

ДОДАТОК А

Перелік джерел посилання за науковими напрямками керівника та науковців
кафедри програмної інженерії

9. Брухтій С.С., Ревенчук І.А. Тенденція розвитку методів рендеренгу web-сайтів для пошукової оптимізації сайтів (SEO): матеріали 28-го Міжнарод. молодіжного форуму «Радіоелектроніка та молодь у XXI столітті». Зб. матеріалів форуму. Т. 6., Харків: ХНУРЕ. 2024. С.918-919. (DOI: <https://doi.org/10.30837/IYF.IIS.2024.918>).

10. Мандрика М. Ревенчук І.А. Дослідження методів та інструментів моніторингу веб-сервісів: матер XV Міжнар наук.-практ конф. "Innovative Approaches to the Progressive Solution of Scientific Research Problems". Іспанія 27-29.03.2024. Р.59-61.

11. Моруга Д.І., Ревенчук І.А. Unified Approach for Development of Graphical User Interface in Cross-platform Applications: XXII International Scientific and Practical Conference «Modern Scientific Research: Theoretical and Practical Aspects». Oslo, Norway 8-10.05.2024. Р.73-76.

ДОДАТОК Б

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ



Ім'я користувача:
Олійник Олена Володимирівна каф. ПІ

ID перевірки:
1016354973

Дата перевірки:
13.06.2024 06:06:54 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
13.06.2024 06:20:06 EEST

ID користувача:
100012353

Назва документа: 2024_М_ПІ_ІПЗм_22_3_Брухтій_С_С_скорочений

Кількість сторінок: 39 Кількість слів: 6940 Кількість символів: 50084 Розмір файлу: 941.40 KB ID файлу: 1016159072

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

1.25%
Схожість

Найбільша схожість: 0.26% з Інтернет-джерелом (<https://almanac.httparchive.org/es/2019/performance>)

0.76% Джерела з Інтернету

8

Сторінка 41

0.49% Джерела з Бібліотеки

7

Сторінка 41

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

3

Підозріле форматування

7
сторінок

ДОДАТОК В

Слайди презентації

Харківський національний університет радіоелектроніки

1

Кафедра програмної інженерії

Кваліфікаційна робота магістра

Дослідження методів рендерингу web-сайтів для пошукової оптимізації сайтів (SEO)

Виконав: ст. гр. ІПЗм22-3 Брухтій С.С.

Керівник: доцент каф. ППревенчук І.А.

2

Актуальність дослідження

- ефективний рендеринг веб-сайтів
- дослідження різних методів рендерингу
- оптимальний метод рендерингу для покращення SEO
- важливість для бізнесу

Аналіз проблемної галузі

SEO (Search Engine Optimization)

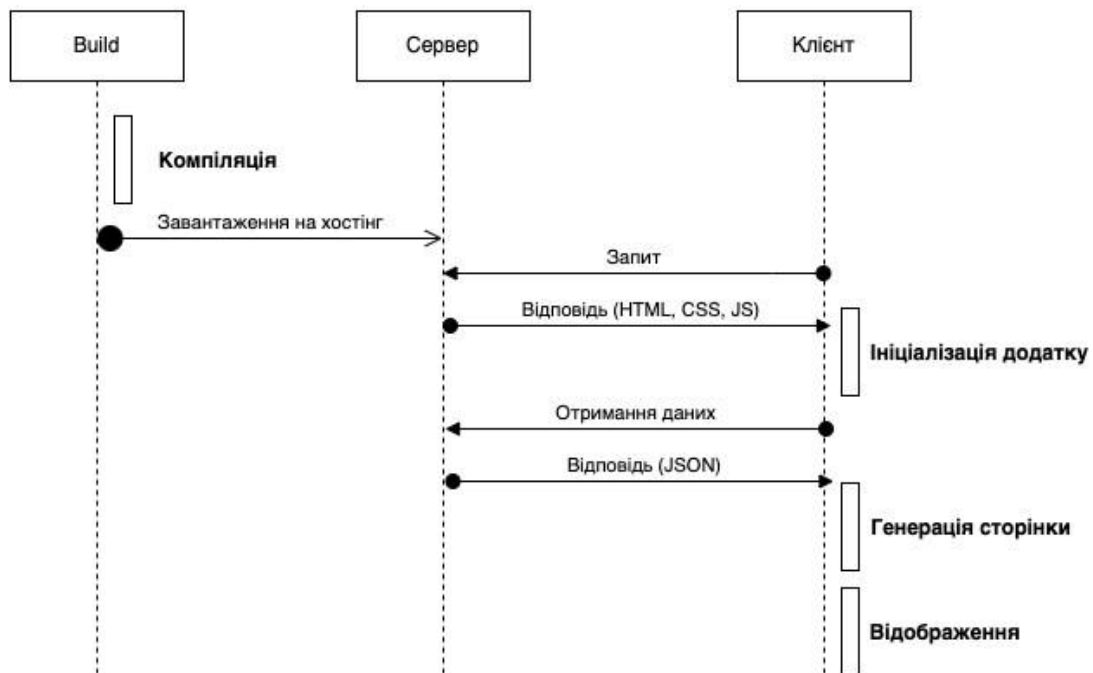
Зовнішнє:

- Лінкбїлдінг (створення зовнішніх посилань)
- Активність у соціальних мережах
- Відгуки та рейтинги:
- Гостьові пости

Внутрішнє:

- Мета-теги
- Структура сайту
- Швидкість завантаження сторінки
- Чистий та структурований код

Аналіз проблемної галузі - CSR



Аналіз проблемної галузі - CSR

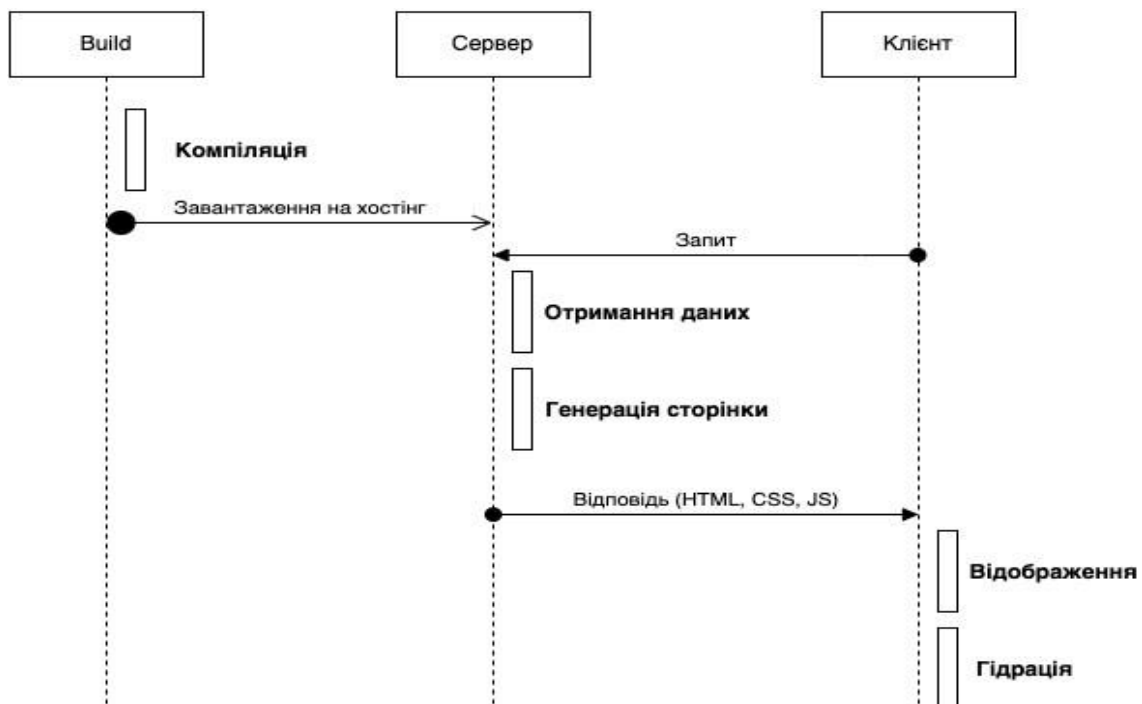
Переваги:

- інтерактивність
- швидкі відповіді на дії
- створення інтерактивних додатків

Недоліки:

- індексація контенту
- повільне початкове завантаження
- проблеми з мета-тегами

Аналіз проблемної галузі - SSR



Аналіз проблемної галузі - SSR

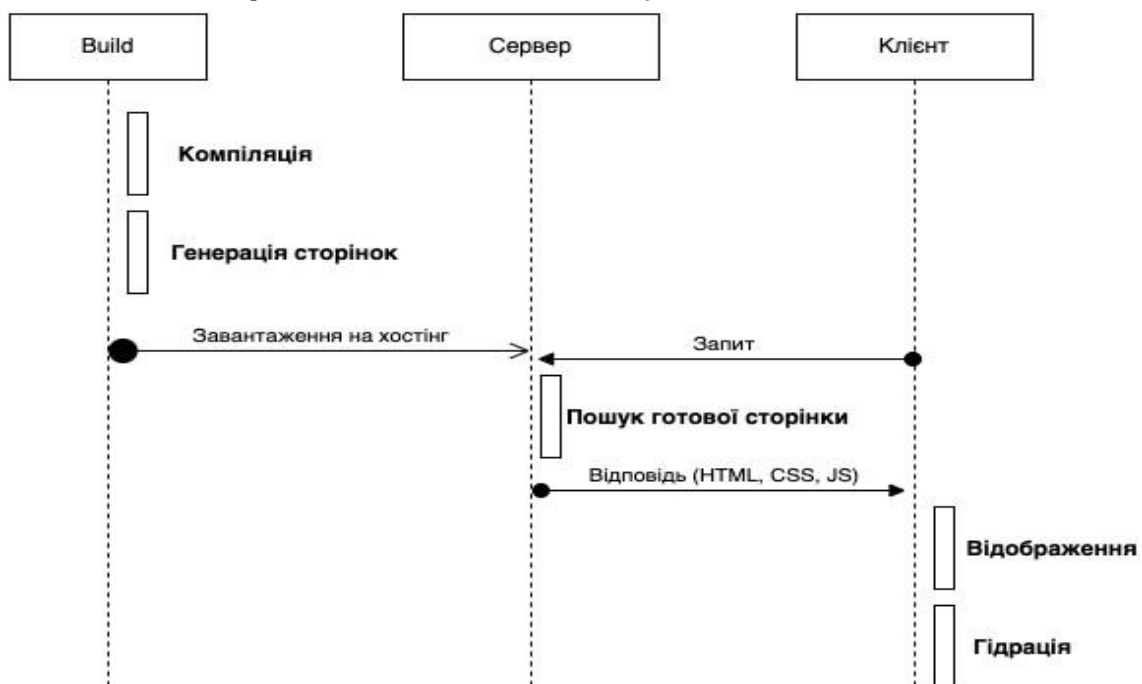
Переваги:

- швидке відображення контенту
- Пошукові роботи легше знаходять

Недоліки:

- навантаження на сервер
- проблеми з кешуванням
- складність реалізації

Аналіз проблемної галузі - SSG



Аналіз проблемної галузі - SSG

Переваги:

- висока швидкість завантаження
- пошукові роботи легко знаходять

Недоліки:

- тільки статичний контент
- час генерації
- складність реалізації

Постановка задачі

Необхідно:

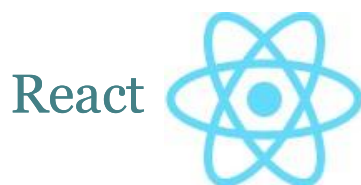
- розробити методику оцінки ефективності різних методів рендерингу в контексті SEO
- провести експериментальні дослідження з метою порівняння різних методів рендерингу
- виокремити найкращі практики оптимізації рендерингу для пошукових систем.

Критерії оцінки процесу рендерингу

Критерії за версією Google Web Vitals:

- Time to First Byte (TTFB)
- First Contentful Paint (FCP)
- Largest Contentful Paint (LCP)
- ~~First Input Delay (FID)~~ -> Largest Contentful Paint (LCP)

Засоби проведення дослідження



Веб сторінка для досліджу

Веб сторінка включає:

- структура: заголовки, абзаци, картинки, інші теги
- Мета дані
- Посилання
- JavaScript інтерактивність (особливо для CSR)

Реалізація веб застосунків

```
const getComponent = (index) =>
  <div class="item" key=${index}>
    <div class="item-image">
      <img src=${imagesLinks[index % 10]} alt="Item Name"/>
      <div class="item-actions">
        <button type="button" class="add-to-cart-btn">Add to Cart
      </div>
    </div>
    <div class="item-details">
      <h3 class="item-title">Item Title: ${index + 1}</h3>
      <p class="item-description">This is a brief description of
      <a href="https://hidden-refuge-97632-934ef58eadcb.herokuapp.com">
      <div class="item-meta">
        <span class="item-price">${19.99}</span>
        <span class="item-rating">Rating: ★★★★★</span>
      </div>
    </div>
  </div>;
```

Шаблон
компоненту

Генератор компонентів

```
export const ComponentsBlock = ({ componentsQty }) => {
  const getComponents = () => {
    const arrayWithObjects = Array.from({ length: componentsQty }, (_, index) => (getComponent(index)));
    return arrayWithObjects;
  }

  return (
    <div className="components-block">
      <h1 className="font-weight-light components-qty">Components quantity: {componentsQty}</h1>
      <div id="components" className="components">
        {getComponents()}
      </div>
    </div>
  );
};
```

Реалізація веб застосунків: Node.js

```
.post('/getSSR', (req, res) => {
  const componentsQty = req.body.componentsQty;

  const componentBlock = getComponentsBlock({ componentsQty });
  res.send(componentBlock);
})
```

POST SSR

POST SSG

```
.post('/getSSG', (req, res) => {
  const componentsQty = req.body.componentsQty;

  const file = `generatedHtml/${componentsQty}.html`;
  res.sendFile(file, {root: __dirname })
})
```

The screenshot shows a file explorer on the left with a directory structure: .github, assets, components, JS components.js, generatedHtml (containing 100.html, 250.html, 500.html, 1000.html), node_modules, public, views, .env, .gitignore, app.json, and htmlGenerator.js. The main window displays the content of '100.html', which is an HTML snippet for a component. The code includes a title 'Components', a list of items with an image, a 'button type="button" class="add-to-cart-btn"', and item details like 'Item Title: 1', a description, a price of '\$19.99', and a rating of 'Rating: ★★★★★'.

Згенеровані HTML файли

Реалізація веб застосунків: SSR та SSG

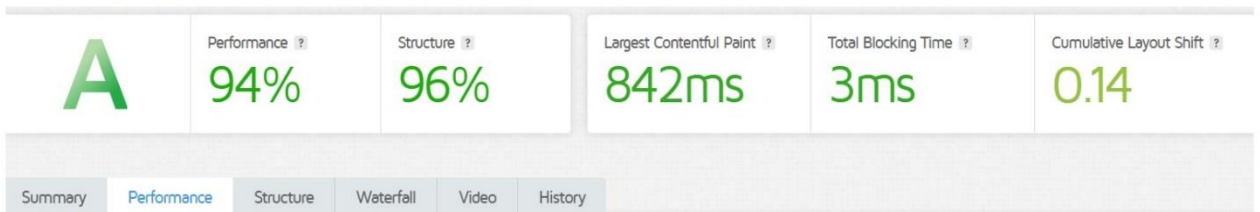
The screenshot shows a code editor with two files: 'App.js' and 'ComponentsBlock.js'. In 'App.js', a function 'App()' returns a JSX element with a 'ComponentsBlock' component and a prop 'componentsQty={1000}'. In 'ComponentsBlock.js', an asynchronous function 'requestsSSRComponents' sends a POST request to 'https://brukhtii-node-js-ssr-ssg-4dfd839344ba.herokuapp.com/getSSR'. The 'ComponentsBlock' component uses 'useState' and 'useEffect' to fetch data from 'requestsSSRComponents' and set it to the state. The component returns a 'div' with 'dangerouslySetInnerHTML' set to the fetched HTML.

Реалізація веб застосунків: CSR

```

src > components > JS ComponentsBlockjs > ...
5 export const ComponentsBlock = ({ componentsQty }) => {
6
7   const getComponent = (index) => ...
8   </div>;
9
10  const getComponents = () => {
11    const arrayWithObjects = Array.from({ length: componentsQty
12    return arrayWithObjects;
13  }
14
15  return (
16    <div className="components-block">
17      <h1 className="font-weight-light components-qty">Components
18      <div id="components" className="components">
19        {getComponents()}
20      </div>
21    </div>
22  );
23
24 JS Appjs x
25 src > JS Appjs > App
26 1 import React from "react";
27 2 import { ComponentsBlock } from "../components/ComponentsBlock";
28 3
29 4 function App() {
30 5   return (
31 6     <div className="app">
32 7       <h1>Web page to test CSR in SEO</h1>
33 8       <ComponentsBlock componentsQty={1000} />
34 9     </div>
35 10  );
36 11 }
37 12 export default App;
  
```

Збір даних



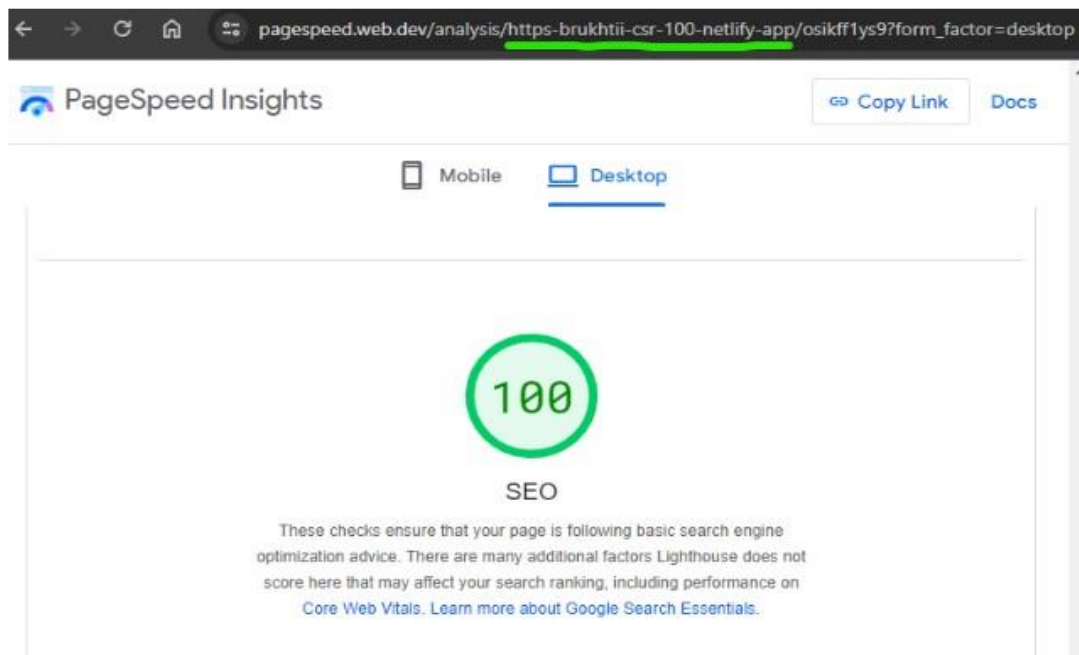
Performance Metrics

The following metrics are generated using Lighthouse Performance data.

Metric details

First Contentful Paint How quickly content like text or images are painted onto your page. A good user experience is 0.9s or less. Learn more.	Good - Nothing to do here 734ms	Time to Interactive How long it takes for your page to become fully interactive. A good user experience is 2.5s or less. Learn more.	Good - Nothing to do here 791ms
Speed Index How quickly the contents of your page are visibly populated. A good user experience is 1.3s or less. Learn more.	Good - Nothing to do here 759ms	Total Blocking Time How much time is blocked by scripts during your page loading process. A good user experience is 150ms or less. Learn more.	Good - Nothing to do here 3ms
Largest Contentful Paint How long it takes for the largest element of content (i.e., a hero image) to be painted on your page. A good user experience is 1.2s or less. Learn more.	Good - Nothing to do here 842ms	Cumulative Layout Shift How much your page's layout shifts as it loads. A good user experience is a score of 0.1 or less. Learn more.	OK, but consider improvement 0.14

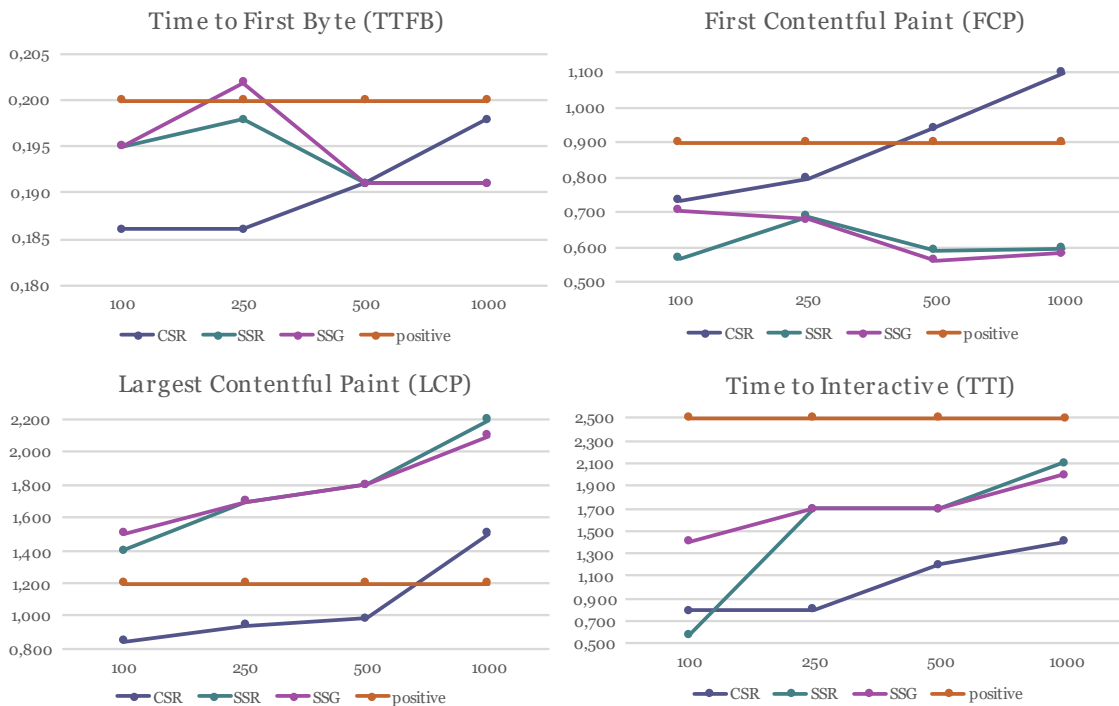
Збір даних



Збір даних

Metric/components	100	250	500	1000	positive
CSR					
Time to First Byte (TTFB)	0,186	0,186	0,191	0,198	0,20
First Contentful Paint (FCP)	0,734	0,797	0,943	1,100	0,90
Largest Contentful Paint (LCP)	0,842	0,938	0,981	1,500	1,20
Time to Interactive (TTI)	0,791	0,797	1,200	1,400	2,50
SSR					
Time to First Byte (TTFB)	0,195	0,198	0,191	0,191	0,20
First Contentful Paint (FCP)	0,568	0,690	0,593	0,597	0,90
Largest Contentful Paint (LCP)	1,400	1,700	1,800	2,200	1,20
Time to Interactive (TTI)	0,568	1,700	1,700	2,100	2,50
SSG					
Time to First Byte (TTFB)	0,195	0,202	0,191	0,191	0,20
First Contentful Paint (FCP)	0,705	0,681	0,562	0,584	0,90
Largest Contentful Paint (LCP)	1,500	1,700	1,800	2,100	1,20
Time to Interactive (TTI)	1,400	1,700	1,700	2,000	2,50

Аналіз даних



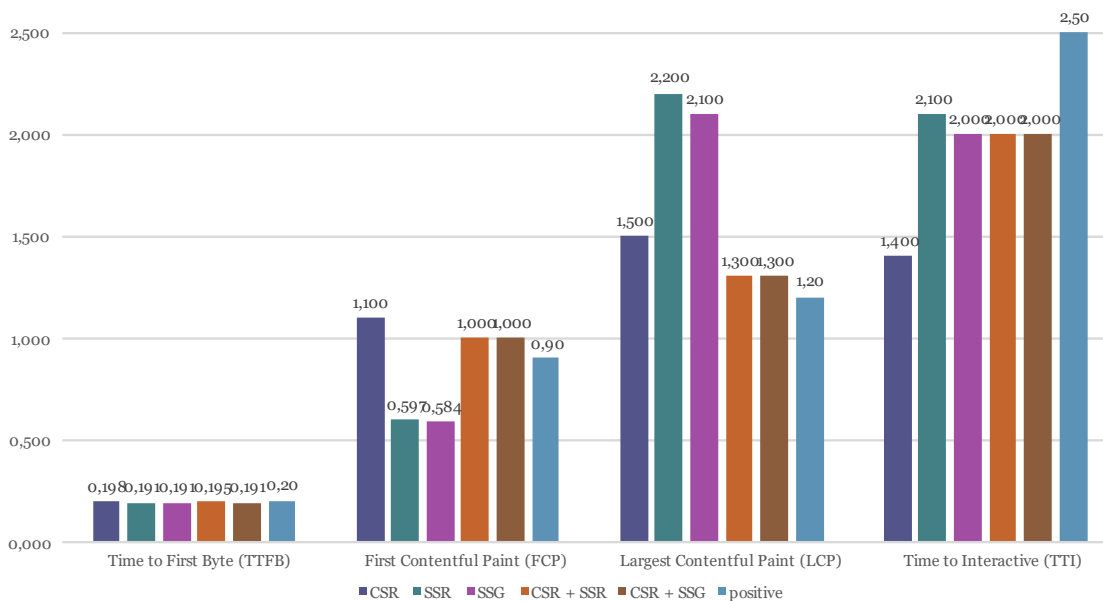
Гібридний рендеринг

```
function App() {
  return (
    <div className="app">
      <h1>Web page to test SSR + CSR in SEO</h1>
      <div className="modified-components">
        <CSRComponentsBlock componentsQty={500} />
        <SSRComponentsBlock componentsQty={500} />
      </div>
    </div>
  );
}
```

Metrix/rendering type	CSR	SSR	SSG	CSR + SSR	CSR + SSG	positive
Time to First Byte (TTFB)	0,198	0,191	0,191	0,195	0,191	0,20
First Contentful Paint (FCP)	1,100	0,597	0,584	1,000	1,000	0,90
Largest Contentful Paint (LCP)	1,500	2,200	2,100	1,300	1,300	1,20
Time to Interactive (TTI)	1,400	2,100	2,000	2,000	2,000	2,50

Гібридний рендеринг

Порівняння метрик різних методів рендерингу



Висновки

- SSR і SSG забезпечують кращу індексацію та швидкість завантаження, тоді як CSR підходить для динамічних, інтерактивних додатків.
- Найкращі результати для індексації показав SSG, оскільки він генерує статичні сторінки, легко доступні для пошукових систем.
- Проведені експерименти підтвердили, що для великих додатків найкраще підходить комбінований підхід, поєднуючи переваги SSR і CSR.
- Використання гібридних методів рендерингу допомагає досягти балансу між швидкістю, інтерактивністю та SEO.
- Застосування статичного рендерингу для контенту що рідко змінюється, та динамічного рендерингу для інтерактивних елементів.

ДОДАТОК Г

Апробація результатів роботи

Брухтій С.С., Ревенчук І.А. Тенденція розвитку методів рендеренгу web-сайтів для пошукової оптимізації сайтів (SEO): матеріали 28-го Міжнарод. молодіжного форуму «Радіоелектроніка та молодь у XXI столітті». Зб. матеріалів форуму. Т. 6., Харків: ХНУРЕ. 2024.С.918-919. (DOI: <https://doi.org/10.30837/IYF.IIS.2024.918>)

ДОДАТОК Д

Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення вимогам ДСТУ 3008: 2015

ДОДАТОК Е

Результати тестування метрик на електронному ресурсі GTmetrix

CSR 100: <https://gtmetrix.com/reports/brukhtii-csr-100.netlify.app/lfBQpKk3/>

CSR 250: <https://gtmetrix.com/reports/brukhtii-csr-250.netlify.app/qeNfWZni/>

CSR 500: <https://gtmetrix.com/reports/brukhtii-csr-500.netlify.app/ge7SJqSr/>

CSR 1000: <https://gtmetrix.com/reports/brukhtii-csr-1000.netlify.app/LQY5VGaV/>

SSR 100: <https://gtmetrix.com/reports/brukhtii-ssr-100.netlify.app/AKe84F0Z/>

SSR 250: <https://gtmetrix.com/reports/brukhtii-ssr-250.netlify.app/GkshoGw0/>

SSR 500: <https://gtmetrix.com/reports/brukhtii-ssr-500.netlify.app/OtJjZL3Q/>

SSR 1000: <https://gtmetrix.com/reports/brukhtii-ssr-1000.netlify.app/AIYzI0Oh/>

SSG 100: <https://gtmetrix.com/reports/brukhtii-ssg-100.netlify.app/FZOx94oq/>

SSG 250: <https://gtmetrix.com/reports/brukhtii-ssg-250.netlify.app/1hqoZQXg/>

SSG 500: <https://gtmetrix.com/reports/brukhtii-500-ssg.netlify.app/TBsWWLZe/>

SSG 1000: <https://gtmetrix.com/reports/brukhtii-ssg-1000.netlify.app/UfwLNCyH/>

Комбіновані рендеринги:

CSR + SSR: <https://gtmetrix.com/reports/brukhtii-ssr-csr-1000.netlify.app/O0Q8MVia/>

CSR + SSG <https://gtmetrix.com/reports/brukhtii-ssg-csr-1000.netlify.app/6p9LzdRk/>

Репозиторії:

CSR: <https://github.com/Brukhtii/csr>

SSR: <https://github.com/Brukhtii/ssr>

SSG: <https://github.com/Brukhtii/ssg>

Node.js: <https://github.com/Brukhtii/node-js-ssr-ssg>

ДОДАТОК Ж

Порівняння метрик різних методів рендиригу

