International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

Modern Front End Web Architectures with React.Js and Next.Js

Mochammad Fariz Syah Lazuardy[1], Dyah Anggraini[2]
[1, 2]Department of Computer Science, Gunadarma University, Depok, West Java, Indonesia-Pincode-16424
Email Address: [1] mochammadfariz11(at)gmail.com, [2] d_anggraini(at)staff.gunadarma.ac.id

Abstract**—***Before technological advances in the field of web development today, web developers usually handle applications from 2 sides at once, namely from the front-end and back-end sides which are commonly referred to as full-stack developers. However, nowadays, the terms back-end and front-end are two separate elements, so that front-end web development can only focus on things related to the front-end, such as display and animation on the user interface without having to think about back-end side. One technology that is often used on the web front-end is the React.js library. However, the React.js library basically supports the client-side-rendering method, so when you want to use the React concept with the server-side-rendering method, the Next.js framework is needed. The Next.js framework is a React framework that has a myriad of features, so it is used as a technology in building front-end apps of information system of state civil apparatus (SIASN). This study aims to discuss the advantages and disadvantages of the React.js library and Next.js framework as one of the technologies used in the development of SIASN web applications. The result of this research is to provide an overview of the steps taken by the developer when building the SIASN application from the front-end side. In addition, this study provides points of advantages and disadvantages of React.js and Next.js as the main technology in building user interface of SIASN application.*

Keywords**—** *SSR, CSR, Data Fetching, React.js, Next.js.*

I. INTRODUCTION

Nowadays web applications can do a lot of work besides just displaying static data. Modern web applications have become an important part of the daily ecosystem due to the demands of clients' needs. Therefore, development on the front-end side of the web plays an important role in website creation because it becomes a means of user interface with the system. A good web front-end can attract users with all the convenience, feasibility, and features available.

One of the factors that influence the optimization of front-end web technology is choosing the right library and front-end framework. Generally, Java Script libraries and frameworks play an important role in terms of scalability, maintenance, modularity and making web pages more interactive. The advantages of the React.js library and the Next.js framework were then applied to the development of the SIASN (state civil apparatus information system) at the State Civil Service Agency.

The State Civil Service Agency also knowns as Badan Kepegawaian Negara (BKN) is an Indonesian non-ministerial government agency tasked with carrying out government duties in the field of state personnel management [1]. The State Civil Service Agency has problems in the quality of ASN (State Civil Apparatus) data caused by the unavailability of access for ASN to make repairs and update data independently [2].

The development of the SIASN web application is a challenge considering that this application will be used by all ASN in Indonesia. Front-end web as a user interface with the SIASN system must be optimized from the aspect of user interface, user experience and application performance.

II. LITERATURE REVIEW

A. Information System

An information system is an integrated environment of hardware, software, and people in charge of processing data into useful information. Information systems have many important functions within an organization so almost every organization uses information system services to solve various existing problems. The information system serves to provide information to all levels of management, namely high, middle to lower-level management. In essence, the information system has 3 main activities, namely receiving data as input, then processing data such as calculations, updating, and others and finally obtaining information from data that was previously processed as output.

Information systems, in general, can be classified by dimensions, infrastructure, and level of use[3].

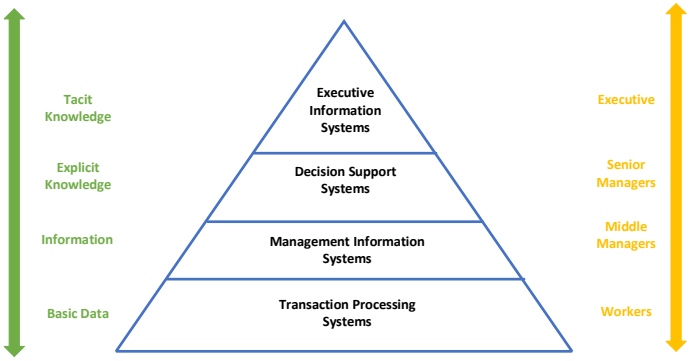

Figure 1. Types and levels of information systems (Faalih, 2018)

B. React.js

Title React or aslo known as React.js or ReactJS is a Java Script-based front-end library for building user interfaces or UI components. React is managed by Facebook, a community of developers and companies [4].

In a study conducted by Venkat in 2021, it was explained that web development technology before 2015 was about scripting and rendering. At that time HTML and CSS were used for the front-end, while the back-end used PHP.

Mochammad Fariz Syah Lazuardy and Dyah Anggraini, "Modern Front End Web Architectures with React.Js and Next.Js," *International Research Journal of Advanced Engineering and Science*, Volume 7, Issue 1, pp. 132-141, 2022.

Developer puts static HTML pages in some folders and renders those files using PHP. Developers use this method for decades, so there is no visible revolution on a website until there is a Java Script library like React.js.

React.js has several excellent features that make front-end web development faster and more responsive.

C. DOM

DOM (Document Object Model) is a representation of page or HTML document resources into objects. DOM can be manipulated so that web pages become more interactive and attractive.

Most web development frameworks have DOM interactions that are performed directly by bringing the entire DOM tree control over each web page when activated. Furthermore, when a large collection of information has to be adapted, the presentation of the web page is affected. Nonetheless, React.js uses something known as a virtual DOM.

Correlation with virtual DOM and native DOM is made using the diff() algorithm and only nodes that change along the line will reflect into the DOM tree [5]

D. React Component

In front-end web development using the React.js library, React uses the component paradigm in building views and displaying data. Components are small UI (User Interface) elements that can pass data to a View as props that can change over a period of time. Components are reusable so they can be used repeatedly. Components make it easy for developers to build UI by separating the UI into several parts, so that the design and process of building the UI is very effective and well structured. Components are like JavaScript functions they perform the same task but in a different environment and approach. Components take inputs known as props and return them in the form of React elements that can be seen on the user's screen [6]

On the official React js documentation, it is explained that there are two types of components, namely class components and functional components which are commonly called Hooks [7]. React supports writing and provides options for creating components using classes or functions. Then each component in React.js has a life cycle which is commonly known as a component lifecycle. There are 4 outlines of the class component lifecycle, namely:

1. Mounting: These methods are called in the following order when an instance of a component is being created and inserted into the DOM. An example of a Mounting method in a component class are:
   - constructor()
   - static getDerivedStateFromProps()
   - render()
   - componentDidMount()
2. Updating: Updates to components can be caused by changes to props or state. These methods are called in the following order when the component is re-rendered. An example of an Updating method in a component class are:
   - static getDerivedStateFromProps()
   - shouldComponentUpdate()
   - render()
   - getSnapshotBeforeUpdate()
   - componentDidUpdate()
3. Unmounting: Unmounting the component is called when the component is removed from the DOM. An example of an Unmounting method in a class component is:
   - componentWillUnmount().
4. Error Handling: This method is called when an error occurs during rendering in the component's lifecycle. An example of an Error Handling method in a class component are:
   - static getDerivedStateFromError()
   - ComponentDidCatch().

Meanwhile, when using functional components or Hooks, the lifecycle of a component is enough to use the useEffect method which can perform Mounting, Updating, Unmounting, and Error Handling processes.

E. Webservice

The development of application software using web services has grown rapidly. Many technology companies use web services in their business. A web service will be described as a method for exchanging or communicating information between devices over a network[8]. The web server has access to the database to request the data required by the client and returns the response data in JSON/XML format. According to research conducted by Halili and Ramadani explained that SOAP (Simple Object Access Protocol) and REST (Representational State Transfer) are the two protocols that are most often used to exchange messages. The REST API is a simple and easy to maintain API built on top of the REST architecture [9].

F. REST

Before the emergence of REST and the internet became popular, integration between web application services was quite difficult to implement. API (Application Programming Interface) is a media liaison between one application to another application. When a request is made by a client via a RESTful API, a resource is sent to the endpoint. The data is sent via HTTP protocol in JSON, XML, and other formats. Other information such as metadata, authorizations, URIs, cookies, and so on is sent via HTTP headers. In Singh's research in 2021, it was explained that the REST API was chosen to be the chosen standard in the world because:

1. Flexibility: Migration from one server to another can be done easily, besides that the REST API can also move to a different database at any time.
2. Scalability: Since there is a separation between client and server, upgrading individual components can be done without difficulty.
3. Easy to build and uses less resources.
4. Reliability: More resistant to failure.

Singh's 2021 research explains that there are two basic steps that need to be taken to make a REST API request. Among others, as follows:

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

1. The client accesses a specific location in the database via the REST API and the state method that must be executed. This is known as a request.
2. The server executes the method and returns the data to the client. This is called response.

In addition, there are four main components that constitute the request-response process. Which is mentioned as follows:
1. Endpoint: An endpoint is a specific location for a resource. The REST API uses a URI (Uniform Resource Identifier) that creates an address for a resource.
2. Header: Metadata or information about a request or response contains important things such as authorization, data type content, cookies and other information. Generally, headers contain information such as Content-Type, Authorization, and Cache-Control.
3. Method: The method is the type of interaction that is being performed by the resource. REST API helps us in developing web applications that allow for CRUD (Create, Read, Update, Delete).
4. Data (or Body): Contains payload data sent or received from the server. A REST Body can contain almost any media type, but the most popular is application/json.

There is a convention for writing endpoints on the REST API which is described in the table below:

TABLE I. Endpoint convention for REST API (Singh, 2021)

Endpoint	HTTP Method	Description
api/client	GET	Get all clients
api/client/new	GET	Display a form to add a new user
api/client	POST	Adds a client to the database
api/client/tom	GET	Get tom's information
api/client/tom/edit	GET	Show a form/input to edit client tom's information
api/client/tom	PUT	Update tom's information
api/client/tom	DELETE	Delete tom's data

G. Redux

Redux is a library that functions as state management when developers use the React.js library. Redux provides another product namely React Redux as the official React binding for Redux. React Redux allows React components to read data from the Redux store and dispatch actions to the central data store to update the data. Developers can easily manage the application state with the help of global access feature [10]

As React applications get more complex tracking the state during debugging becomes more difficult, but using Redux it becomes easier to track the state during debugging because Redux can send error reports to the server so developers can know if there are errors. Redux has 3 main components namely:
1. Action: Action is a type of command that represents information in the form of a payload and has a type.
2. Reducer: Reducer changes state when it receives an action via dispatch. Reducer is generally a set of functions that contains a set of logic such as switch-case and if-else that serves to determine what will be executed based on the type of action received.
3. Store: Store is a place to store the entire application state in the form of a tree. Redux can change the state contained in the store by sending a dispatch to the action.

H. Client-Side Rendering

Client-side rendering is a relatively newer type of rendering compared to server-side rendering. Client-side rendering is generally used to build web with SPA (Single Page Application) model. Client-side rendering is gaining popularity along with the development of Java Script libraries and frameworks.

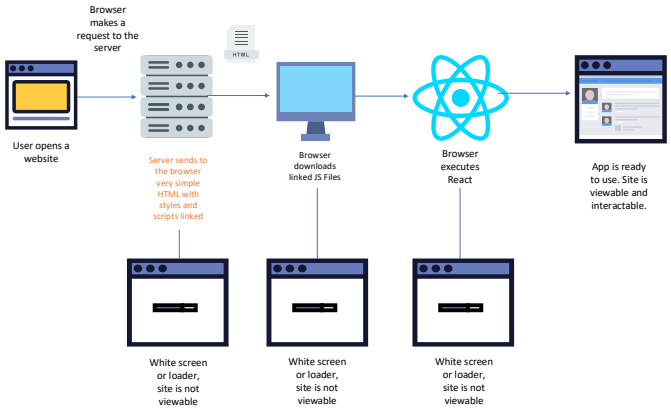

Figure 2. Client-Side Rendering (Kishore dan BM, 2020)

In Figure 2 it is explained that in this methodology correspondence with the server occurs only to obtain run-time information. Also, this method doesn't reload the entire UI after making a request to the server. The client-side structure knows how to refresh the UI by changing the data through re-rendering the UI with specific DOM components [11].

I. Server-Side Rendering

Server-side rendering is the traditional type of rendering that is done on the server-side. Server-side rendering was quite popular among web developers before Java Script libraries and frameworks developed rapidly as they are today. One of the most popular web languages for server-side rendering is PHP. In fact, before companies like Facebook used Java Script libraries like React JS, Facebook first used the PHP language.

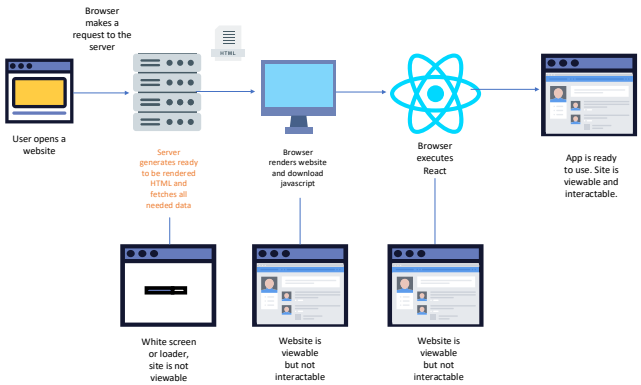

Figure 3. Server-Side Rendering (Kishore dan BM, 2020)

In a study conducted by Kishore and BM in 2020, it was explained that in server-side rendering as shown in Figure 3,

Mochammad Fariz Syah Lazuardy and Dyah Anggraini, "Modern Front End Web Architectures with React.Js and Next.Js," *International Research Journal of Advanced Engineering and Science*, Volume 7, Issue 1, pp. 132-141, 2022.

when the client starts the request-response cycle on the web page, the server will create an HTML page by obtaining explicit client information and sending it to the client's machine via the web. Programs in the browser interpret the data and render the page. All these procedures i.e. getting information from the database, creating HTML page, and sending it to the customer happens in milliseconds.

When the user clicks again on the link on the web page, the browser will make a request to the server and the whole process is carried over by the server again. This process not only increases the workload on the server but also consumes unnecessary internet bandwidth.

J. Data Fetching

On the official Next.js page it is explained that this framework has the ability to perform several data fetching mechanisms depending on application needs [12]. Some of the data fetching mechanisms in the Next.js framework are explained as follows:

1. CSR: Client-Side-Rendering uses useEffect on React Hooks to perform data fetching from the client-side, with CSR then data from the API will be fetched every time a page request occurs from the client-side (After the page is rendered, then data fetching is performed).

CSR mechanism can be done at the component and pages level. At the component level data is retrieved during component installation (mounting process), and component content is updated as data changes. If done at the pages level, the data is fetched at runtime (while the program is running), and the page content is updated as the data changes. At the component level data fetching on the client-side is useful when the web page does not require SEO indexing, does not require pre-rendered data, or when the web page content is frequently updated.

2. SSR: Server-Side-Rendering uses a special function called getServerSideProps to perform data fetching from the server-side, getServerSideProps only runs on the server-side and never runs in the browser. Unlike CSR, the SSR data fetching mechanism can only be done at the page level and cannot be applied to other files.

The data fetching mechanism with SSR is suitable when a web application needs to pre-render pages whose data must be retrieved at the time of request to the server. (Before the web page loads, the getServerSideProps function will be executed first, then a delay occurs, and after that, the data is served with the web content). The difference between SSR and CSR lies in when the API is hit. In the CSR, API it is hit after the page is loaded, while in the SSR API it is hit before the page is loaded.

3. SSG: Static-Site-Generation uses a special function called getStaticProps to perform data fetching from the server-side, getStaticProps only runs on the server-side and never runs in the browser. Just like SSR, SSG data fetching mechanism can only be done at the page level and cannot be applied to other files. The data fetching mechanism with SSG invokes the getStaticProps function one time when the page is built. The data fetching mechanism with SSG is suitable for web applications with data that does not change frequently and tends to be static. In addition, this mechanism can improve SEO because data is rendered on the server-side and application performance tends to be fast because the data fetching process is only carried out during the build process.

4. ISR: Incremental-Static-Regeneration uses an additional prop called revalidate which is added to the getStaticProps function. On the official Next.js page it is explained that Next.js allows developers to create or update static pages. ISR allows developers to use Static Generation per page, without the need to rebuild the entire application. In contrast to SSG, ISR has the advantage that at certain times and conditions, file pages can rebuild and fetch new data.

III. METHODOLOGY

A. Stages of Making Front-End SIASN

The first stage is to prepare a web front-end environment such as installing Node JS, Next JS, Postman, Adobe xd and others. The second stage is to receive a design prototype from the UI/UX team so that the front-end developer knows what the requested design looks like and can simulate navigation on the application. The third stage is slicing from the design into HTML, CSS and Java Script code that is understood by the computer. At this stage the front-end developer uses the assets provided by the UI/UX team such as icons and logos. After the third stage is creating a web interface, then the fourth stage is to receive the Postman collection in the form of a set of API URLs and API documentation from the back-end team. Then in the fifth stage, after receiving the collection API, the front-end developer can connect the web view with the data prepared by the back-end team in the form of a URL. Before connecting the API with the display, the front-end developer needs to do API testing on the Postman tools. If the API URL is working, you can proceed to the sixth stage, which is to analyze the data flow on the React component structure of the SIASN application. If it doesn't work then the front-end developer can issue an issue to the back-end team so that the API URL can be fixed. The seventh stage is connecting the data with the application display. Finally, in the eighth stage, after connecting the API URL with the application, the front-end developer can display the data that has been provided by the back-end team to the application view.

IV. ADVANTAGES OF REACT.JS

The advantages of the React.js library based on literature review and case studies of the SIASN application are explained as follows:

A. Supports writing Java Script Code with Class and Function Paradigms

Based on official React js documentation, it is explained that there are two types of components, namely class components and functional components which are commonly called Hooks. Therefore, React supports writing and provides options for creating components using classes or functions.

B. Reusable Component

Using the component concept in building an application, so that if a similar component is found in the application, it

Mochammad Fariz Syah Lazuardy and Dyah Anggraini, "Modern Front End Web Architectures with React.Js and Next.Js," *International Research Journal of Advanced Engineering and Science*, Volume 7, Issue 1, pp. 132-141, 2022.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

can be categorized as a reusable component. Using reusable components can help developers save lines of code and processing time.

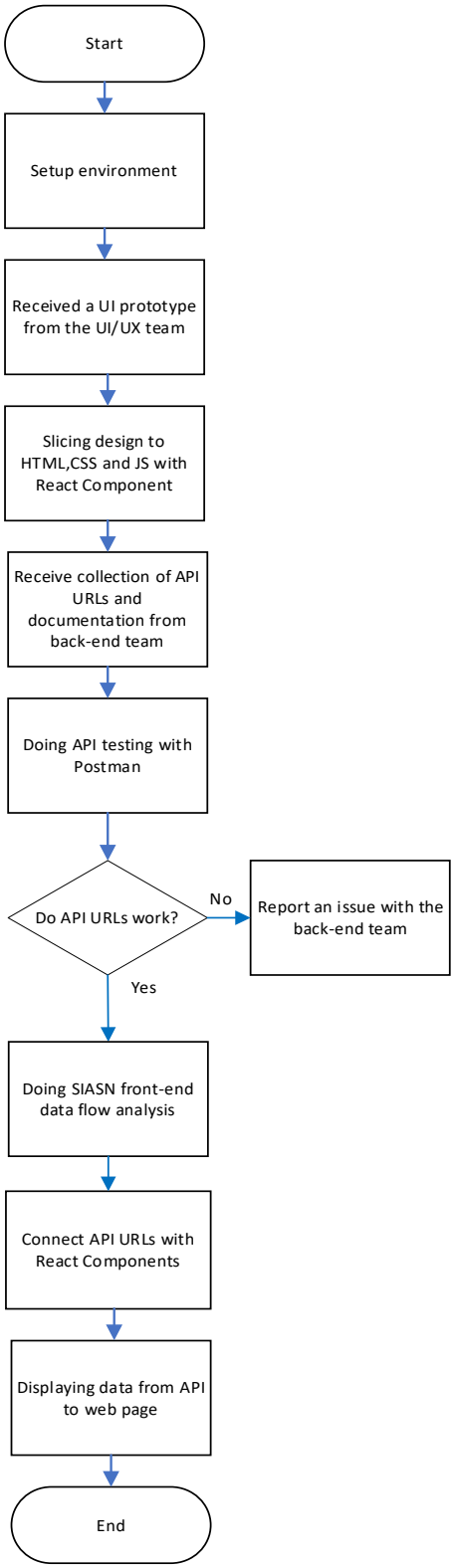

Figure 4. Stages of Making Front-End SIASN

In slicing the SIASN front-end design into a React Component with HTML, CSS and JS code, the researcher analyzed the component structure into reusable (dynamic) and static components. Components that have the outermost structure such as headers, sidebars and footers are categorized as reusable components. As for the page content in the application, which changes frequently according to the service name and menu, it is made into a dynamic component that is wrapped into a layout component. Layout components are components that already have a header, sidebar and footer structure for the application as a template for a component with suitable content.

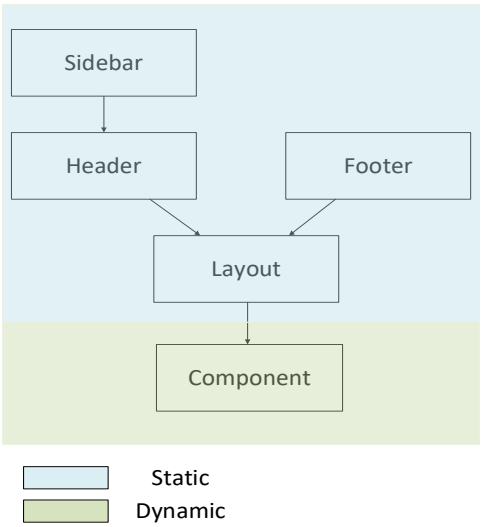

Figure 5. SIASN app slicing structure

C. *Supports Client Side Rendering*

Supports CSR in rendering website pages, so the load on the server can be reduced.

V. DISADVANTAGES OF REACT.JS

The disadvantages of the React.js library based on literature review and case studies of the SIASN application are explained as follows:

A. *State persistence*

Data contained in the state of a component can be lost when the web is refreshed. For this reason, local storage is needed to store state data. However, the local storage capacity is limited, which is as much as 5 MB.

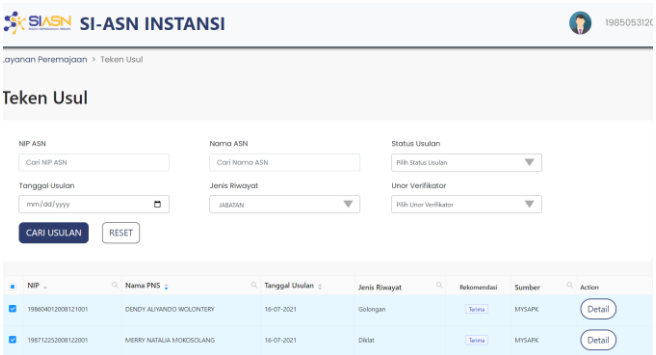

Figure 6. Table of first page teken usul

Mochammad Fariz Syah Lazuardy and Dyah Anggraini, "Modern Front End Web Architectures with React.Js and Next.Js," *International Research Journal of Advanced Engineering and Science*, Volume 7, Issue 1, pp. 132-141, 2022.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

In SIASN Instansi app rejuvenation service (layanan peremajaan) with menu teken usul tolak usulan, an example of using local storage is used to store the proposal id at the beginning of the page when the user selects some data in the table for the proposal sign to be approved or rejected on the next page.

When the user checks the ant design table, the details of the proposed data will be stored in local storage so that it can be used on the next page, namely the reject or approve proposal page. The following snippet of program code saves the proposed data to local storage.

```
const rowSelection = {
  onChange: (selectedRowKeys,selectedRows) => {
    console.log('selectedRowKeys: ', selectedRowKeys);
    console.log('selectedRows: ', selectedRows);
    localStorage.setItem('penampungID',
JSON.stringify(selectedRows))
  },
  getCheckboxProps: (record) => ({
    disabled: record.name === 'Disabled User',
    // Column configuration not to be checked
    name: record.name,
  }),
};
```

Then after checking the ant design table on the first page of menu teken usul, then when the user clicks on the next button to move to the step-2 page the selected data will be stored in the Google Chrome's local storage with the name of penampung ID.

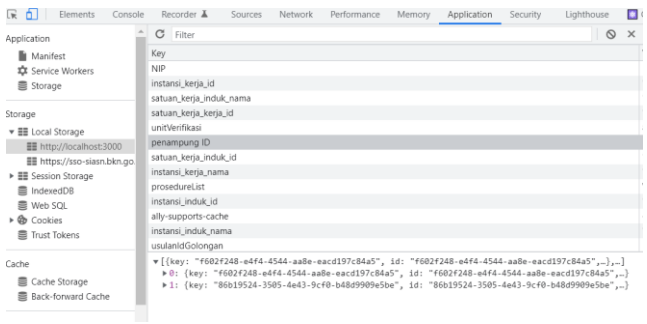

Figure 7. Local storage penampung ID table teken usul

Penampung ID variable stored in local storage, serves to sort the selected data in the first page table, to be displayed in the second table with the choice of reject or approve the proposal.

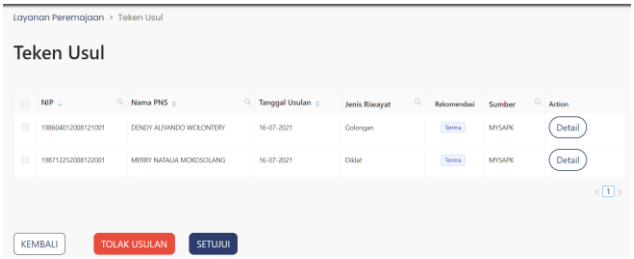

Figure 8. Table of second page teken usul

The data in the table on the second page of teken usul is the data from local storage that was previously stored. The following snippet of program code takes the proposed data stored from local storage and stored in the state to be displayed again in the second table.

```
function TekenUsulanPeremajaan2() {
  const router = useRouter()
  const [modalShow, setModalShow] =
React.useState(false)

  const [getTekenUsul, setTekanUsul] = useState([])
  const [passphrase, setPassphrase] = useState()
  // let passphrase

  const dispatch = useDispatch()

  useEffect(function () {
    const data =
      localStorage.getItem('penampung ID') !== null
        ? JSON.parse(localStorage.getItem('penampung
ID'))
        : []
    setTekanUsul(data)
  }, [])
```

The following is a snippet of program code to send data that is temporarily stored in the state via props to the component table teken usul on the second page.

```
return (
  <>
    {/* breadcrumb */}
    <Breadcrumb></Breadcrumb>

    {/* title */}
    <div className='wizard__title'>
      <h2 className='mb-5'>Teken Usul</h2>
    </div>

    <TablePU2
dataTable={getTekenUsul}></TablePU2>

    {/* Component input form */}
    <div className='paddingDown3'></div>

    <div className=' wizard__inputForm-buttonBox
mt-3'>
      <Link
href='/layananPeremajaan/persetujuanUsulan'>
        <Button className='mr-5 mb-5
wizard__formButtonReset' type='submit'>
          Kembali
        </Button>
      </Link>
      <Link
href='/layananPeremajaan/persetujuanUsulan/tolak'>
        <Button className='mr-4 mb-5
wizard__formButtonTolak' type='submit'>
```


Mochammad Fariz Syah Lazuardy and Dyah Anggraini, "Modern Front End Web Architectures with React.Js and Next.Js," *International Research Journal of Advanced Engineering and Science*, Volume 7, Issue 1, pp. 132-141, 2022.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

```
                        Tolak Usulan
                    </Button>
                </Link>
                <Button
                    className='mb-5
wizard__formButtonSubmit'
                    type='button'
                    // onClick={(e) => handleSetujui(e)}
                    onClick={() => setModalShow(true)}>
                    Setujui
                </Button>
            </div>
```

B. *White Screen or Blank Screen Initial Load*

The React.js library basically supports CSR, so the initial load causes a relatively long blank screen.

Unlike Next.js which uses SSR, in plain React.js applications it uses the CSR method to display web views. In Figure 9 it can be seen that on the network tab, no HTML files are displayed first. The html file of the CSR method is bundled in a Java Script file, so there is no concept of pre-rendering. This allows users to have blank screen experience on initial load.

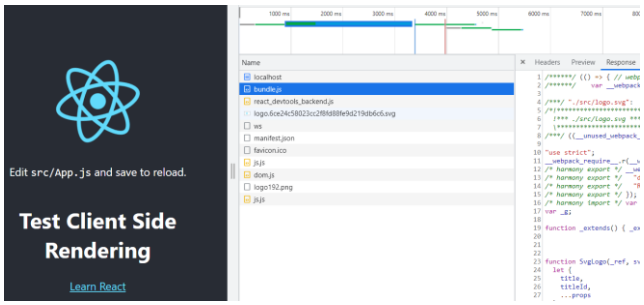

Figure 9. Tab network CSR with plain React.js

In Figure 10 it can be seen, when the user reloads the page, he only sees a white screen. This is because the HTML file is still being downloaded in Java Script bundle.

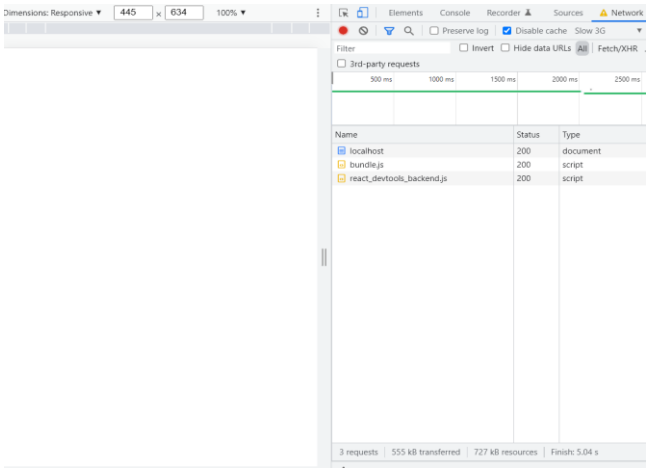

Figure 10. Initial load CSR with plain React.js

C. *Page Routing with Third Party Library*

The React.js library should require the installation of additional libraries such as react-router-dom for navigation.

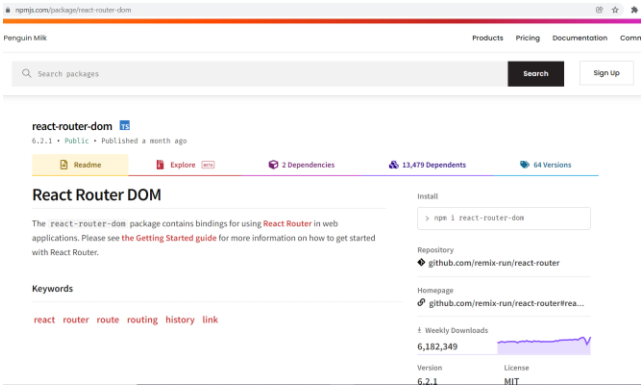

Figure 11. Library React router DOM

D. *Only Supports Data Fetching from Client-Side*

The React.js library only supports data fetching from the client-side by using the useEffect function so it takes extra time to process data fetching when the web page loads.

E. *Prop Drilling*

Basically, for data communication between React.js components, only use props. However, using props has a weakness, when the communication component has a level too far, it will cause prop drilling. To overcome this, it requires the installation of third-party libraries such as react-redux so that the state is global and can be accessed by any component level.

F. *Not Good for SEO*

Using Client-side rendering is not good for the search engine optimization (SEO) process, because the HTML file is rendered on the client-side.

VI. ADVANTAGES OF NEXT.JS

The advantages of the Next.js framework based on literature review and case studies of the SIASN application are explained as follows:

A. *Buiilt-in CSS Support*

The Next.js framework already has built in CSS support. At the beginning of the installation there is a global css file and modules for certain components.

Global.css is a CSS file in the Next.js framework that is global, which means that all CSS code is accessible for all components. Whereas module.css is generally intended for certain components.

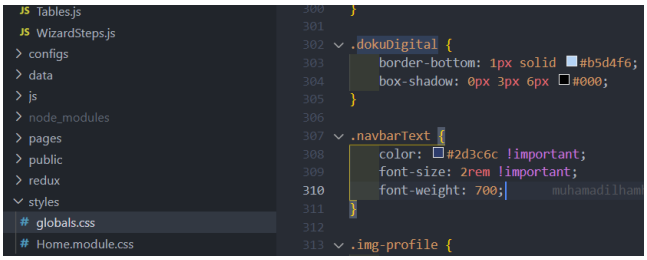

Figure 12. Built in CSS support

Mochammad Fariz Syah Lazuardy and Dyah Anggraini, "Modern Front End Web Architectures with React.Js and Next.Js," *International Research Journal of Advanced Engineering and Science*, Volume 7, Issue 1, pp. 132-141, 2022.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

B. *Easy Routing Mechanism*

The Next.js framework already has easy routing mechanism support located in the pages folder, so it doesn't need a third-party library to do routing.

Routing the React component is quite easy by writing the file name directly in the pages folder provided by Next.js. Then the URL address bar automatically points to the name of the file in the pages folder.

After completing the component stage and creating a routing file in Next.js' pages folder, the researcher imports the components contained in the components folder into a routing file that has been wrapped in component Layout as a static component so that a component can be displayed during the deployment process.

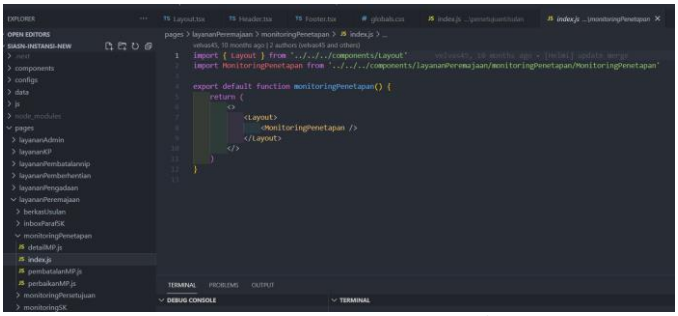

Figure 13. Import component on pages folder

In Figure 14 it can be seen that when the researcher accesses the URL in the address bar, it will automatically call the imported component in the routing file.

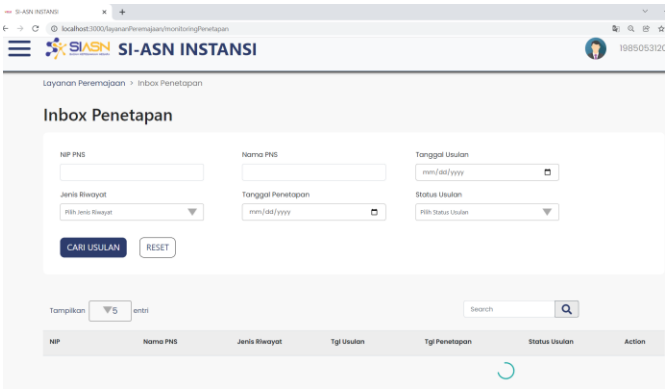

Figure 14. Components are displayed during the routing process.

C. *Pre-rendering*

The Next.js framework already supports SSR in rendering web pages, so makes the user will not see a blank page in the initial load and reduced the load on browser.

It can be seen in Figure 15 during the build process, the SIASN application uses the SSR method in displaying web pages.

It can be seen in Figure 16 with the SSR mechanism, the proposal monitoring (monitoring usulan) of rejuvenation serivce (layanan peremajaan) displays the HTML file first, while the Java Script file is being downloaded to be able to interact and display data.

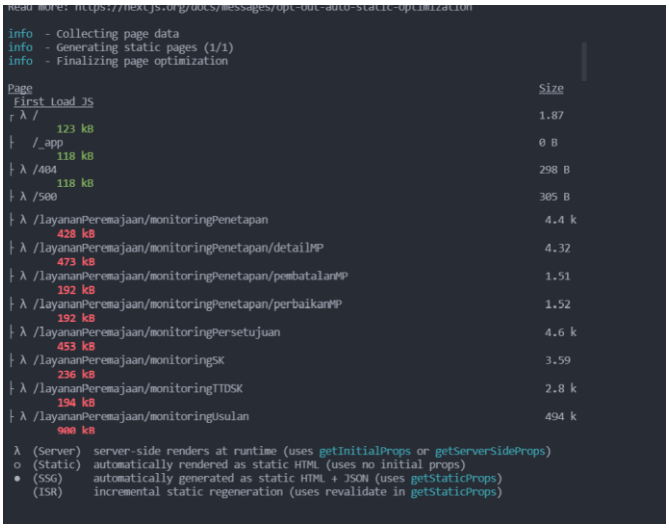

Figure 15. SSR on SIASN app during build process

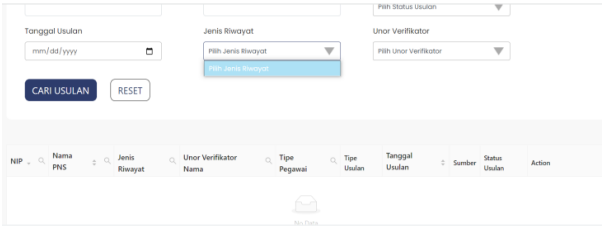

Figure 16. Pre-Rendering with Next.js

Figure 17 shows the sequence in which files are downloaded on the network browser tab. Next.js will first download the HTML file, before downloading Java Script file.

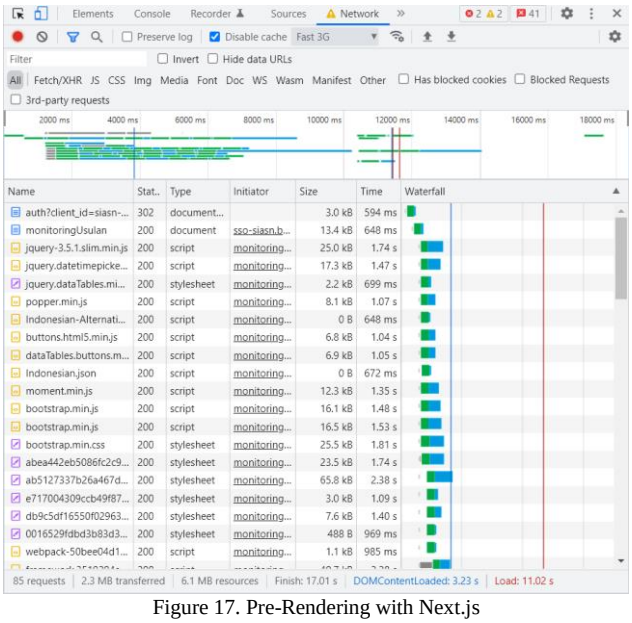

Figure 17. Pre-Rendering with Next.js

Figure 18 shows that on the network browser tab, Next.js first response will display the HTML file that has been rendered from the server side, before the Java Script file that makes the web interactive. This makes the user will not see a blank page in the initial load, but as a drawback this method

Mochammad Fariz Syah Lazuardy and Dyah Anggraini, "Modern Front End Web Architectures with React.Js and Next.Js," *International Research Journal of Advanced Engineering and Science*, Volume 7, Issue 1, pp. 132-141, 2022.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

requires reloading the page to request rendered HTML files from server.

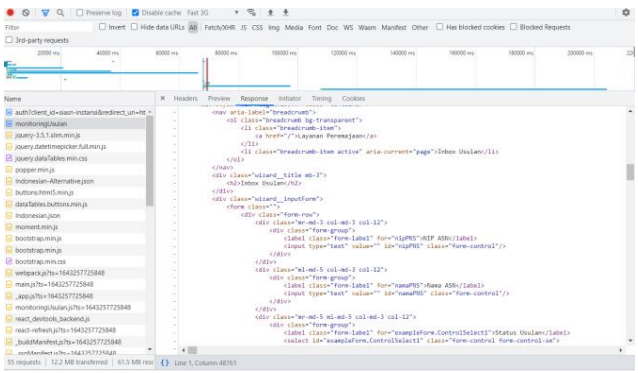

Figure 18. Tab network response SSR with Next.js

D. *Various Data Fetching Mechanism*

Unlike plain React.js App which only supports data fetching with CSR, on official documentation the Next.js framework supports data fetching mechanisms by CSR, SSR, SSG and ISR. So that the data fetching mechanism can be adjusted based on application needs.

E. *Pre-rendering*

On official documentation, Next.js supports pre rendering with SSR and SSG methods so it can be adjusted based on application needs. In SSG mechanism, pre-rendering method generates the HTML during the build process. Then pre-rendered is reused on each request. While SSR mechanism, pre-rendering method generates the HTML on each request.

It can be seen in Figure 19 during the build process, the SIASN application uses the SSR method in displaying web pages.

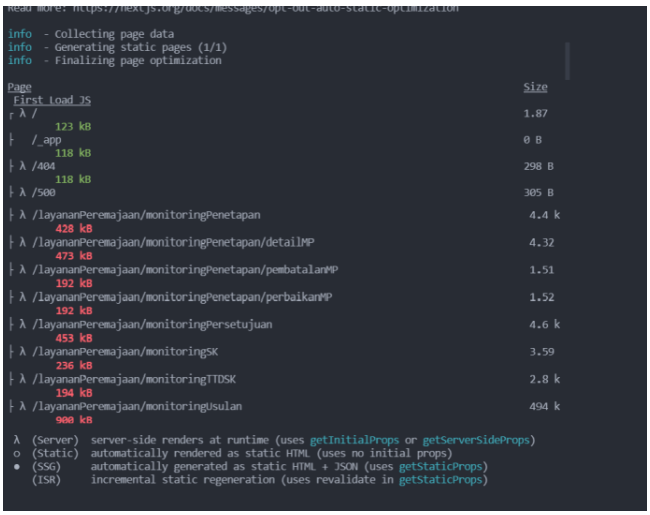

Figure 19. SSR on app during build process

It can be seen in Figure 20 with the SSR mechanism, the proposal monitoring (monitoring usulan) of rejuvenation serivce (layanan peremajaan) displays the HTML file first, while the Java Script file is being downloaded to be able to interact and display data.

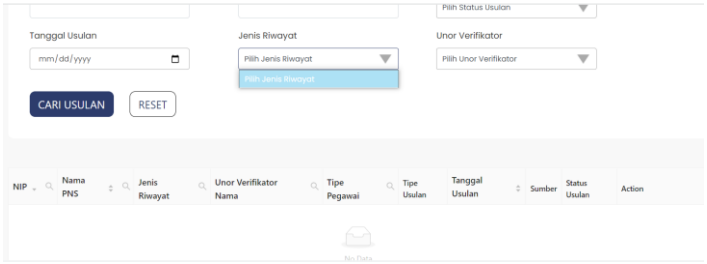

Figure 20. Pre-Rendering with Next.js

Figure 21 shows the sequence in which files are downloaded on the network browser tab. Next.js will first download the HTML file, before downloading Java Script file.

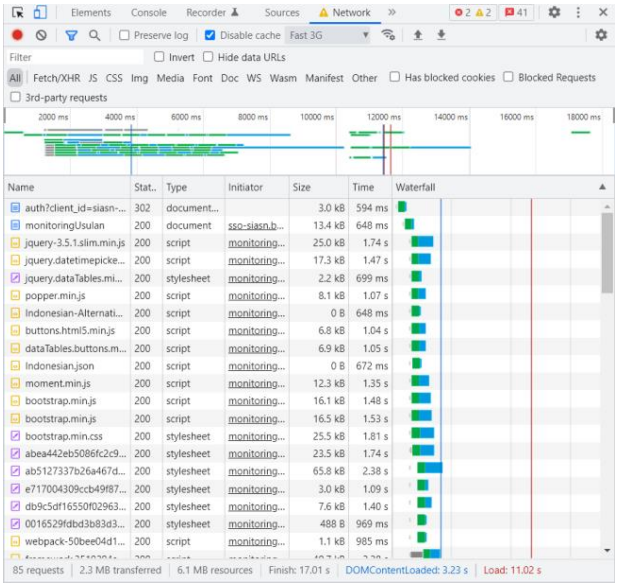

Figure 21. Tab network Pre-Rendering with Next.js

Figure 22 shows that on the network browser tab, Next.js' first response will display the HTML file that has been rendered from the server side, before the Java Script file that makes the web interactive. This makes the user will not see a blank page in the initial load, but as a drawback this method requires reloading the page to request rendered HTML files from server.

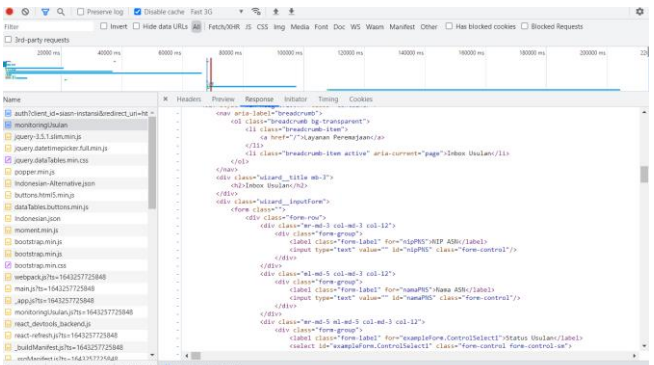

Figure 22. Tab network response with Next.js

F. *Good for SEO*

Using SSR (server-side rendering) is good for SEO (search

Mochammad Fariz Syah Lazuardy and Dyah Anggraini, "Modern Front End Web Architectures with React.Js and Next.Js," *International Research Journal of Advanced Engineering and Science*, Volume 7, Issue 1, pp. 132-141, 2022.

engine optimization), rather than CSR (client-side rendering) because HTML files are rendered on the server side.

VII. Disadvantages of NEXT.JS

The disadvantages of the Next.js framework based on literature review and case studies of the SIASN application are explained as follows

A. High Server Load

Because in SIASN application, Next.js framework using the SSR (server-side rendering) method, hence the load on the server increases due to frequent requests to the server.

B. Have to Wait for Site to Become Interactive

Pre-rendering in the SSR (server-side rendering) method can lead to the user enters the page, then the HTML file has been rendered and provided by the server. However, the user can only view the page view and cannot interact with the page until the browser successfully executes React.

C. Full Page Reload

By using SSR (server-side rendering), when changing pages, user would experience full page reload because user have to request an HTML file that has been rendered on the server.

References

[1] F. A. Prasetyo, "Badan Kepegawaian Negara (BKN)," *tribunnewswiki*, 2019. https://www.tribunnewswiki.com/2019/10/24/badan-kepegawaian-negara-bkn (accessed Nov. 02, 2021).

[2] K. R. I. B. Yogyakarta, "Wakil Kepala Bkn: Siasn Solusi Benahi Kualitas Data Kepegawaian," *Yogyakarta.Bkn.Go.Id*, 2021. https://yogyakarta.bkn.go.id/berita/2021/10/wakil-kepala-bkn-siasn-solusi-benahi-kualitas-data-kepegawaian (accessed Nov. 02, 2021).

[3] F. Falih, "A Review Study of Information Systems," *Int. J. Comput. Appl.*, vol. 179, no. 18, pp. 15–19, 2018, doi: 10.5120/ijca2018916307.

[4] B. Venkat, S. Indla, Y. Puranik, P. G. Student, and P. E. S. M. College, "Review on React JS," vol. 5, no. 4, pp. 1137–1139, 2021.

[5] A. Bhalla, S. Garg, and P. Singh, "Present Day Web-Development Using ReactJS," *Int. Res. J. Eng. Technol.*, vol. 7, no. 5, pp. 1154–1157, 2020.

[6] P. S. Maratkar and P. Adkar, "React JS – An Emerging Frontend Javascript Library Virtual DOM React One-Way Data Flow JSX Syntax," vol. 4, no. 12, pp. 99–102, 2021.

[7] M. Platforms, "Component and Props," 2022. https://reactjs.org/docs/components-and-props.html (accessed Jan. 25, 2022).

[8] F. Halili and E. Ramadani, "Web Services: A Comparison of Soap and Rest Services," *Mod. Appl. Sci.*, vol. 12, no. 3, p. 175, 2018, doi: 10.5539/mas.v12n3p175.

[9] U. Singh, "REST API Framework : Designing and Developing Web Services," *Int. Res. J. Eng. Technol.*, vol. 8, no. June, pp. 815–817, 2021.

[10] S. G. V and A. Sandeep, "Comprehensive Analysis of React-Redux Development Framework," *Int. J. Creat. Res. Thoughts www.ijcrt.org*, vol. 8, no. 4, p. 4230, 2020, [Online]. Available: www.ijcrt.org.

[11] P. Kishore and M. B M, "Evolution of Client-Side Rendering over Server-Side Rendering," vol. 3, no. 2, pp. 1–10, 2020.

[12] Vercel, "Data Fetching Overview," 2022. https://nextjs.org/docs/basic-features/data-fetching/index (accessed Jan. 24, 2022).