

## **SERVER-SIDE РЕНДЕРИНГ В NEXT.JS. ОСОБЛИВОСТІ, ПЕРЕВАГИ ТА НЕДОЛІКИ**

Server-side рендеринг (SSR) – це підхід до рендерингу веб-сторінок, при якому HTML генерується на сервері та відправляється до клієнта у готовому вигляді. Цей підхід важливий для проектів, що мають індексуватися пошуковими системами.

Переваги SSR:

- покращена SEO-оптимізація, завдяки тому, що повний HTML передається на клієнт, пошукові системи можуть індексувати сторінки з динамічним контентом так само легко, як статичні сторінки;
- швидкий рендеринг для користувача, що дає користувачу отримати готову HTML одразу, без необхідності чекати на завантаження JavaScript для створення контенту;
- підтримка персоналізації дозволяє серверу динамічно рендерити сторінки залежно від запитів, зокрема, на основі автентифікації чи локалізації;
- безпечний доступ до даних, а це означає, що всі запити виконуються на сервері, API-ключі та інша конфіденційна інформація не розкриваються клієнту.

До недоліків SSR можна віднести:

- збільшення часу відповіді сервера, тому що генерація сторінки на сервері додає затримку перед відправкою відповіді, особливо для складних запитів до баз даних;
- залежність від сервера при високому навантаженні на сервер, тому що можуть виникати проблеми з продуктивністю, особливо для великих проектів із високою відвідуваністю;
- складність реалізації із-за того, що проекти з SSR часто мають складнішу архітектуру через необхідність керувати серверним рендерингом і клієнтськими інтерактивними компонентами;
- обмеження кешування, тому що динамічний рендеринг зменшує ефективність використання кешу, оскільки сторінки можуть бути унікальними для кожного запиту.

SSR вирішує одну з основних проблем React – відправка неповного HTML при переходу на сторінку, що унеможливує SEO-оптимізацію та індексацію пошуковими системами. Існує безліч інструментів і технологій для генерації статичних сторінок: Next.js, Gatsby, Vite. Однак серед кількох рішень, для виконання нашого проекту обрано Next.js через його гнучкість, широкий спектр можливостей, високу продуктивність та швидкість розробки, що забезпечується потужним та інтуїтивним для програміста інструментарієм.

Слід відмітити деякі особливості SSR з використанням Next.js:

- виконання динамічного рендерингу, що забезпечує формування HTML-сторінок для кожного запиту до сервера. Це дозволяє обробляти специфічні запити в залежності від параметрів URL або автентифікації користувача;
- можливість інтеграції з API, що дає можливість серверу отримувати дані із зовнішніх джерел перед рендерингом сторінки. Це забезпечує повністю готовий до взаємодії контент для користувача.
- виконання SEO-оптимізації, тому що генерується повний HTML-контент, який пошукові системи можуть легко індексувати;
- використання спеціальних методів, які дозволяють налаштовувати рендеринг на сервері. Вони забезпечують контроль над тим, які дані мають бути завантажені перед відправкою сторінки користувачу.

Висновки. Отже, SSR Next.js є потужним інструментом для створення динамічних, SEO-оптимізованих додатків, де важлива швидкість завантаження та персоналізація контенту. Однак, необхідно враховувати компроміс між продуктивністю сервера та потребами користувачів, щоб обрати правильну стратегію рендерингу для конкретного проекту.

### **Список використаних джерел**

1. Node.js v23.3.0 Documentation. Node.js – Run JavaScript Everywhere. URL: <https://nodejs.org/docs/latest/api/> (date of access: 18.11.2024).
2. Next.js. The React Framework. URL: <https://nextjs.org/docs> (date of access: 18.11.2024).