

Student's Name

Professor's Name

Course Number

Date

Pre-rendering with Next.js

Next.js is a React.js framework with static generation and server-side rendering capabilities. Pre-rendering is an essential feature that gives a Next.js app a significant performance boost and advantage in terms of search engine optimization (SEO) compared to a traditional React app. Better performance and search engine optimization improves user experience and broadens user base coverage.

Client-side Rendering and Pre-rendering

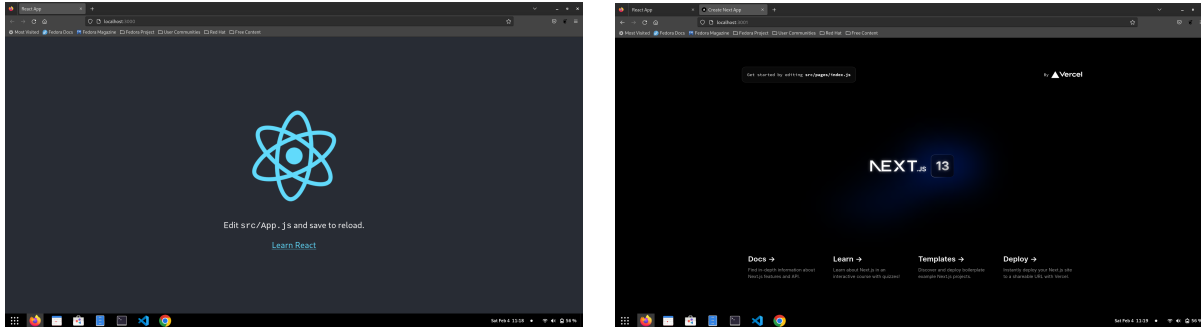
A typical React application uses client-side rendering. This is where minimal HTML is sent to the client side with an id root and a JavaScript bundle file that will initialize the react components to create the dom nodes through hydration (Patadiya). On the other hand, pre-rendering is where the HTML is generated on the server side and sent to the browser.

Demonstration

Two projects, "react-app" and a "next.js app," demonstrate pre-rendering. The two apps are created by running the commands "npx create-react-app my-app" and "npx create-next-app@latest" on the terminal, respectively. The folder containing the two apps is run on Virtual studio (VS) code. By navigating to the respective folder, the two apps are run by the "npm start" and "npm run dev" commands, respectively, where the user interfaces are rendered on the browser.

Figure 1

User interfaces for React.js app and Next.js app, respectively



The two user interfaces seem to work the same way; however, their source codes highlight the differences in their functionalities. In figure 2, the traditional react app source code only has minimal HTML and an empty “div” tag, whereas, in the Next.js app, the source code contains all the HTML for the entire app.

Figure 2

Source codes for React.js app and Next.js app, respectively

```

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8" />
5     <link rel="icon" href="/favicon.ico" />
6     <meta name="viewport" content="width=device-width, initial-scale=1" />
7     <meta name="theme-color" content="#000000" />
8     <meta
9       name="description"
10      content="Web site created using create-react-app"
11    />
12    <link rel="apple-touch-icon" href="/logo192.png" />
13    <!--
14      manifest.json provides metadata used when your web app is installed on a
15      user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/
16    -->
17    <link rel="manifest" href="/manifest.json" />
18    <!--
19      Notice the use of in the tags above.
20      It will be replaced with the URL of the 'public' folder during the build.
21      Only files inside the 'public' folder can be referenced from the HTML.
22

```

Pre-rendering and Performance

Pre-rendering improves the performance of the application in the following ways. In a traditional react app, since only the minimal HTML and the bundled JavaScript are downloaded, the user interface will only render after the JavaScript executes; hence, a blank page is rendered on the browser in the interval between execution and rendering (Gardón). On the other side, pre-rendering reduces load time, since the HTML is readily available on download, reducing user wait time.

Pre-rendering and Search-engine Optimization

In applications such as blogs and ecommerce, search engine optimization is a concern (Gardón). With a react app, a search engine will only see a “div” tag with an ID “root” if it hits the application page. In a pre-rendered app, all the content is readily available; hence, the code is easily indexed and cached by CDN, and, therefore, easy to search during engine optimization.

Next.js improves performance and has proven to provide better SEO as compared to a traditional React.js app. However, this does not mean that all developers should shift to Next.js, since there are still some merits of React.js over Next.js. Therefore, if performance and SEO are of concern, a developer should consider Next.js over React.js in their development and production.

Works Cited

Gardón, Diego S. "Next.js Vs. React: The Difference & Best Frontend Framework." *Snipcart*,

8 Apr. 2022, snipcart.com/blog/next-js-vs-react. Accessed 4 Feb. 2023.

Patadiya, Jaydeep. "Next JS Vs React : Which Framework to Choose for Front End in

2022?" *Radixweb*, 19 Oct. 2022, radixweb.com/blog/nextjs-vs-react. Accessed

4 Feb. 2023.