

NEXT.JS ПРОТИ ANGULAR: ПРАКТИЧНЕ ПОРІВНЯННЯ НА ОСНОВІ РОЗРОБКИ ВЕБ-ДОДАТКІВ

Кириченко В.В., Жученко Т.В., науковий керівник Кириченко В.В.

Перед початком створення застосунку у розробника завжди постає питання: «Яку технологію обрати?». Якщо йдеться про веб-застосунки, то тут маємо досить багато варіантів, але є два лідери: Next.js та Angular. Тож необхідно вирішити якому із фреймворків надати перевагу. Порівняємо Next.js (версія 14.2.3) та Angular (версія 18.1.1) на основі практичного досвіду розробки ідентичних веб-додатків[1, 2] із використанням обох фреймворків. Кожен додаток складався з двох сторінок: форми для створення токенів та списку створених токенів.

Почнемо з порівняння досвіду розробки отриманому під час написання веб-застосунків. Angular, розроблений та підтримуваний Google, надає комплексний фреймворк з чітко визначеною структурою. При розробці додатку для створення токенів, сильна типізація через TypeScript та система впровадження залежностей Angular забезпечували надійну архітектуру. Однак, це супроводжувалося помітною складністю, а саме:

- Налаштування проекту вимагало значно більше шаблонного коду;
- Компоненти потребували більш багатослівних визначень з декораторами та метаданими;
- Навчання було складнішим, вимагаючи розуміння специфічних для Angular концепцій, таких як модулі, сервіси та впровадження залежностей.

Система компонентів Angular, хоч і потужна, вимагала більше коду для реалізації повторно використовуваних елементів у додатку. Процес розробки був більш структурованим, але менш гнучким та інтуїтивним.

Next.js, побудований на React і розроблений Vercel, забезпечив помітно інший досвід розробки:

Ініціалізація проекту була простою з мінімальною конфігурацією;

Файлова система маршрутизації усунула потребу в складних визначеннях маршрутів;

Компонентна модель React зробила створення та повторне використання компонентів більш інтуїтивним;

Середовище розробки пропонувало швидший зворотний зв'язок з гарячим перезавантаженням.

Створення повторно використовуваних компонентів для сторінок форми та списку токенів було значно простішим у Next.js. Простота фреймворку дозволила швидші ітерації розробки та легше вирішення задач.

Досвід розгортання виявив одну з найбільш значущих відмінностей між фреймворками. Додаток Next.js розгортався швидко і зручно на Vercel з мінімальною конфігурацією, так як всі процеси автоматизовано, потрібно тільки підключити репозиторій[4] з кодом до Vercel і посилання на веб-застосунок готове за кілька секунд.

Angular вимагав більше кроків та налаштувань для розгортання, хоча і був розгорнутий на Firebase – платформі Google.

Інтеграція Vercel з Next.js забезпечила додаткові оптимізації продуктивності, можливість розгортання на платформі Vercel представляла значну перевагу для Next.js.

Розглянемо порівняння показників швидкодії веб-застосунків, які наведено у таблиці нижче(табл. 1).

Метрика	Angular	Next.js	Різниця
Початковий розмір бандлу	301.64 кБ	89.6 кБ	На 70,3% менше з Next.js
Час першого завантаження	0,41с	0,3с	На 26,8% швидше з Next.js

Метрика	Angular	Next.js	Різниця
Час до інтерактивності	10мс	0мс	На 100% швидше з Next.js
Час збірки	45с	18с	На 60% швидше з Next.js
Рядки коду	870	520	На 40% менше коду з Next.js
Запуск сервера розробки	12с	4с	На 67% швидше з Next.js

Табл. 1. Показники швидкодії

Ці метрики чітко демонструють переваги ефективності Next.js за кількома параметрами продуктивності. Значно менший розмір бандлу перекладається на швидший час завантаження, що безпосередньо впливає на користувацький досвід. Крім того, зменшений час збірки та запуску сервера розробки сприяють більш ефективному процесу розробки.

На основі цього практичного порівняння розробки ідентичних веб-додатків, Next.js демонструє чіткі переваги над Angular у кількох ключових областях. Досвід розробки був більш оптимізованим та інтуїтивним, повторне використання компонентів було простішим у реалізації, розгортання було значно легшим, а навчання швидшим.

Показники продуктивності надають кількісні докази ефективності Next.js, з розмірами бандлів на 70,3% меншими та часом завантаження більш ніж на 25% швидшим, ніж реалізація на Angular. Ці покращення перекладаються на кращий користувацький досвід та більш ефективні робочі процеси розробки.

У той час як Angular пропонує комплексний фреймворк з чітко структурованою архітектурою додатків, Next.js надає більш гнучкий підхід, який прискорює розробку, зберігаючи продуктивність.

Хоча в цьому порівнянні Next.js демонструє переваги, але це стосується невеликих та середніх проектів, важливо зазначити, що Angular має свої сильні сторони, які роблять його кращим вибором для корпоративних проектів. Angular забезпечує суворо регламентовану структуру, що є перевагою для великих команд та корпоративних проектів, де важлива стандартизація коду, пропонує комплексний набір інструментів «із коробки» щоб не додавати сторонні бібліотеки, які можуть застаріти.

Отже, для проектів, які пріоритезують ефективність розробників, швидший вихід на ринок та просте розгортання, Next.js виявляється кращим вибором, а для масштабних корпоративних проектів із комплексною бізнес-логікою краще обрати Angular.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Веб-застосунок написаний на Next.js [Електронний ресурс]. – Режим доступу: <https://token-form.vercel.app>
2. Веб-застосунок написаний на Angular [Електронний ресурс]. – Режим доступу: <https://token-form-angular-spa.web.app>
3. Репозиторій з кодом веб-застосунку Angular [Електронний ресурс]. – Режим доступу: <https://github.com/tetiana-zhuchenko/token-form-angular-public>
4. Репозиторій з кодом веб-застосунку Next.js [Електронний ресурс]. – Режим доступу: <https://github.com/tetiana-zhuchenko/token-form>

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**



ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ АСПЕКТИ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ

**ЗБІРНИК НАУКОВИХ ПРАЦЬ ЗА МАТЕРІАЛАМИ
VII ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ
СТУДЕНТІВ І АСПІРАНТІВ
*24 квітня 2025 року***

Київ 2025

УДК 004

Відповідальна за випуск: М.І. Лендел

Збірник наукових праць за матеріалами VII Всеукраїнської науково-практичної конференції студентів і аспірантів «ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ АСПЕКТИ РОЗРОБКИ КОМП'ЮТЕРНИХ СИСТЕМ '2025», 24 квітня 2025 року, НУБіП України, Київ. – 452 с. (електронне видання)

Відповідальність за зміст публікацій несуть автори.

Передрук матеріалів, а також використання їх будь-якій формі допускається лише з дозволу авторів