

РОЗРОБКА ВЕБ-ДОДАТКУ НА ОСНОВІ ФРЕЙМВОРКІВ REACT TA NESTJS

Вінницький національний технічний університет

Анотація

В роботі розглянуто переваги створення веб-додатків за допомогою фреймворків React і NestJS та наведено приклад створення додатку на основі цих технологій.

Ключові слова: веб-додаток, веб-розробка, створення додатків, JavaScript, React, NestJS.

Abstract

This work examines the advantages of creating web applications using the React and NestJS frameworks and provides an example of creating an application based on these technologies.

Keywords: web application, web development, creating apps, JavaScript, React, NestJS.

Вступ

За останні десятиріччя веб-розробка та створення веб-додатків стали надзвичайно актуальними. У сучасному цифровому світі вони відіграють ключову роль у процесах цифрової трансформації. Веб-додатки значно полегшують повсякденні завдання, пропонуючи ефективні рішення для освіти, роботи та дозвілля.

Із розвитком мови JavaScript з'явилася велика кількість фреймворків для створення веб-додатків, серед яких React став одним із найпопулярніших. Цей фреймворк забезпечує зручний спосіб розробки масштабованих та функціональних інтерфейсів.

React конкурує з багатьма іншими популярними JavaScript фреймворками, такими як Angular, Vue.js, Svelte. [1] Кожен із них має свої сильні сторони, але React вигідно відрізняється у деяких аспектах, а саме у простоті та гнучкості у порівнянні з Angular та кращій у продуктивності в порівнянні з Vue.js.

NestJS є потужним фреймворком для створення серверних додатків на основі Node.js, який також здобуває велику популярність серед розробників завдяки численним перевагам, наприклад, типізації за допомогою Typescript та швидкості розробки через спрощення основних задач. [2]

У комбінації NestJS і React дозволяють створювати потужні, масштабовані та ефективні веб-додатки з високою продуктивністю, безпекою та простотою в підтримці. Поєднання цих технологій забезпечує легкість інтеграції клієнтської та серверної частини, високу продуктивність та підтримку більшості сучасних стандартів та практик розробки програмного забезпечення.

Результати дослідження

В результаті досліджень технологій, що підходять для створення масштабованих, високофункціональних та продуктивних веб-додатків було сформульовано переваги розробки за допомогою React і NestJS в порівнянні з іншими технологіями.

Переваги розробки веб-додатків із використанням фреймворку React [3]:

1. Простота вивчення та використання. Для розробників, які мають базове розуміння HTML та JavaScript, вивчення цього фреймворку буде відносно легким та зрозумілим.

2. Швидкий рендеринг. Фреймворк React включає функцію віртуального DOM-дерева, що діє як спрощене представлення реального DOM-дерева і дозволяє оптимізувати відображення контенту шляхом мінімізації прямих маніпуляцій елементами.

3. Модульність компонентів. React дозволяє створювати веб-додатки за допомогою компонентів, які легко змінювати, оновлювати та повторно використовувати без впливу на інші частини коду.

4. Підтримка спільноти. Розробники мають доступ до великої кількості онлайн-посібників, стартових наборів, форумів і зустрічей, що суттєво полегшують навчання та вирішення проблем.

5. Підходить для великих додатків. Завдяки модульній архітектурі та компонентній структурі фреймворку React розробники отримують можливість створення та підтримки додатків великого об'єму. Фреймворк React дозволяє будувати складні інтерактивні системи, використовуючи стан керування, що також покращує організацію коду в масштабних проектах.

NestJS також має суттєві переваги порівняно з іншими фреймворками, а саме [4]:

1. Масштабованість та впорядкованість. NestJS підтримує модульну архітектуру, що дозволяє розробникам легко організовувати програми в модулі, компоненти, служби та контролери.

2. Вбудована підтримка мікросервісів. У NestJS є зручний пакет, що обробляє найбільшу кількість комунікацій служб і підтримує низку транспортних протоколів, таких як gRPC.

3. Попередня обробка даних та безпека. NestJS пропонує засоби, що дозволяють розробникам перехоплювати вхідні запити та застосовувати логіку попередньої обробки до того, як вони досягнуть контролерів.

4. Підтримка Typescript. На відміну від Node.js, NestJS має вбудовану підтримку TypeScript, що робить розробку додатків простіше та усуває можливі проблеми завдяки типуванню даних.

Розглянемо послідовність процесів розробки веб-додатків з використанням React та NestJS [5]:

1. Налаштування середовища розробки. Для розробки на React та NestJS потрібно налаштувати середовище розробки, встановити менеджер пакетів npm та налаштувати редактор коду.

2. Створення проекту. Після налаштування середовища розробки, нові проекти можна створити за допомогою терміналу або інструментів розробки.

3. Розробка інтерфейсу користувача. Використовуючи інструменти React, розробники можуть побудувати інтерфейс користувача для веб-додатку. React також підтримує численні бібліотеки із готовими компонентами для створення інтерфейсу користувача.

4. Розробка серверної частини. За допомогою командного інтерфейсу розробники можуть генерувати модулі, контролери та сервіси, необхідні для роботи додатків. NestJS також має вбудовану підтримку роботи з базами даних, автентифікацією, валідацією запитів та обробкою помилок, що дозволяє швидко створювати якісні серверні додатки.

5. Управління даними. За допомогою мови JavaScript [6] можна додати логіку додатків, обробку подій та взаємодію із сторонніми сервісами та API, а також інтеграцію із серверною частиною. Для управління даними з клієнтської сторони зручно використовувати React або бібліотеки Redux, Context API.

6. Тестування. React та NestJS підтримують бібліотеки, що надають інструменти для автоматизованого тестування веб-додатків, деякі бібліотеки підходять для React і NestJS одразу.

7. Збірка та розгортання. Після успішного тестування додатку можна зібрати його у мініфіковані файли та розгорнути за допомогою інструментів для веб-хостингу.

8. Оптимізація. Після розробки можливо оптимізувати продуктивність, використовуючи GitHub Pages, Netlify або Vercel.

Розглянемо процес створення веб-додатку за допомогою React і NestJS. Створимо окрему папку для кожного проекту. Для створення React-додатку в терміналі виконаємо команду `prx create-react-app`. Для створення додатку NestJS, перейшовши до іншої папки виконаємо команди `npm i -g @nestjs/cli` та `nest new`. Створені проекти можна запустити за допомогою команди `npm run start`. [7]

Висновки

Фреймворки React і NestJS є ефективними та зручними засобами для розробки сучасних веб-додатків із високою продуктивністю. Завдяки компонентній архітектурі, гнучкості та швидкодії, React часто обирають розробники як оптимальне рішення для створення інтерфейсів.

Використання React у розробці значно спрощує створення веб-додатків, забезпечує їх масштабованість і покращує загальну продуктивність. У свою чергу, NestJS дозволяє швидко налаштувати серверну логіку, що не лише підвищує ефективність, а й розширює функціональні можливості застосунку.

Завдяки активній підтримці спільноти, широкому вибору додаткових бібліотек та переваг у поєднанні технологій React та NestJS залишаються одними із найпопулярніших інструментів у веб-розробці.

Поєднання React із NestJS дозволяє створювати повноцінні веб-додатки з високою ефективністю та зручністю. React відповідає за клієнтську частину, забезпечуючи динамічні та інтерактивні інтерфейси,

тоді як NestJS працює із серверної сторони, забезпечуючи масштабовану, модульну та ефективну серверну логіку. Використання цих двох технологій разом дає змогу створювати потужні та стабільні додатки, які підтримують масштабування, мають зручну архітектуру та високу продуктивність.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. React [Електронний ресурс] – Режим доступу: <https://react.dev/> (дата звернення: 15.05.2025).
2. Beighton, B. NestJS: The Pros and Cons [Електронний ресурс] – Режим доступу: <https://bradbeighton.medium.com/nestjs-the-pros-and-cons-aff714607b07> (дата звернення: 15.05.2025).
3. Порівняння фреймворків JavaScript: Svelte проти React [Електронний ресурс] – Режим доступу: <https://www.dreamhost.com/blog/uk/svelte-proti-react/> (дата звернення: 15.05.2025).
4. NestJS. Офіційна документація [Електронний ресурс] – Режим доступу: <https://docs.nestjs.com/> (дата звернення: 15.05.2025).
5. Srivastava A. Setting Up a Simple API with Nest.js and React.js: A Step-by-Step Guide [Електронний ресурс] – Режим доступу: <https://medium.com/%40aviralsrivastava57/setting-up-a-simple-api-with-nest-js-and-reactjs-a-step-by-step-guide-javascript-30b868548df6> (дата звернення: 15.05.2025).
6. JavaScript Tutorial [Електронний ресурс] – Режим доступу: <https://www.w3schools.com/js/> (дата звернення: 15.05.2025).
7. React and NestJS: Building a Full-Stack Application [Електронний ресурс] – Режим доступу: <https://medium.com/@chauhananubhav16/react-and-nestjs-building-a-full-stack-application-ffea6098769e> (дата звернення: 15.05.2025).

Кривоzubченко Ксенія Денисівна – здобувач вищої освіти групи ІІСТ-21б, кафедра автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: schlkr7@gmail.com.

Богач Ілона Віталіївна – к.т.н., доцент кафедри автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: ilona.bogach@gmail.com.

Kryvohubchenko Kseniia Denysivna – student of IIST-21b group, Department of Automation and Intelligent Information Technologies, Faculty of Computer Systems and Automatics, Vinnytsia National Technical University, Vinnytsia, e-mail: schlkr7@gmail.com.

Bogach Ilona Vitaliivna – Associate Professor of Automation and Intelligent Information Technologies Department, Faculty of Computer Systems and Automatics, Vinnytsia National Technical University, Vinnytsia, e-mail: ilona.bogach@gmail.com.