

СИСТЕМА НА БАЗІ LARAVEL, ОПТИМІЗОВАНА ЗА ДОПОМОГОЮ FRONT END ТЕХНОЛОГІЙ

З приходом потужних комп'ютерів необхідність у зручності користувацького інтерфейсу зростає, оскільки зростає конкуренція у сфері інформаційних технологій. Невпинно відбувається розвиток інтерактивності та спрощення веб систем для користування ними. Саме це стало причиною створення нових методів оптимізації, розробки зручного та інформативного дизайну систем. Не менш важливим є забезпечення інтуїтивно зрозумілих інтерфейсів, що дає змогу використовувати систему більш ефективно та знижувати поріг для початку взаємодії користувача з системою.

З кожним роком все більше розробників створюють складні системи, які існують у веб середовищі та доступні віддалено користувачам. Створюються технології, які дозволяють розробити веб сервіси та системи якомога швидше, забезпечуючи надійність системи та зручність для користувача. Окремо необхідно зазначити ріст популярності мобільних пристроїв та їх використання для доступу в мережу інтернет. Зважаючи на це, розробники приділяють увагу проектуванню веб систем, орієнтованих також на мобільні пристрої, такі як телефони та планшети.

Для створення веб систем прийнято розподіляти розробку на Front End – розробку користувацького інтерфейсу та Back End – розробку серверної частини. Для взаємодії цих двох частин використовується спеціальний інтерфейс, який, незалежно від технологій і мов, на яких написані сервер та клієнтський інтерфейс, дозволяє передавати дані з клієнтської частини до серверної та навпаки. Найбільш популярними прикладами даних «мостів» є технології JSON та XML. Перша технологія є текстовим рядком, який містить необхідні дані. У другому випадку – це багаторівнева розмітка, яка схожа на розмітку html [1].

Не менш важливим фактором для створення зручного інтерфейсу є час очікування даних, з якими взаємодіє користувач. Чим менше час очікування, тим більш ефективною є система та її зручність для користування.

Нажаль, неможливо при роботі з веб системою зовсім прибрати затримку між взаємодією клієнтом та сервером, але значно зменшити час виконання запиту можливо та потрібно. Для цього необхідно, наскільки це є можливим, перенести функціонал системи на клієнтську частину, з метою

зменшення кількості запитів до сервера, та забезпечити відсутність повторних перезавантажень там, де вони не є необхідними.

Лідером з розробки клієнтських частин в мова програмування JavaScript. Довгий час вона була у версії ES5, що не давало змогу популяризувати об'єктно-орієнтовне програмування на JavaScript, оскільки версія ES5 не підтримує зручне використання класів та створення об'єктів, як в інших мовах програмування. Після переходу на нову версію ES6, у JavaScript з'явилась можливість створювати класи та об'єкти класичним способом, як в Java, PHP, Python та інших класичних мовах програмування. Перехід на нову версію JS надав можливість створювати бібліотеки та фреймворки для розробки клієнтських частин та популяризувати мову JavaScript для розробки веб систем.

Незважаючи на те, що постійно створюються нові бібліотеки та фреймворки для розробки інтерфейсів користувача, найбільш популярними технологіями для розробки клієнтських інтерфейсів є бібліотека React, фреймворк Angular, бібліотека VueJS.

Бібліотека React розроблена компанією Facebook і впевнено займає лідируюче положення у світі інформаційних технологій. Дана бібліотека має невеликий розмір, використовує віртуальний DOM для розробки системи, що дозволяє оптимізувати систему та підвищити швидкість її роботи [2].

VueJS, як і React використовує віртуальний DOM, представляє реактивність та компонентну структуру, фокусується на кореневій бібліотеці, в той час, як інші функції такі як маршрутизація і управління глобальним станом належать до інших бібліотек. В React не використовуються HTML теги для написання коду але, для зручності розробки, React має свій JSX синтаксис, який допомагає полегшити візуальне розділення коду на елементи структури та функціональні можливості. VueJS дає вибір у використанні JSX синтаксису або HTML тегів. VueJS та React для написання повноцінних веб систем потребують додаткових технологій, такі як Flux або Redux, які допомагають управляти станом даних [3].

Аналогом описаних бібліотек для написання клієнтської частини веб системи є фреймворк Angular. Даний фреймворк розроблений компанією Google та постійно має велику спільноту. Angular, починаючи з другої версії, є основним конкурентом React та VueJS. Даний фреймворк використовує мову Type Script, що в свою чергу, підвищує поріг для вступу у розробку на даному фреймворку. Використання Type Script та робота з реальним DOM є однією із ключових відмінностей даного фреймворку від попередньо розглянутих бібліотек [3].

Для оптимізації системи, розробленої на Laravel, необхідно;

- забезпечити зменшення часу очікування при виконанні типових задач, які не потребують перезавантаження сторінки;

- оптимізувати перше завантаження сторінки браузером користувача;
- забезпечити розподілення клієнтської частини (Front End) та серверної (Back End);
- оптимізувати клієнтську частину, яка використовує DOM дерево;
- підвищити зручність використання системи клієнтом.

За основу беремо торговельну систему, яка розроблена на фреймворку Laravel, з використанням шаблону проектування MVC (модель, вид, контролер). Дана система має клієнтську частину, яка складається з файлів шаблонізатору BLADE. При переході на сторінки, фреймворк створює запит до сервера, який крім необхідних даних надає розмітку сторінки, що повторює попередню завантажену розмітку HTML. Крім того при виконанні дій, таких як створення замовлення, додавання товару до кошику знову відбувається запит до сервера, який повторює попередню розмітку. Дана система використовує бібліотеку JQuery, яка працює з DOM деревом, тим самим збільшує час очікування роботи системи.

Для вирішення поставлених завдань обрана бібліотека React, за допомогою якої розроблено новий інтерфейс користувача, що забезпечує відсутність перезавантажень сторінок при переході по сайту. Також частина важливих функцій, які виконувалися фреймворком Laravel, виконується за допомогою React. Торговельна система використовує віртуальний DOM для фонового виконання задач перед тим, як вивести результати на сторінку сайту. Бібліотека JQuery вилучена, а функціонал за який вона відповідала, розроблено на Javascript ES6. Для забезпечення підтримки старішими браузерами клієнтської частини, ES6 переведено у попередню версію ES5 та мінімізовано для більш швидкого завантаження.

Оптимізація торговельної системи, розробленої на Laravel, за допомогою сучасних Front End технологій не торкнулася функціональних можливостей самої системи та її цілісності. Торговельна система так само виконує поставлені перед нею задачі, використовує шаблон проектування MVC, Dependency Injection, Fasad та має ті самі маршрути, що використовувалися системою до оптимізації. Тим паче, оптимізація системи принесла додатковий шаблон проектування MVVM, забезпечила більш зручний інтерфейс та знизила навантаження на сервер.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Алексеев, Г.В. Современные методы разработки компьютерных систем / Г.В. Алексеев. - СПб.: ГИОРД, 756. - 2012 с.
2. Ершов А.В. Интерактивные системы // А.В. Ершов, Ю.В. Гуляев, Н.А. Карпов, А.Ю. Пахомов. – М.: ЛИПС, 2014. – 596 с.
3. Сравнение с другими фреймворками. [Электронный ресурс]. – Режим доступа: <https://ru.vuejs.org/v2/guide/comparison.html>