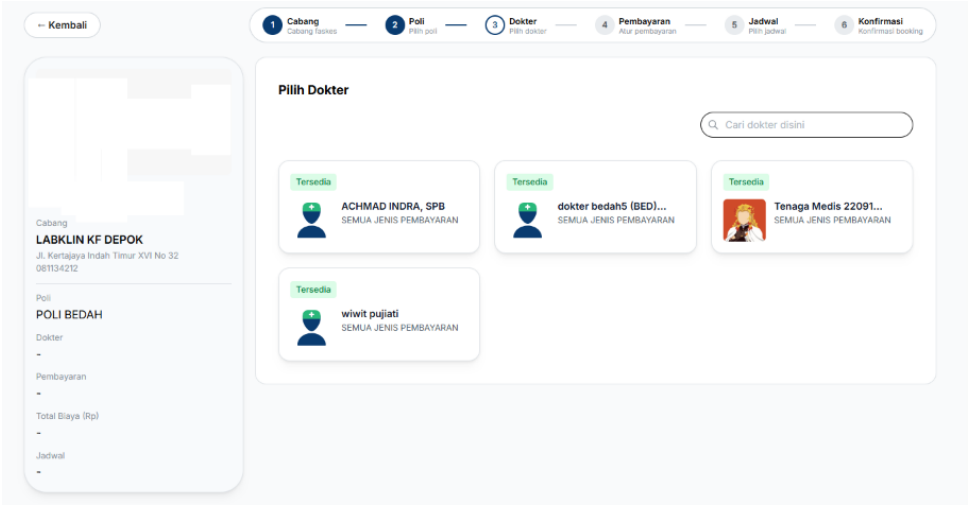

Fig. 11: Doctor Selection Page

Based on the results of the implementation, the use of card-style components in the list of doctors has improved the readability of the information and made it easier for users to compare the available doctor options.

3.3.6. Payment Page

The payment page is used to select the payment method to be used during the healthcare service booking process.

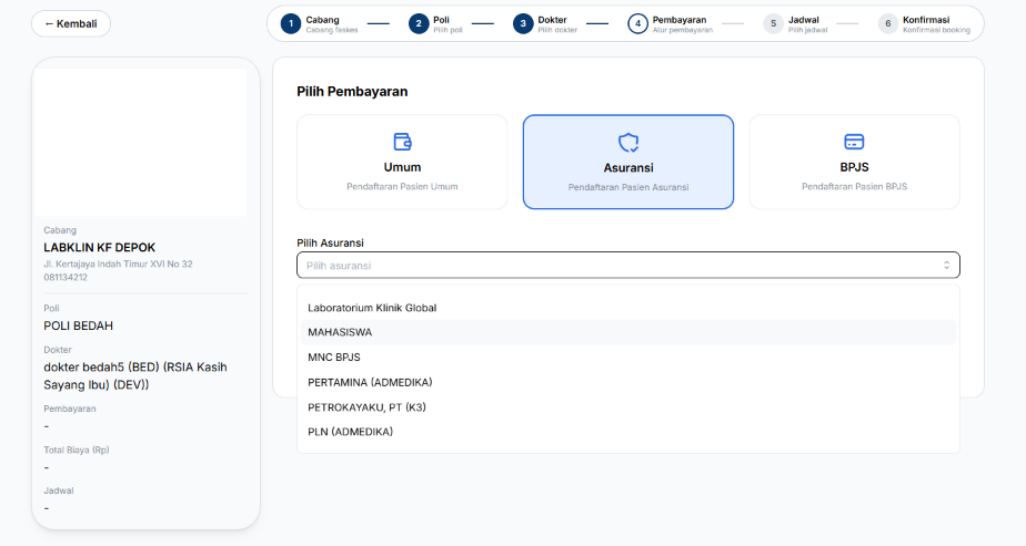

Fig. 12: Payment page

The interface is designed with a focus on simplicity so that users can complete the payment process without having difficulty understanding the available options.

3.3.7. Schedule Selection Page

The schedule page is used to determine the available dates and times for appointments based on the doctor selected earlier.

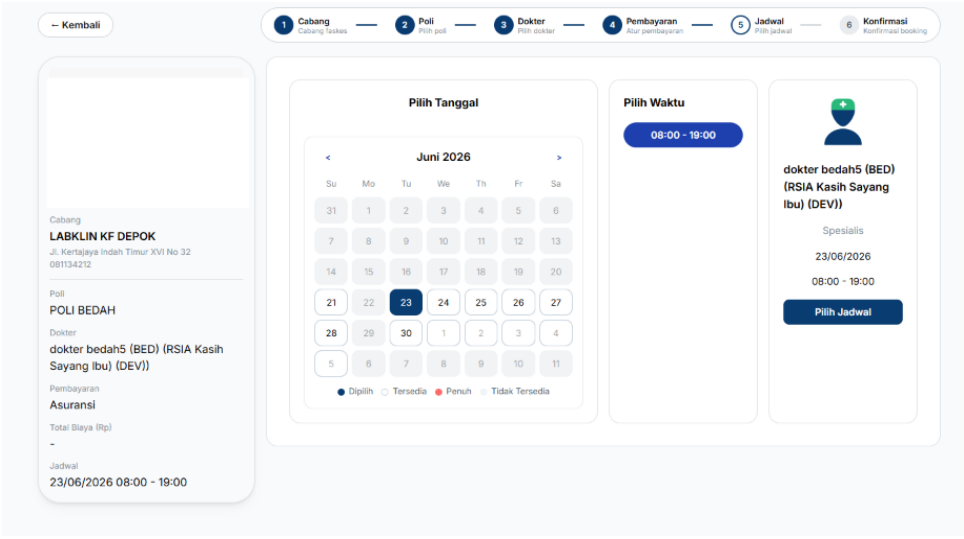

Fig. 13: Schedule selection page

Integration with the API allows the system to display the schedule dynamically, so users can only select appointment times that are still available. This helps reduce the potential for scheduling conflicts.

3.3.8. Booking Confirmation Page

The confirmation page is the final step in the appointment booking process. On this page, the system displays a summary of the information selected by the user before the booking process is completed.

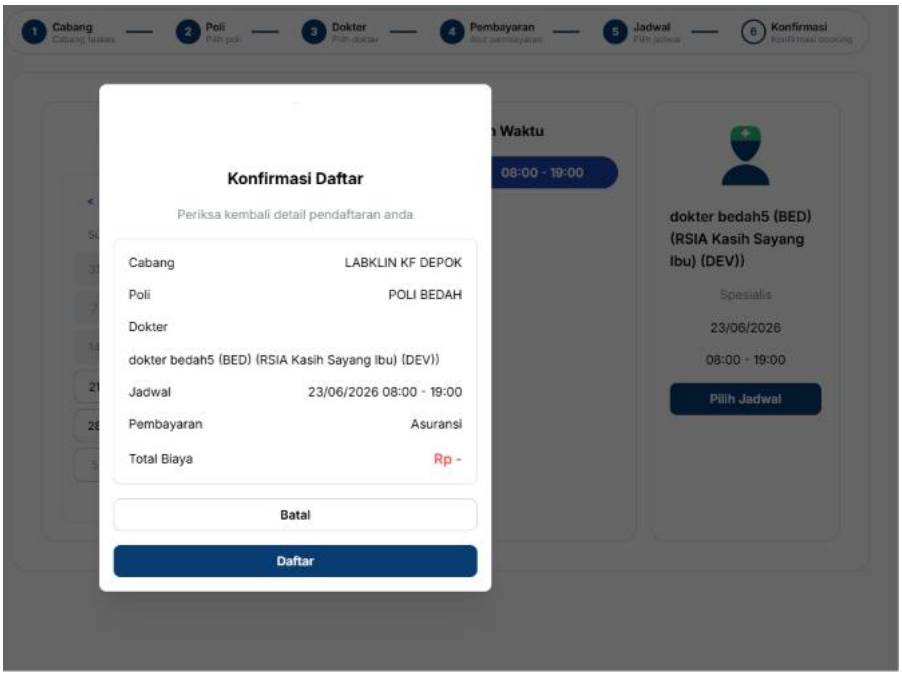

Fig. 13: booking confirmation page

Through the confirmation page, users can double-check the booking details they have selected to minimize data entry errors before the information is saved to the system.

3.4. System Functional Testing

System testing was conducted using the Black Box Testing method to ensure that every function within the patient booking feature operates in accordance with user requirements. Testing was performed on all stages of the booking process, from branch selection through to confirmation.

Number	Feature Tested	Description	
1	Branch Selection	Selecting a healthcare facility branch	Successful
2	Clinic Selection	Selecting a service clinic	Successful
3	Package Selection (Laboratory & Vaccination)	Selecting an examination package (multiple-choice option)	Successful
4	MCU Package Selection	Selecting a Medical Check-Up examination package	Successful
5	Doctor Selection	Selecting an available doctor	Successful
6	Payment	Selecting a payment method	Successful
7	Schedule Section	Selecting a consultation schedule	Successful
8	Booking Confirmation	Saving booking data	Successful
9	Interface Responsiveness	Accessing the system on a mobile device	Successful

Table 2: System Testing

Based on the testing results that have been conducted, all tested features successfully functioned according to the system requirements. The data selected by users could be carried through each stage of the booking process without any data loss, while the API integration was able to display information dynamically according to the selected services. In addition, the developed interface was capable of adapting its display to various screen sizes, ensuring a comfortable user experience on both desktop and mobile devices.

4. Conclusion

Based on the results of the study, the development of the web-based Patient Booking Portal User Interface (UI) was successfully implemented to support easier access to digital healthcare services. The development was carried out using React JS, Remix JS, TypeScript, and Mantine UI, focusing on the appointment booking feature, which consists of branch selection, clinic selection, doctor selection,

payment, schedule selection, and booking confirmation stages. The developed interface is capable of presenting information in a more structured manner, simplifying user navigation, and providing a more intuitive and responsive user experience across various devices.
The results of functional testing using the Black Box Testing method indicate that all tested features functioned according to the system requirements. Each stage of the booking process could be completed successfully, from service selection to storing booking data within the system. In addition, integration with the Application Programming Interface (API) enables healthcare service information to be displayed dynamically and in accordance with the actual system conditions. Therefore, the development of the web-based Patient Booking Portal user interface can help improve access to digital healthcare services while providing a more effective and efficient user experience for users.

References

[1] Carini, E., Villani, G., Pezzullo, A., Gentili, A., Barbara, A., Di Mario, M., Ricciardi, W., & Boccia, S. (2021). The Impact of Digital Patient Portals on Health Outcomes, Health Care Quality, and Patient Attitudes: Updated Systematic Literature Review. Journal of Medical Internet Research, 23(9), e26189.
[2] El Yaman, N., Bachour, H., Hajj, A., Salameh, P., Zeenny, R. M., & Sacre, H. (2023). Utilization of Patient Portals and Associated Factors among Adults: A Cross-Sectional Study. BMC Medical Informatics and Decision Making, 23(1), 189.
[3] Oudbier, S. J., van der Vaart, R., Kelders, S. M., & Ossebaard, H. C. (2025). Patients' Experienced Usability and Satisfaction with Digital Health Solutions in a Home Setting. JMIR Medical Informatics, 13, e63703.
[4] Alshamrani, A., & Bahattab, A. (2015). A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model. International Journal of Computer Science Issues, 12(1), 106–111.
[5] El Saddik, A., & Orozco, M. (2022). User Experience Design Principles for Interactive Digital Systems. Journal of Interaction Design and Architecture, 53, 45–58.
[6] Wibowo, A., Nugroho, Y., & Prasetyo, R. (2023). Implementation of Black Box Testing in Web-Based Information Systems. Journal of Information Systems Engineering and Business Intelligence, 9(2), 120–129.
[7] Cahyadi, R., & Akbi, D. R. (2026). Implementasi Restfull API Menggunakan Json Web Token Pada Website Manajemen Infrastruktur. Jurnal Ilmiah Penelitian dan Pembelajaran Informatika, 11(2), 996–1009.
[8] Grossman, L. V., Masterson Creber, R., Ryan, B., Vawdrey, D. K., & Ancker, J. S. (2022). Patient Portal Use and Experience Among Patients with Chronic Conditions: A Systematic Review. Journal of Medical Internet Research, 24(6), e31570.
[9] Alharbi, A., Alotaibi, R., Alghamdi, S., & Alzahrani, A. (2024). Factors Influencing User Experience and Usability of Digital Health Applications: A Systematic Review. International Journal of Medical Informatics, 184, 105367.
[10] Haleem, A., Javaid, M., Singh, R. P., & Suman, R. (2023). Digital Healthcare Applications and Their Impact on Modern Healthcare Systems. Sensors, 23(2), 788.