

## ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ЕЛЕКТРОННОЇ КОМЕРЦІЇ

*Т.В. Панфілов<sup>1</sup>, О.М. Нікуліна<sup>2</sup>*

<sup>1</sup> *магістрант кафедри інформаційних систем та технологій, НТУ «ХПІ», Харків, Україна*

<sup>2</sup> *завідувачка кафедри інформаційних систем та технологій, д-р. техн. наук, НТУ «ХПІ», Харків, Україна  
[tpanfilov.dev@gmail.com](mailto:tpanfilov.dev@gmail.com)*

Сучасний світ важко уявити без інтернету. Web-застосунки мають величезний вплив на життя людей по всьому світу. Вони охоплюють безліч аспектів сучасного життя, починаючи з онлайн-комунікацій та соціальних мереж, які об'єднують мільйони людей, і закінчуючи електронною комерцією, яка робить покупки доступними у будь-якому куточку світу. Майже кожна людина сьогодні має доступ до комп'ютеру, ноутбуку чи смартфона. Кожен такий пристрій має свої технічні характеристики і різну якість інтернет-підключення, тож перед розробниками стає не проста задача якнайкраще оптимізувати швидкість завантаження і роботи додатків для якнайбільшої кількості користувачів. Особливо це стосується застосунків електронної комерції.

Характеристики швидкості роботи і завантаження сайту напряму впливають на алгоритми пошуку у web-браузерах і відповідно мають великий вплив для бізнесу. Окрім того, дослідження, що були проведені Aberdeen Group показали, як затримка при завантаженні сайту у три секунди може викликати втрату до 40% потенційних клієнтів. Саме тому актуальність теми оптимізації завантаження та швидкості роботи додатку важко недооцінити.

Метою роботи є підвищення ефективності завантаження web-додатку на прикладі проектування та створення інформаційної системи електронної комерції.

Для досягнення мети досліджені технології оптимізації швидкості завантаження додатку та застосувати їх на практиці. Додаток створено за принципами клієнт-серверної архітектури, де існує клієнтська частина, або інтерфейс користувача і серверна, що відповідає за обробку запитів і даних. Для написання інтерфейсу web-додатку було застосовано мову Java Script та бібліотеку ReactJS. З боку серверної частини було застосовано мову Java Script на базі платформи NodeJS з використанням технології GraphQL.

Оптимізація завантаження web-застосунку грає велику роль. Основними метриками, які визначають ефективність завантаження та продуктивність веб-застосунку, є First Contentful Paint, Largest Contentful Paint, Time to Interactive, Total Blocking Time, Cumulative Layout Shift та Speed Index. Важливість цих метрик полягає в тому, що вони допомагають розробникам визначати та покращувати ключові аспекти продуктивності, що безпосередньо впливають на користувацький досвід та загальну ефективність веб-застосунку. Окрім того треба звертати увагу на розміри файлів додатку. Для оптимізації швидкості завантаження сторінки було задіяно наступне:

– розділення коду (code splitting) – дозволяє розділяти код веб-застосунку на менші частини (бандли), завантажуючи їх лише тоді, коли це потрібно, що зменшує початковий розмір основного бандлу та прискорює початкове завантаження;

– кешування вендорів (vendors caching) – це сторонні бібліотеки та фреймворки (React, jQuery тощо), які змінюються значно рідше, ніж основний код, ідея полягає в тому, щоб зберігати ці файли у кеші браузера на тривалий час;

– ледаче завантаження (lazy-loading) – це техніка, яка дозволяє завантажувати лише ті компоненти або зображення, які потрібні користувачу на поточний момент.

– мініфікація (minification) – це процес видалення непотрібних пробілів, коментарів і скорочення імен змінних у файлах CSS, JavaScript та HTML, що зменшує розмір файлів, які передаються до клієнта;

– видалення мертвого коду (tree-shaking) – видаляє з бандлу код, який не використовується, це важливо для JavaScript, оскільки вендорські бібліотеки можуть містити багато невикористаних частин;

– оптимізація зображень та використання форматів – зображення це один із найбільших обсягів даних, які завантажуються користуваче, оптимізація розміру, конвертація в ефективні формати (WebP/AVIF) і використання responsive images (srcset) можуть суттєво покращити швидкість завантаження.

Суттєво підвищує швидкість завантаження і роботи додатку – використання CDN (дозволяє отримувати статичні файли з найближчих до користувача серверів, що знижує затримки), проте через матеріальні і технічні обмеження його використання було неможливим.

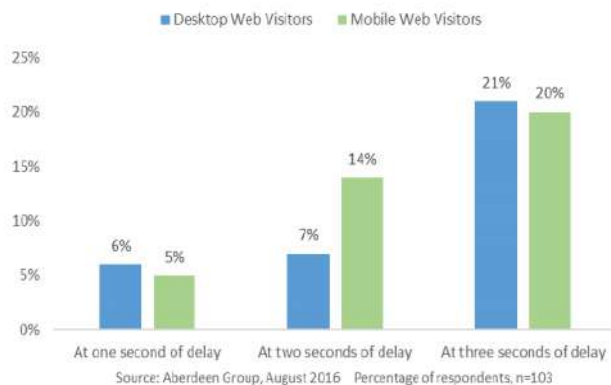


Рис. 1 – Вплив швидкості завантаження застосунку на втрату потенційних клієнтів.

Окрім використання даних технологій для оптимізації інтерфейсу користувача варто зазначити використання технології GraphQL. GraphQL це мова запитів для API та середовище виконання цих запитів. Що дозволяє клієнтам отримувати лише ті дані, які їм потрібні, у структурованому вигляді. На відміну від традиційних REST API, де дані отримуються з фіксованих кінцевих точок, GraphQL дає можливість запитувати конкретні поля для різних ресурсів в одному запиті. Це підвищує ефективність роботи з API, зменшуючи об'єм переданих даних та кількість запитів.

Таким чином було побудовано інформаційну систему електронної комерції, що містить у собі набір мінімальних необхідних функцій із використанням сучасних технологій оптимізації.

#### Список літератури:

1. Casimir, S. Client-Server Web Apps with JavaScript and Java / S. Casimir // O'Reilly Media. – 2014. – 257 p.
2. Jeremy, L. Wagner, Web Performance in Action / L. Jeremy // Manning Publications – 2016. – 376.
3. Офіційна документація React [Електрон. ресурс]. – Режим доступу: <https://react.dev/> – React The library for web and native user interfaces.