

## **ОПТИМІЗАЦІЯ ВЕБ-ДОДАТКІВ З ВИКОРИСТАННЯМ АСИНХРОННОГО ПРОГРАМУВАННЯ В JAVASCRIPT**

У сучасному світі люди не уявляють свого життя без гаджетів. Щодня з'являються все нові і нові додатки та веб-ресурси, мета яких зробити життя людей простіше. Однією з основних вимог користувачів до веб-додатків є швидкість роботи, зручність інтерфейсу та зрозумілий інтерфейс. У світі технологій постійно з'являються нові засоби та методи для оновлення вже існуючих веб-додатків та створення нових. ІТ спеціалісти спрямовують свої дослідження на пошук технологій, що значно покращать продуктивність веб-додатків та зможуть урізноманітнити їх функціонал.

На даному етапі розвитку веб-технологій основними мовами програмування є JavaScript, який використовується як на клієнтській, так і на серверній стороні завдяки платформі Node.js. JavaScript є основою фронтенд-розробки і використовується для створення і відображення веб-сторінок та взаємодії з користувачами. Для розробки фронтенду існують різні фреймворки та бібліотеки, такі як React, Angular і Vue.js. React, наприклад, спеціалізується на створенні користувацьких інтерфейсів, Angular надає структурований підхід до фронтенд-розробки, а Vue.js відомий своєю легкістю вивчення та використання. На бекенді для створення серверної логіки та обробки запитів використовуються різні мови програмування. Node.js, який базується на JavaScript, дозволяє розробляти серверні додатки. Крім того, інші мови, такі як Python, Ruby, PHP і Java, також використовуються для бекенд-розробки незалежності від вимог проекту. Фронтенд відповідає за створення користувацького інтерфейсу, відображення даних та взаємодію з користувачами, в той час як бекенд забезпечує обробку серверних запитів, виконання бізнес-логіки та взаємодію з базами даних, забезпечуючи бекграунд-функціональність веб-додатка.

Створення веб-додатків на JavaScript – це складний та багатоетапний процес, що починається зі збору функціональних вимог та розробки дизайну інтерфейсу. Після цього розробники створюють структуру додатку, включаючи HTML-сторінки для відображення контенту та JavaScript-код для обробки взаємодії з користувачем. Для покращення продуктивності та реактивності веб-додатків, розробники використовують асинхронне програмування.

Однією з ключових стратегій є використання Promises для управління асинхронними операціями та їх обробки. Асинхронні функції (`async/await`) дозволяють писати код, який виглядає як синхронний, але виконується асинхронно, спрощуючи розуміння та підтримку коду.

Callbacks були популярним способом управління асинхронним кодом до появи Promises та `async/await`. Promises пропонують більш структурований спосіб керування асинхронним кодом. `Async/await` надає більше читабельності, роблячи код подібним до синхронного стилю.

Кешування даних є важливою стратегією оптимізації, оскільки дозволяє зберігати результати попередніх запитів на клієнтському боці, зменшуючи кількість необхідних запитів до сервера та покращуючи відгук додатку. Мінімізація запитів до сервера включає в себе зменшення кількості запитів та оптимізацію їх структури, щоб зменшити навантаження сервера та покращити швидкість завантаження сторінок. Паралельні асинхронні запити, такі як використання AJAX або Fetch API, дозволяють взаємодіяти з сервером одночасно, покращуючи реактивність додатку та знижуючи час очікування результатів. Керування завантаженням ресурсів на вимогу (`lazy loading`) – це важлива стратегія для зменшення часу завантаження сторінок, дозволяючи завантажувати ресурси лише при необхідності.

Ще одним важливим аспектом оптимізації є робота над швидкістю виконання JavaScript-коду. Відсутність зайвих обчислень та надмірної вкладеності може покращити продуктивність додатку. Використання інструментів профілювання дозволяє ідентифікувати та усунювати вузькі місця в коді, забезпечуючи більш ефективну роботу додатку. Оптимізація також включає в себе мінімізацію використання ресурсів браузера, таких як пам'ять та процесор, що допомагає забезпечити стабільну та високошвидкісну роботу на різних пристроях та браузерах.

Загалом, оптимізація веб-додатків з використанням асинхронного програмування включає в себе комплексний підхід, спрямований на покращення продуктивності, реактивності та користувацького досвіду, що стає важливим в умовах сучасного веб-середовища.

### **Список використаних джерел**

1. Асинхронне програмування: що це таке та його особливості. 2023. URL: <https://foxminded.ua/asynkhronne-prohramuvannia/>