

УДК 004.4

DOI: <https://doi.org/10.17721/3041-2323.2024.286-291>

Олег ПРОЙДАКОВ, студ.  
ORCID ID: 0009-0003-6603-2101  
e-mail: [oleh.proidakov@gmail.com](mailto:oleh.proidakov@gmail.com)  
Київський національний університет  
імені Тараса Шевченка, Київ, Україна

## АНАЛІЗ І ВИБІР ТЕХНОЛОГІЙ ДЛЯ ВЕБСИСТЕМИ УПРАВЛІННЯ ЦИФРОВОЮ БІБЛІОТЕКОЮ

*Розглянуто вибір технологій для розроблення веборієнтованої системи управління цифровою бібліотекою з використанням сучасних технологій, таких як React, Nest.js, PostgreSQL, wagmi та viem. Система призначена для полегшення запитів на оновлення даних про книги та дозволяє здійснювати платні публікації книг. Детально описано архітектуру та функціональність платформи. Висвітлено деякі можливості платформи, зокрема й інтеграцію з технологією блокчейн для полегшення транзакцій.*

**Ключові слова:** *цифрова бібліотека, web3, криптовалюти, платіжні, React, Nest.js, PostgreSQL, wagmi, viem.*

### Вступ

Сучасні цифрові бібліотеки набули великого поширення як інструменти збереження, управління та розповсюдження знань у цифрову епоху. Основна мета таких систем полягає в тому, щоб зробити доступ до інформації зручнішим та ефективнішим, надаючи високий рівень безпеки даних і можливість інтерактивної взаємодії з користувачами. Нині важливим є питання підтримки актуальності даних у бібліотеці, а також упровадження новітніх технологій, зокрема і криптовалют, для реалізації економічних взаємодій між користувачами й авторами. Особливістю цієї системи є можливість подання заявок на оновлення даних про книги безпосередньо користувачами, що дозволить частково децентралізувати процес оновлення та покращення бібліографічної інформації.

### Результати

Для розроблення системи, яка одночасно має легко масштабуватись і витримувати навантаження, вважаю доцільним обрати

© Пройдаков Олег, 2024

мову TypeScript. Це також обґрунтовно тим, що сучасні браузері підтримують не так багато мов і використання однієї мови і для серверної, і для клієнтської частини пришвидшить процес розроблення. Використання фреймворку NestJS (n. d.) у свою чергу, забезпечить високу модульність і масштабованість, що дозволить легко інтегрувати механізми для оновлення та покращення бібліографічної інформації. Завдяки тому, що NestJS підтримує MVC архітектуру, можна ефективно організувати серверну частину, розділивши її на окремі сервіси для оброблення запитів, автентифікації користувачів та роботи з даними. Це спростить адміністрування системи та підтримку стабільності роботи. MongoDB (n. d.) і PostgreSQL (n. d.) є популярними базами даних. Було б доцільно порівняти їх, щоб визначити, яка БД краще задовольнить потреби вебсистеми цифрової бібліотеки. Для цього я провів збір даних і навів результати у табл. 1.

*Таблиця 1*

**Порівняння MongoDB та PostgreSQL**

<b>Критерій</b>	<b>MongoDB</b>	<b>PostgreSQL</b>
Модель зберігання даних	Документоорієнтована, зберігає дані у вигляді JSON-подібних документів (BSON)	Реляційна (зберігає дані у вигляді таблиць із чіткими зв'язками між даними таблиці, реляції)
Тип бази даних	NoSQL (нереляційна база даних)	SQL (реляційна база даних)
Строгість схеми	Безсхемна (гнучка структура даних, можна легко змінювати модель даних без необхідності міграцій)	Схемна (вимагає чіткої визначеної структури для збереження даних, необхідні міграції за змін схеми)
Підтримка горизонтального масштабування	Так	Так

Закінчення табл. 1

Критерій	MongoDB	PostgreSQL
Цілісність даних і надійність	Менш сувора цілісність даних, дозволяє "гнучкі" відносини між об'єктами	Висока цілісність даних і підтримка референційної цілісності через реляційні обмеження (FOREIGN KEYS)
Швидкість вставки*	~25,340 оп/с	~12,460 оп/с
Розширені можливості індексації	Підтримує індексацію JSON-документів, геопросторову індексацію	Підтримка складних типів індексів (B-Tree, GiST, GIN, BRIN), а також JSONB для швидкого пошуку

\* Швидкість вставки виміряно у віртуальному середовищі з 2 vCPU на контейнер.

Отже, PostgreSQL має кілька переваг, які роблять його кращим вибором для системи управління цифровою бібліотекою. MongoDB – це NoSQL база даних, відома своєю гнучкістю у зберіганні неструктурованих даних і простотою горизонтального масштабування. Однак така структура не завжди підходить для складних запитів і забезпечення цілісності даних. MongoDB може бути корисним для роботи з документами або даними без чіткої структури, але PostgreSQL надає кращу стабільність, продуктивність і надійність для критично важливих застосунків, особливо коли потрібна строгість і точність у роботі з даними.

PostgreSQL, як реляційна база даних, пропонує вищу надійність і відповідність стандарту ACID. Це дозволяє ефективно працювати зі складними транзакціями, підтримувати зв'язки між таблицями, а також виконувати складні аналітичні запити. Завдяки своїй підтримці складних типів даних і розширених індексацій, PostgreSQL забезпечує високу продуктивність при обробленні великих обсягів інформації. Крім того, PostgreSQL має гнучкі можливості масштабування, що дозволяє ефективно використо-

увати його як для невеликих проєктів, так і для великих корпоративних систем.

React (n. d.) разом із Vite забезпечать швидке і зручне розроблення клієнтської частини вебзастосунку. React – потужна бібліотека для створення динамічного інтерфейсу користувача з можливістю створення компонентів, що спрощує підтримку та масштабування застосунку. Vite, у свою чергу, забезпечує миттєве перезавантаження під час розроблення та швидкого збирання проєкту завдяки сучасній архітектурі, що значно пришвидшує процес розроблення, оскільки Vite компілює лише необхідні частини коду під час змін, що робить його ідеальним для великих і складних проєктів. Разом ці технології забезпечують продуктивну роботу як під час розроблення, так і при запуску застосунку в загальний доступ.

Однією з ключових особливостей системи буде можливість публікації книг авторів за криптовалюту. Для інтеграції блокчейн-технологій доцільно використати дві популярні бібліотеки — wagmi та viem, які дозволяють ефективно взаємодіяти з EVM-сумісними мережами. Це дасть авторам змогу публікувати свої роботи без посередників. Така інтеграція створить глобальну платформу, незалежну від традиційних фінансових систем, що робить публікацію контенту доступнішою і ефективнішою. Це забезпечить не лише зручність для авторів, але і стимулюватиме розвиток інновацій у сфері децентралізованих фінансів і контенту.

Сучасне розроблення вебзастосунків значною мірою зосереджується на створенні ефективних, динамічних та естетично привабливих інтерфейсів. З розвитком технологій з'являються нові підходи до розроблення, які дозволяють прискорити роботу над зовнішнім виглядом застосунків і спростити підтримку їхньої гнучкості й адаптивності. Різні інструменти та бібліотеки стають невід'ємною частиною роботи над клієнтською частиною, де особливу увагу приділяють дизайну, анімації та взаємодії користувача із системою. Для створення комплексних інтерфейсів і зручних форм нині існує чимало інструментів, таких як TailwindCSS, NextUI, framer-motion.

Ще однією важливою частиною сучасного вебзастосунку є форми, і для їх оброблення в React найчастіше використовують бібліотеку react-hook-form. Цей інструмент дозволяє легко ство-

рювати та керувати складними формами з валідацією, зводячи до мінімуму кількість коду, необхідного для їх реалізації.

Завдяки оптимізованій роботі, `react-hook-form` зменшує кількість повторних провантажень, що позитивно впливає на продуктивність і робить взаємодію з формами плавною та зручною для користувача. Порівняно з `Formik` (n. d.), який вимагає відносно більше коду і часто призводить до надлишкових рендерів під час оновлення полів, `react-hook-form` є значно ефективнішим рішенням.

Використання бібліотеки `axios` є важливим аспектом у розробленні сучасних вебзастосунків, оскільки вона значно спрощує управління HTTP-запитами. `Axios` (n. d.) забезпечує зручний API для виконання асинхронних запитів, підтримує автоматичне оброблення JSON і дозволяє легко налаштовувати заголовки та обробники помилок. Це дозволяє розробникам швидше реалізувати функціонал, зосередившись на функціоналі застосунку, а не на детальному обробленні запитів.

Спільно з іншими бібліотеками, такими як `SWR`, `axios` може використовуватися для оптимізації процесу отримання даних. `SWR` автоматично кешує результати запитів та оновлює їх у фоновому режимі, що забезпечує швидкий доступ до інформації, зменшуючи час завантаження та покращуючи досвід користувачів. Комбінування цих інструментів дозволяє створити більш оптимізовану вебсистему, здатну швидко реагувати на зміни даних і забезпечувати високу продуктивність, що є критично важливим для успішних вебзастосунків у сучасному середовищі.

#### **Дискусія і висновки**

Під час проведеного дослідження детально проаналізовано широкий спектр сучасних технологій, що можуть бути використані для розроблення вебсистеми управління цифровою бібліотекою. Особливу увагу приділено вибору інструментів, які забезпечують інтеграцію блокчейн-технологій, що дозволяє підвищити безпеку та прозорість оброблення даних. Крім того, обрано бібліотеки, які сприяють поліпшенню досвіду користувачів під час роботи з формами та виконанні HTTP-запитів.

#### **Список використаних джерел**

MongoDB. (n. d.). *MongoDB*. <https://www.mongodb.com/docs/atlas>  
NestJS. (n. d.). *Документація до фреймворку NestJS*. <https://nestjs.com/>

PostgreSQL. (n. d.). *Документація для PostgreSQL*. <https://www.postgresql.org/docs/> (Дата звернення: 17.09.2024).

React Hook Form. (n. d.). *Документація для React Hook Form*. <https://react-hook-form.com/> (Дата звернення: 27.09.2024).

#### **References**

MongoDB. (n. d.). *MongoDB*. <https://www.mongodb.com/docs/atlas>

NestJS. (n. d.). *NestJS*. <https://nestjs.com/>

PostgreSQL. (n. d.). *PostgreSQL*. <https://www.postgresql.org/docs/>

React Hook Form. (n. d.). <https://react-hook-form.com/>

**Отримано редакцією журналу / Received: 23.09.24**

**Прорецензовано / Revised: 27.09.24**

**Схвалено до друку / Accepted: 01.10.24**

**Oleh PROIDAKOV, Student**

**ORCID ID: 0009-0003-6603-2101**

**e-mail: oled.proidakov@gmail.com**

**Taras Shevchenko National University of Kyiv, Kyiv, Ukraine**

### **ANALYSIS AND SELECTION OF TECHNOLOGIES FOR A DIGITAL LIBRARY MANAGEMENT WEB SYSTEM**

*This article explores selecting technologies for the development of a web-based digital library management system using modern technologies such as React, Nest.js, PostgreSQL, wagmi, and viem. The system is designed to facilitate book data update requests and enable paid book publications.*

*It details the architecture and functionality of the platform. It outlines some of capabilities of the platform, highlighting the integration with blockchain technology for facilitating transactions.*

**Keywords:** *digital library, web3, cryptocurrency payments, React, Nest.js, PostgreSQL, wagmi, viem.*

Автор заявляє про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The author declares no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.