

УДК 004.92:004.514

ДОСЛІДЖЕННЯ МЕТОДІВ РЕНДЕРІНГУ КОРИСТУВАЦЬКОГО ІНТЕРФЕЙСУ НА БАЗІ ФРЕЙМВОРКІВ REACT ТА SVELTE

Свиридов В. Е.

Науковий керівник – д.т.н., проф. Четвериков Г. Г.

Харківський національний університет радіоелектроніки, каф. ПІ
м. Харків, Україна

e-mail: vadym.svyrydov@nure.ua

In the era of rapid development of web technologies, the React and Svelte frameworks emerge not just as tools, but as key fundamental components of modern programming. Their role in web application development is defined not only by technical capabilities but also by a profound impact on the web industry ecosystem. These frameworks pave the way for creating innovative and dynamic interfaces, providing developers with advanced tools for implementing even the most complex projects. In this context, the significance of React and Svelte lies not only in providing convenient tools but also in transforming the approach to web development itself, which becomes a crucial factor in creating efficient, user-friendly, and scalable web applications.

Дане дослідження побудоване на глибокому аналізі технічних параметрів методів рендерінгу, які надають фреймворки React та Svelte. В рамках цього аналізу ретельно вивчаються ключові аспекти, такі як продуктивність, зручність використання, реакція на зміни, а також інші технічні характеристики, що визначають ефективність роботи інтерфейсів, розроблених з використанням цих фреймворків. Враховуючи широкий спектр технічних показників, ми маємо можливість висвітлити якісні та кількісні аспекти функціонування систем, побудованих з використанням React та Svelte, що дозволить зробити об'єктивне порівняння їх ефективності та придатності для використання у практичних веб-розробках.

Огляд існуючих альтернативних рішень

Рендерінг на стороні клієнта (Client-Side Rendering, CSR) передбачає, що весь HTML та JavaScript код генерується на боці клієнта, тобто в браузері. Цей підхід забезпечує швидку ініціалізацію, адже користувач швидко бачить сторінку. В той же час, він може мати проблеми з оптимізацією для SEO і затримками у відображенні через потребу в більшому обсязі даних для завантаження. Крім того, він вимагає підтримки JavaScript, інакше сторінка може відображатися некоректно або взагалі не відображатися.

Рендерінг на стороні сервера (Server-Side Rendering, SSR) означає, що HTML код генерується на сервері і відправляється клієнту як готова сторінка. Цей підхід полегшує SEO оптимізацію, оскільки сторінка

відправляється зі всім контентом на стороні сервера, і забезпечує швидкий перший рендер. Проте він вимагає більше обчислювальних ресурсів на сервері та обмежує використання клієнтських технологій.

Інкрементальний рендерінг (Incremental Rendering) – це підхід, при якому рендериться лише та частина сторінки, яка змінилася, замість повного перерендерингу всього документа. Це допомагає поліпшити продуктивність, оскільки оновлюється лише та частина сторінки, яка потребує змін, та зменшує навантаження на браузер. Але впровадження такого підходу може бути складнішим і вимагати більшого рівня досвіду та контролю над станом додатка.

React

Бібліотека React[1], створена компанією Facebook, стрімко заволоділа світом веб-розробки. Зараз це лідер у галузі веб-розробки по всьому світу. Росту React сприяє активна спільнота, що обговорює та розвиває його. Три ключові особливості React роблять його особливо привабливим: декларативний характер, компонентна структура додатків та простота інтеграції з існуючими технологічними стеками. Декларативний характер React дозволяє створювати інтерактивні інтерфейси з легкістю. Просто проектуючи візуальні елементи для кожного стану додатка, React бере на себе решту, спрощуючи читання коду та відлагодження. З використанням технології віртуального DOM, React визначає, які частини DOM потрібно оновити на проміжному рівні між кодом розробника та реальним DOM, що забезпечує високу швидкість рендерингу сторінок. React розроблений так, щоб бути нейтральним до технологій у стеку інструментів розробника, що дозволяє інтегрувати React-фронтенд з різними бекендами, такими як Node.js, Ruby on Rails, Spring Boot, PHP та інші.

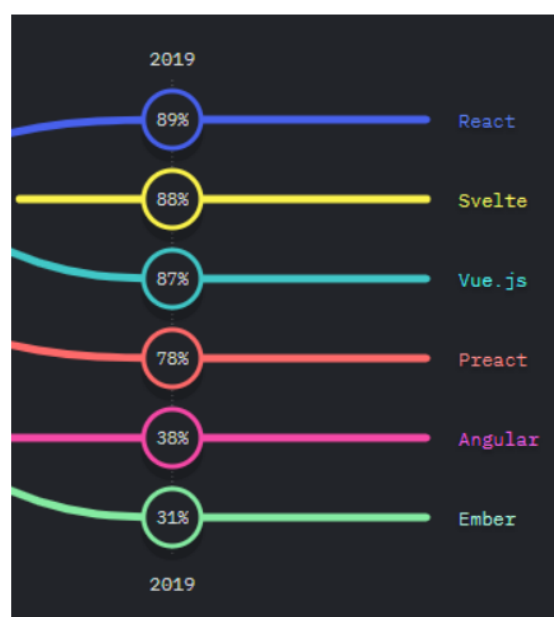


Рисунок 1 – Рейтинг Фронт-Енд фреймворків [1]

Svelte

Svelte – це інструмент, спрямований на революцію у парадигмі розробки фронтенду[2]. Створений Річем Харрісом, цей інноваційний підхід до створення інтерфейсів активно обговорюється в професійній спільноті. Якщо ви ще не переглянули відео, де Річ Харріс розповідає про Svelte 3, ми настійно рекомендуємо це зробити. Порівняно з React, який забезпечує актуальний стан інтерфейсу та виконує інші дії безпосередньо в браузері, Svelte працює під час компіляції проекту. Основна відмінність між React та Svelte полягає в тому, що Svelte під час збірки проекту перетворює ваш додаток в оптимізований JavaScript-код, а не інтерпретує код додатка під час його роботи. Крім того, Svelte не використовує техніки порівняння реального та віртуального DOM. Компілятор Svelte бере декларативний компонент і перетворює його в ефективний імперативний низькорівневий код, який безпосередньо взаємодіє з DOM. Це дозволяє зменшити навантаження на браузер, покращити швидкість роботи сторінок та сприяти оптимізації роботи JS-двигка браузера

Сильні сторони React

- Активна та підтримувана спільнотою платформа.
- Обширний набір інструментів для тестування коду, орієнтованого на React.

- Інтеграція з TypeScript.

- Якісні засоби розробки.

- Обилина корисних компонентів, створених спільнотою.

- Багато вакансій для розробників React.

Сильні сторони Svelte

- Висока продуктивність, що підвищує користувацький досвід на сайтах, побудованих на Svelte.

- Оснований на веб-стандартах (HTML, CSS, JS).

- Відсутність використання JSX.

- Компактний код.

- Обсяг коду Svelte-дodatка у середньому на 40% менший, ніж у аналогічного React-дodatка.

- Оптимізація збірки під час компіляції, що призводить до мінімального розміру кінцевого коду.

- Легка інтеграція бібліотеки в існуючі проекти.

Слабкі сторони React

- Використання віртуальної DOM, що може знижувати продуктивність.

- Додаткове навантаження на систему, що впливає на продуктивність.

- Часті зміни в бібліотеці.

- Недостатня своєчасність оновлення документації через швидкий темп розробки.

- JSX-синтаксис може бути складним для розуміння.

- Обмежене (у порівнянні з React) спільнота користувачів.
- Відсутність вбудованої підтримки TypeScript.
- Обмежена кількість розширень для редакторів коду, засобів підсвічування синтаксису, наборів компонентів, інструментів розробки.
- Маленька кількість вакансій, що потребують знання Svelte (наразі так, але ситуація може змінитися).

Отже, вибір між React і Svelte залежить від конкретних потреб проекту[4], вимог щодо продуктивності та вподобань розробника. Розглядаючи їхні переваги та недоліки, можна зробити висновок, що обидва фреймворки мають свої унікальні переваги та можуть бути використані для успішної розробки веб-додатків з високоякісним користувацьким інтерфейсом.

Список використаних джерел:

1. 2019 State of JS: Front-End Framework [<https://2019.stateofjs.com/front-end-frameworks/>].
2. Документація React [<https://reactjs.org/docs/getting-started.html>].
3. Документація Svelte [<https://svelte.dev/docs>].
4. Рязанцев І.О, Кириченко І.В. Розробка веб-системи для створення інтернетмагазинів, Харків, ХНУРЕ, Радіоелектроніка та молодь у XXI столітті. Т.6. 2021. [<https://nure.ua/wp-content/uploads/2021/R&M/konferencija-6.pdf>].