ISSN: 2581-1274

Development of Online NAS Application User Guide Website using NextJS 13 and REST API to Enhance User Understanding

Hajjul Ikram [a,1,*], Muhammad Ardiansyah [b,2]

[a,b] *University of Teuku Umar, Meureubo, Meulaboh 23681, Indonesia*
[1] *Hajjulikram@gmail.com;* [2] *m.ardiansyah@utu.ac.id**
* Corresponding author

ARTICLE INFO

Article history:
Accepted

Keywords:
Nusantara Certification Application System
User Guide
NextJS 13
REST API
Application Efficiency

ABSTRACT

This research focuses on developing a user guide for the Nusantara Certification Application System (NAS) through an online platform, utilizing NextJS 13 technology and REST API to enhance comprehensive understanding and optimal utilization of NAS features. The implementation of NextJS 13 ensures a responsive and adaptive interface, while the REST API optimizes speed and smooth access to required data. Through a combination of structured video tutorials and information sources in PDF format, the objective of this guide is to provide comprehensive insight into each feature offered within the NAS application. Research findings confirm the success of integrating NextJS 13 technology and REST API, significantly enhancing understanding and efficiency in NAS application use. The primary role of this implementation lies in aiding users to deeply comprehend and optimize the benefits of the NAS application while expediting access to necessary information. In conclusion, the utilization of cutting-edge technology in user guide development has proven to have a positive impact by broadening user insights into the application and enhancing productivity and efficiency in its use. This underscores the significance of employing advanced technology in delivering information to modern application users.

I. Introduction

The Nusantara Certification Application System (NAS) has emerged as a fundamental pillar in the journey toward ease and efficiency in handling professional certifications. Developed by PT Nusantara Aplikasi Sertimedia, NAS serves as a system aiding Professional Certification Institutions (LSP) and Competency Test Centers (TUK) in operating more efficiently.

In an increasingly technology-dominated world, a profound understanding of the NAS application is crucial to harness its full potential. To achieve this and strengthen user comprehension, this journal is proudly presented. The primary focus of this journal is the "Development of Online NAS Application User Guide," a critical endeavor to guide users in operating this system with greater skill and understanding.

To accomplish this goal, the author has employed a robust approach by leveraging the innovative Next.js 13 technology and integrating REST API services. Through this guide, users will be equipped with the knowledge needed to master the NAS application, enhance efficiency, and ensure optimal data security.

This journal aims to provide a comprehensive overview of the development of this user guide, emphasizing the Waterfall development method chosen by the author. This approach allows meticulous planning, in-depth analysis, thorough design, robust development, rigorous testing, and efficient implementation.

In the subsequent chapters, the author will guide you through the steps of developing this website, with the hope that the user guide developed will significantly contribute to supporting NAS application users and assist them in fully utilizing this system.

II. Literature Review

A. Typescript

TypeScript, as a modern development language supporting static compilation, empowers developers to write more structured, clear, and understandable JavaScript code. It serves as a platform implementable across various contexts, such as NodeJS and browsers supporting ECMAScript or more advanced versions. With optional features like static data types, classes, and interfaces, TypeScript offers developers the opportunity to enhance the quality of produced software through a more systematic and cohesive approach to coding. [1]

B. ReactJS

ReactJS, as a library providing a robust framework for reusable user interface components, not only enables the development of complex web applications without the need for page refresh but also plays an integral role as the View part in MVC architecture. With this approach, react provides clear structure in application development. Through the utilization of the Document Object Model (DOM) and support for server-side rendering using NodeJS, as well as mobile app development using React Native, react offers high effectiveness in modern application development. [2]

C. NextJS 13

Next.js is a JavaScript framework that extends the functionalities of React. Its primary focus lies in developing responsive and robust web applications by offering a range of features that enhance user experience and facilitate efficient web app development. This framework stands out for its ability to handle Server-side Rendering (SSR) and Static Site Generation (SSG), enabling applications to be rendered on the server-side before being sent to the browser and generating static sites at compile time to boost performance and speed in user experience.

D. Maintaining the Integrity of the Specifications

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

E. Single Page Application

A Single Page Application (SPA) is a web application model consisting of just one page that doesn't require a reload when users interact. In SPAs, JavaScript is responsible for updating content without the need to reload the entire page. Belluano (2018) explains that the primary aim of using SPAs is to lessen server load by reducing data requests and minimizing the required server resources. [3]

F. REST

The Representational State Transfer (REST) architecture in web services enables resource access via specific HTTP URLs. In this context, clients have the ability to request access using the HTTP protocol solely with URLs, without additional steps. Each URL refers to a set of programs that can be executed and provide responses to the request sender. REST utilizes HTTP request methods like GET, POST, PUT, and DELETE to execute commands received by the server. [4]

G. API

An API, or Application Programming Interface, is a guide that outlines various elements such as interfaces, classes, functions, and structures necessary in software development. Serving as a bridge between applications or websites created by developers and various available services, APIs enable access to and utilization of functionalities from other systems, both within the same environment and across platforms. Beyond mere documentation, APIs grant access to features needed in application development, facilitating seamless interaction among different software components,

and allowing integration of diverse functionalities without needing to understand the implementation details behind the scenes. In this role, APIs foster opportunities for collaboration among technological platforms, enhance development efficiency, and provide room for scalability and flexibility improvements in applications. [5][6]

III. Method

The writer has opted for the Waterfall development method as the primary framework, providing a solid foundation for designing, developing, and refining this application.

Fig. 1. Waterfall SDCL

A. Planning

In the planning phase, the writer meticulously maps out the project. They define the project's objectives in detail, aiming to optimize user understanding of the NAS application. In this segment, the investigator delineates the issue and delimits the research's extent to ascertain the procedural stages involved in resolving the focal problem. This encompasses the identification of essential resources, financial allocations, and technical expertise necessary for the research endeavor [6]. Additionally, the writer deeply identifies project needs and requirements, ensuring efficient integration with NextJS 13 and REST API. A tight schedule is established to guide the execution of this project.

B. Analysis

The analysis phase focuses on understanding user needs and how the NAS application is utilized in real-life scenarios. The writer conducts in-depth research into the challenges users face and how the user guide can address these issues. The findings from this analysis aid the writer in designing a more robust user guide architecture, enabling users to properly navigate the NAS application.

C. Design

In the design phase, the writer elaborates on the user guide design in comprehensive detail. Attention is given to the user interface design, crafting user-friendly interfaces, and planning the integration with REST API. Additionally, the writer plans the structure and format of video tutorials and PDFs to ensure users have effective tools for understanding the NAS application.

D. Development

The development phase marks the point where the writer begins translating all plans and designs into reality. NextJS 13 is employed to implement the user guide's web components. Careful integration with REST API is conducted to ensure smooth data access. Simultaneously, during this phase, the creation of video tutorials and PDF materials commences in accordance with the design plan.

E. Testing & Implementation

Testing is a crucial step in the Waterfall phase. The writer conducts comprehensive testing to ensure that all aspects of the user guide function properly. They ensure seamless integration with REST API and check for any technical issues in video tutorials and PDF materials. Through quality and functionality testing, the writer ensures that the guide is ready for user training.

The implementation phase is when the user guide is deployed to the production environment, ensuring its availability to NAS application users without any hindrance

F. Maintenance

Maintenance is an ongoing phase post-implementation. The writer will continuously monitor the performance of the user guide, respond to user feedback, and update the guide as needed. Maintenance aims to keep the guide relevant, effective, and aligned with technological advancements.

IV. Results and Discussion

This research has produced an online user guide for the NAS application supported by NextJS 13 technology and REST API. The guide aims to effectively enhance user understanding of the NAS application, providing quick and easy access to video tutorials and detailed PDF materials explaining the intricacies of NAS application usage. Robust integration with NextJS 13 enables the creation of a responsive and user-friendly interface, while REST API is utilized to access the necessary data for this guide. The primary goal of developing this guide is to provide effective tools for NAS application users to better understand and efficiently utilize the application.

Upon completing the planning, development, and system testing phases, the outcomes and implementation of the Online NAS Application User Guide Website utilizing NextJS 13 and REST API to enhance user understanding are as follows:

A. The Main Page of The Website.

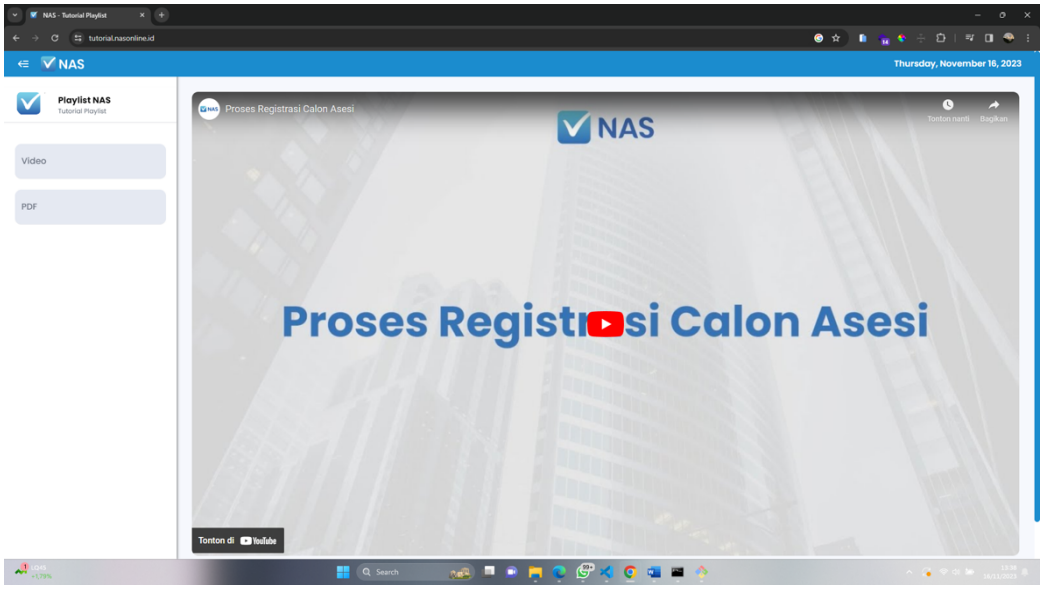

Fig. 2. The main page

On the designed main web page, there's a well-structured layout. The top section displays a divided header. On the left side, there's the NAS logo, reflecting the application's identity. Meanwhile, on the right side, there's information about the current date and day, providing time context to users.

Additionally, there's a well-organized sidebar featuring an accordion that allows users to choose between two main options: a menu for videos or a menu for PDFs. This accordion aids in organizing and displaying menu choices systematically, enabling users to easily select relevant options according to their needs.

The main section of the web page is dedicated to displaying the content chosen by the user. If the user selects the video option, the main page will display the main playable video. Conversely, if the user chooses the PDF option, the main page will display an accessible PDF document. Below the main content area, there's a title explaining the displayed content, enabling users to quickly understand the topic or contents of the selected video or PDF. The overall page design ensures an intuitive and efficient interface for users to navigate and access relevant content based on their needs.

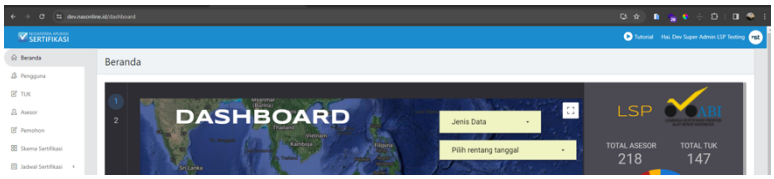

Fig. 3. Icon tutorial

The NAS application provides access to this user guide website. In the website header, there's an icon labeled "Tutorial" that, when clicked, redirects users to the user guide web page.

B. Sidebar Menu

Sidebar menu on this main page will serve as the primary navigation menu to guide users while on the website. Within the sidebar, there are two accordion buttons: Video and PDF. Clicking these buttons will reveal sub-accordion sections.

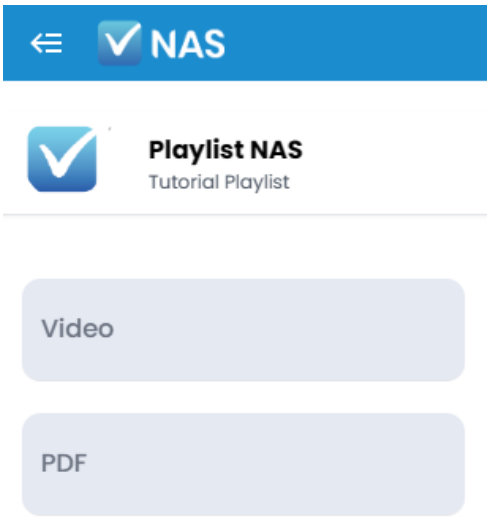

Fig. 4. Sidebar menu

C. Video Accordion

The video accordion button also functions as navigation to select various other menu options. Within the video accordion, there are role-based menu options such as Assessor, Applicant, and LSP Admin. Clicking any of these role menus will expand the accordion, displaying a list of videos with a scrollbar for easy navigation and improved layout aesthetics.

Fig. 5. Video accordion

D. PDF Accordion

The PDF accordion contains documents in PDF format related to roles such as Applicant, Assessor, and LSP Admin. Upon selecting a specific role, the accordion will display a list of corresponding PDF documents. Equipped with a scrollbar feature, users can easily navigate and access the necessary documents based on their roles within the system.

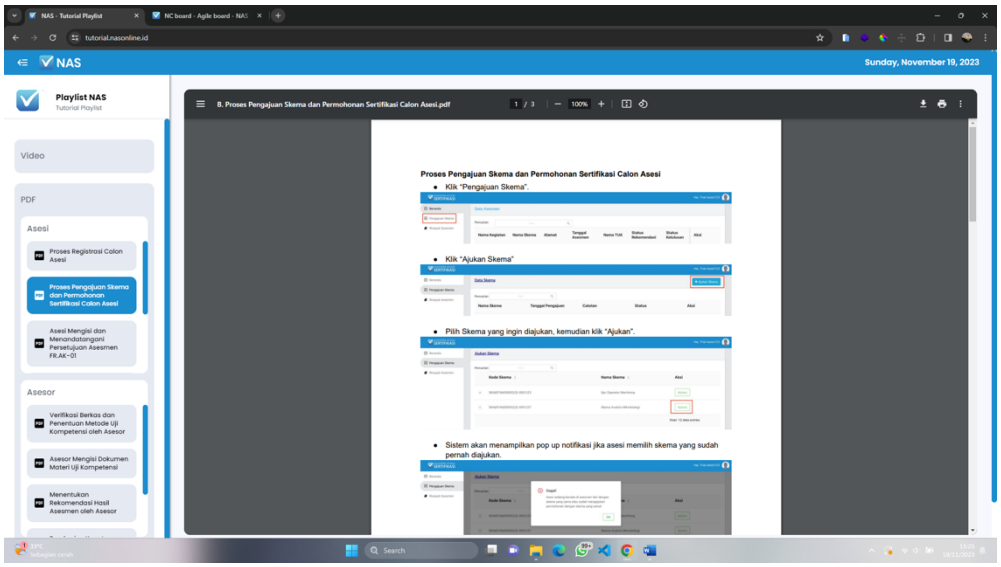

Fig. 6. PDF accordion

V. Conclusion

The development of the NAS application user guide through NextJS 13 technology and REST API has significantly enhanced user understanding and efficiency in application usage. The integration of these modern technologies has enabled the creation of a responsive user interface, expedited data access through REST API, and provided in-depth tutorials. With NextJS 13 technology, user experience is ensured to be satisfying, while REST API facilitates fast and reliable access to necessary information. As a result, the development of this guide has brought significant benefits by offering better comprehension to NAS application users and improving efficiency in its utilization.

References

[1] Microsoft. TypeScript - Documentation. (n.d.). Retrieved November 23, 2023, from https://www.typescriptlang.org/.

[2] Aggarwal, S. (2018). Modern Web-Development using ReactJS. *International Journal of Recent Research Aspects*, 5, 133–137.

[3] Jadhav, M. A., Sawant, B. R., & Deshmuhk, A. (2015). Single Page Application using AngularJS. *International Journal of Computer Science and Information Technologies*, 6(3), 2876–2879.

[4] Perkasa, M. I., & Setiawan, E. B. (2018). Pembangunan Web Service Data Masyarakat Menggunakan REST API dengan Access Token.

[5] Amri, M. S. (2011). Membangun sistem navigasi di Surabaya menggunakan google maps API. *Jurnal Teknik Informatika, Institut Teknologi Sepuluh November Surabaya.*

[6] Sukiman, T. S. A., & Ihsan, M. A. (2023). Information System for South Aceh Historical Stories. *Jurnal Inovasi Teknologi dan Rekayasa, 8*(2), 219–224.