



С. І. Ломницька, Л. М. Журавчак

Національний університет "Львівська політехніка" м. Львів, Україна

РЕАЛІЗАЦІЯ ВЕБЗАСТОСУНКУ ДЛЯ КНИЖКОВОГО МАГАЗИНУ-МАРКЕТПЛЕЙСУ

Наведено результати дослідження, проектування та опису реалізації вебзастосунку для книжкового магазину-маркетплейсу, призначеного для надання ефективної, масштабованої та зручної платформи для взаємодії покупців і продавців на ринку літературної продукції. Дослідження базувалося на аналізі наявних платформ електронної комерції та програмних продуктів для онлайн-продажу книг, визначенні функціональних та нефункціональних вимог до системи, обґрунтуванні вибору сучасних технологій (React.js, Node.js з фреймворком NestJS, PostgreSQL) та проектуванні тривірневої клієнт-серверної архітектури з використанням інтерфейсу RESTful API та принципи односторінкового застосунку (SPA). Встановлено актуальність розроблення книжкових маркетплейсів, особливо в контексті зростання українського книжкового ринку та потреби у цифрових платформах для видавництва і приватних продавців. Визначено ключові функціональні вимоги, що охоплюють реєстрацію та автентифікацію користувачів (покупців, продавців, адміністраторів), управління каталогом книг із розширеними можливостями пошуку та фільтрації, оброблення замовлень, управління профілями, оголошеннями та платіжними операціями, а також систему відгуків та рейтингів. Обґрунтовано вибір технологій: React.js (v19) з Redux та Material UI для створення динамічного та адаптивного користувацького інтерфейсу; Node.js (v22) з фреймворком NestJS (v9) та TypeScript (v5) для розроблення модульного, масштабованого та надійного серверного рівня застосунку, що реалізує бізнес-логіку та API; PostgreSQL як реляційної СУБД (системи управління базами даних), що забезпечує цілісність та ACID-сумісність даних ACID (англ. *Atomicity, Consistency, Isolation, Durability*), з використанням об'єктно-реляційної проєкції (ORM) Sequelize для взаємодії. Спроектовано та детально описано тривірневу клієнт-серверну архітектуру, що забезпечує чіткий поділ відповідальностей, високу супроводжуваність та можливість незалежного масштабування компонентів. Описано реалізацію основних функціональних модулів, враховуючи управління користувачами, товарними позиціями (книгами), замовленнями, платіжними операціями (інтеграція з PayPal), доставкою та системою сповіщень (Nodemailer). Розроблено користувацький інтерфейс системи "Alexandria", враховуючи принципи юзабіліті (зручності використання) та конкурентний аналіз. З'ясовано, що обраний підхід модульного моноліту на базі NestJS є ефективним для поточного етапу розроблення, забезпечуючи баланс між швидкістю розгортання та потенціалом для майбутнього переходу до мікросервісної архітектури (MSA). Оцінено важливість та реалізацію заходів безпеки, враховуючи автентифікацію на підставі JWT (англ. *JSON Web Token*), захист від таких поширених вебвразливостей, як XSS (англ. *Cross-Site Scripting*), CSRF (англ. *Cross-Site Request Forgery*), SQLi (англ. *SQL Injection*), та використання HTTPS (англ. *Hypertext Transfer Protocol Secure*). Запропонований архітектурно-технологічний опис може слугувати прототипом для розроблення комерційних книжкових маркетплейсів та аналогічних платформ в інших галузях, а подальші дослідження можуть бути спрямовані на розширення функціональності, інтеграцію з системами рекомендацій та впровадження елементів штучного інтелекту для персоналізації користувацького досвіду.

Ключові слова: вебзастосунки; користувацький інтерфейс; клієнт-серверна архітектура; електронна комерція; ринок збуту літератури; односторінковий застосунок.

Вступ / Introduction

Світовий книжковий ринок продовжує демонструвати стабільне зростання з прогнозованим обсягом у \$192.1 млрд до 2030 р. [1]. Паралельно ринок електронної комерції переживає ще стрімкішу експансію, досягнувши \$4.49 трлн у 2024 р. та з прогнозом зростання до \$5.06 трлн у 2025 р. [16]. У цьому контексті онлайн-продажі книг, особливо електронних та аудіоформатів, стають дедалі вагомішим сегментом. Розвиток цифрових технологій та зміна споживчих звичок трансформують традиційний видавничий ланцюг вартості, де онлайн-маркетплейси відіграють роль ключових посеред-

ників, з'єднуючи авторів, видавців та читачів.

Онлайн-платформи є одними з каналів дистрибуції, які найшвидше зростають. Хоча традиційні книгарні все ще роблять значний внесок, купівля книг онлайн зростає завдяки знижкам, доступності та миттєвим покупкам. Ця тенденція підтверджує основну передумову створення онлайн-маркетплейсу. Завдання полягає в тому, щоб диференціюватися від наявних онлайн-ритейлерів та запропонувати унікальну цінність [16].

Розроблення системи управління книжковим маркетплейсом є актуальним напрямом, що відповідає сучасним тенденціям розвитку української культури після її активізації у 2022 році. За даними Міністерства куль-

Інформація про авторів:

Ломницька Соломія Ігорівна, студент-бакалавр, кафедра програмного забезпечення.

Email: solomiia.lomnytska.pz.2021@lpnu.ua; <https://orcid.org/0009-0002-7249-3468>

Журавчак Любов Михайлівна, д-р техн. наук, професор, кафедра програмного забезпечення.

Email: liubov.m.zhuravchak@lpnu.ua; <https://orcid.org/0000-0002-1444-5882>

Цитування за ДСТУ: Ломницька С. І., Журавчак Л. М. Реалізація вебзастосунку для книжкового магазину-маркетплейсу.

Науковий вісник НЛТУ України. 2025, т. 35, № 3. С. 141–149.

Citation APA: Lomnytska, S. I., & Zhuravchak, L. M. (2025). Implementation of a web application for a book marketplace. *Scientific Bulletin of UNFU*, 35(3), 141–149. <https://doi.org/10.36930/40350315>

тури та інформаційної політики України, у 2023 р. кількість нових книжкових видань зросла на 73 %, а наклад – на 203 % [5]. З огляду на це зростає потреба у доступних цифрових платформах для молодих видавництв, що дають можливість реалізовувати продукцію та відстежувати ключові бізнес-показники без значних інвестицій у власні ІТ-рішення. Така система також сприятиме розвитку вторинного ринку книг, який в Україні досі слабо наведений у цифровому форматі, попри його активне зростання у США завдяки екологічним та економічним чинникам.

Актуальність цього дослідження полягає у потребі розроблення та обґрунтування архітектурних і технологічних рішень для створення ефективного та зручного вебзастосунку книгарні-маркетплейсу, що відповідає сучасним вимогам ринку та користувачів. Головним завданням дослідження є проаналізувати наявні програмні рішення та розробити вебзастосунок для книжкового магазину-маркетплейсу, який надаватиме користувачам зручний та ефективний інструмент для взаємодії на ринку книг, даючи змогу легко знаходити та купувати книги з різних джерел, а продавцям – розміщувати свої товари, залучати клієнтів та керувати продажами.

Об'єкт дослідження – реалізація вебзастосунку для книжкового магазину.

Предмет дослідження – методи та засоби реалізації вебзастосунку для організації книгарні-маркетплейсу, які забезпечать зручну та ефективну взаємодію з ним відповідних користувачів – покупців, продавців і адміністраторів мережі.

Мета роботи – спроектувати та реалізувати ефективний, масштабований та зручний щодо використання вебзастосунок для книжкового магазину-маркетплейсу з обґрунтуванням вибору технологій та архітектурних рішень, що забезпечить пряму продаж книг зацікавленим особам через мережу Інтернет.

Для досягнення зазначеної мети визначено такі основні завдання дослідження:

- опрацювати наявні ресурси для аналізу платформ і програмних продуктів для онлайн-продажу книг з аналізом їх архітектури, інтеграційних механізмів і способів управління контентом з позицій масштабованості, безпеки та зручності користувача, що дасть змогу виявити поширені підходи розробників до створення подібних платформ, а також сформувати бачення ключових особливостей, які потрібно врахувати під час розроблення власної системи для забезпечення її конкурентоспроможності та ефективності;
- визначити основні функціональні вимоги до системи (реєстрація, каталогізація, пошук, оброблення замовлень) та нефункціональні критерії (продуктивність, надійність, доступність, масштабованість, захист даних) до програмного забезпечення книгарні-маркетплейсу, що забезпечить чітке розуміння обсягу робіт, пріоритетів розроблення та критеріїв успішності проекту, а також міцну основу для подальшого формування архітектури та вибору технологічних рішень;
- обґрунтувати вибір технологій та фреймворків для реалізації клієнтської та серверної частин вебзастосунку з погляду продуктивності, екосистеми та підтримки необхідних інтеграцій, що дасть змогу створити технологічно сучасну, надійну та ефективну платформу, здатну задовольнити визначені функціональні та нефункціональні вимоги, а також забезпечить зручність розроблення, супроводу та масштабування програмного продукту;
- спроектувати та описати загальну архітектуру вебзастосунку, що забезпечить відповідну масштабованість та належну ефективність роботи вебзастосунку;

- описати реалізацію ключових функціональних модулів системи відповідно до визначених вимог, використовуючи інструменти, що надають сучасні фреймворки та бібліотеки мови JavaScript.

Аналіз останніх досліджень та публікацій. Для обґрунтування проектних рішень щодо розроблення сучасних онлайн-маркетплейсів здійснено аналіз літературних джерел за такими основними напрямками: розглянуто характеристики та стратегії функціонування маркетплейсів як бізнес-моделей; виокремлено технологічні особливості та вибір інструментарію для створення вебзастосунків такого типу; вивчено релевантні архітектурні підходи, зокрема клієнт-серверну модель. Такий поділ дає змогу комплексно охопити теоретичну базу, необхідну для проектування та розроблення книжкового маркетплейсу.

Характеристика та стратегії функціонування сучасних маркетплейсів. У роботі [14] автори детально аналізують, як платформи електронної комерції сприяють інтернаціоналізації малих підприємств (МП), які часто стикаються з обмеженістю ресурсів, потрібних для виходу на глобальні ринки. Зокрема, вони виокремлюють сім платформно-специфічних переваг (ПСП) для маркетплейсів для МП: інфраструктурну, маркетингову, експортну, мережеву, комплементарну, управлінську та комунікаційну. Виявлено, що малі фірми ефективно використовують NLB FSAs (англ. *Non-Location-Bound Firm-Specific Advantages*) через зовнішні ресурси платформи замість власної розбудови цих можливостей, що значно знижує бар'єри для виходу на зовнішні ринки. Для книжкового маркетплейсу загальні ПСП мають бути адаптовані: технологічна – розширений пошук за ISBN, жанрами, роками видання, інтеграція з видавничими базами, DRM-системи (англ. *Digital Rights Management Systems*); маркетингова – таргетовані інструменти для читачів, зв'язок із літературними соцмережами й авторськими акціями. Завдяки логіці екстерналізації платформа надає авторам, видавництвам та книгарням готові рішення та ресурси, які вони не в змозі розробити самостійно, що знижує витрати та стимулює активну екосистему. Успіх такої платформи залежить від здатності ефективно зовнішньо надавати складні функції у простому для користувача вигляді.

У роботі [17] досліджено наміри споживачів щодо офіційних брендів онлайн-магазинів порівняно з онлайн-маркетплейсами, наголошено на важливості розуміння поведінкових шаблонів споживачів під час вибору каналу купівлі. Для книжкового маркетплейсу це означає, що недостатньо тільки запропонувати широкий асортимент книг від різних продавців. Потрібно створити надійну платформу, що мінімізує ризики під час купівлі товару від незалежних продавців. Водночас, як і в офіційних магазинах, покупці найбільше цінують вигідні ціни та нові пропозиції, демонструючи відданість тій чи іншій торговій марці, для інших видів комерції важливими є безпека та захист прав споживачів. Верифікація продавців та їхніх оголошень, надійні платежі, що проходять через систему маркетплейсу, відмовистість майданчика – усе це позитивно впливає на вибір користувачів на користь таких платформ. Окрім цього, постійні користувачі маркетплейсів демонструють тенденцію до дослідження декількох каналів збуту продукції перед купівлею товару, що свідчить про важливість репутації платформи та маркетингових страте-

гій. Урахування цих особливостей дасть змогу ефективно конкурувати і з великими універсальними маркетплейсами, і з прямими каналами продажів від видавництв чи окремих авторів. Платформа повинна бути гнучкою та здатною швидко адаптуватися до цих мінливих запитів, а її архітектура дасть змогу інтегрувати такі інноваційні функції з часом, а також надавати безпечні та надійні канали для реалізації процесу продажів.

У дослідженні [6] визначено, що існує значний вплив онлайн-відгуків та оцінок користувачів на рішення осіб про купівлю на електронних маркетплейсах, до того ж довіра клієнтів виступає як посередницький фактор. Це також підтверджує думку про критичну важливість надійності маркетплейсу під час вибору користувачами такої комерційної платформи, яка часто характеризується наявністю позитивних відгуків на систему, товари та продавців, адже попередній досвід інших користувачів значно впливає на загальну репутацію маркетплейсу.

Технологічні особливості розроблення вебзастосунків для маркетплейсів. Вибір JavaScript фреймворку для розроблення користувацького інтерфейсу є одним з ключових рішень. У роботі [4] автори Jakub Kryk та Małgorzata Plechawska-Wójcik здійснюють багатоаспектний порівняльний аналіз JavaScript фреймворків – React.js та Solid.js. Дослідження показало перевагу Solid.js у продуктивності рендерингу, але відзначило його менш зрілу екосистему порівняно з React.js.

Дещо розширеніший порівняльний аналіз наведено у дисертаційній роботі [19], де детально порівняно найпоширеніші front-end фреймворки та бібліотеки React, Vue, Angular та Svelte за низкою метрик. Згідно з цим дослідженням, Svelte продемонстрував найкращу загальну продуктивність у Lighthouse-тестах для реального застосунку та потребував значно менше коду. Vue також показав високу швидкість завантаження та інтерактивності. React, хоч і був дещо повільнішим у деяких тестах продуктивності та операціях над об'єктною моделлю документа (DOM), разом зі Svelte забезпечував кращу стабільність макета. Angular виявився "найважчим" за розміром пакета та споживанням пам'яті, але показав хороші результати за оновлення конкретних DOM-елементів. Важливо, що Vue та React мали значно менший розмір мінімізованого пакета, що свідчить про їхню легкість та високу швидкість налаштування робочого середовища розробника.

Дослідження [10] вказує на важливість всебічної оптимізації вебсайтів для покращення користувацького досвіду та продуктивності. Автори демонструють, як конкретні техніки (вбудовування критичного CSS (Cascading Style Sheets), асинхронне завантаження JavaScript, оптимізація зображень (критично для обкладинок книг на маркетплейсі), ефективне кешування та зменшення кількості HTTP-запитів) можуть значно покращити показники продуктивності, наприклад, FPS (frames per second) для динамічних елементів. Для книжкового маркетплейсу, де користувачі можуть переглядати великі каталоги та взаємодіяти з численними елементами, ці особливості оптимізації є надзвичайно важливими.

У роботі [3] розглядають Node.js як ефективне серверне середовище JavaScript, особливо продуктивне в операціях введення/виведення та під час опрацювання великої кількості одночасних з'єднань. Це робить No-

de.js потенційно привабливим вибором для розроблення бекенду книжкового маркетплейсу. Також у цьому джерелі згадують AJAX (Asynchronous Javascript and XML) як фундаментальну техніку для асинхронної взаємодії між клієнтом та сервером, яка лежить в основі сучасних SPA і забезпечує динамічне оновлення контенту без перезавантаження сторінки – ключова вимога для інтерактивного маркетплейсу. Сучасні фронтенд-фреймворки (React, Vue чи Angular) інкапсулюють подібну функціональність, спрощуючи розробникам створення таких взаємодій.

Архітектурні підходи: клієнт-серверна архітектура. Робота [7] надає цінний огляд фундаментальних клієнт-серверних моделей, що є основою для розроблення будь-якого складного вебзастосунку, включно з книжковим маркетплейсом. Автори визначають клієнт-серверну архітектуру як модель, де клієнтські комп'ютери запитують ресурси у сервера через мережу. Сервер обробляє ці запити та повертає відповідь. Ця модель дає змогу багатьом користувачам одночасно отримувати доступ до одних і тих самих ресурсів, що є фундаментальною вимогою для маркетплейсу. Основні переваги, виокремлені в цій статті, які є критично важливими для книжкового маркетплейсу, є такими: централізоване зберігання даних та віддалений доступ, масштабованість, тобто здатність додавати нові сервери (горизонтальне масштабування) або збільшувати потужність наявних (вертикальне), спрощене та дешевше обслуговування ресурсів, можливість балансування навантаження та відновлення після збоїв. Огляд клієнт-серверних моделей, наведений у цій статті, підтверджує, що ця архітектура є фундаментальною основою для розроблення вебзастосунку книжкового маркетплейсу. Для такого проєкту найбільш релевантною буде трирівнева або n-рівнева (мікросервісна) архітектура. Трирівнева архітектура надає поділ на рівень наведення, застосунку та даних, що зумовлює кращу супроводжуваність через модульність, легшу масштабованість кожного з окремих рівнів залежно від потреб системи, безпеку та продуктивність.

Матеріали та методи дослідження. Дослідження способів розроблення вебзастосунку книгарні-маркетплейсу базувалося на низці сучасних програмних засобів, що забезпечують повторюваність експериментальних результатів і контроль можливих джерел похибок. Для реалізації односторінкового клієнтського інтерфейсу використано документацію та кінцевий імпортований код низки поширених фронтенд бібліотек мовою JavaScript. React.js v19 [12] є класичним рішенням для створення користувацьких інтерфейсів у вебзастосунку поєднано з React Redux для централізованого управління станом, React Router DOM для маршрутизації, Axios для виконання HTTP-запитів. Для стилізації компонент використано матеріали, надані бібліотекою Material UI [9].

Серверну частину створено на платформі Node.js v22 з використанням NestJS для побудови модульної архітектури з Dependency Injection та TypeScript для статичної типізації.

Аналіз предметної області виконано за методикою Domain-Driven Design з оглядом наявних рішень у сфері маркетплейсів із відкритим кодом та вивченням специфікацій RESTful-архітектур [15]. Проєктування архітектури книжкового маркетплейсу спиралося на трирів-

неву модель: Presentation Tier – клієнтський інтерфейс на React.js, Application Tier – сервер, що складається з модулів NestJS, і Data Tier – база даних на PostgreSQL, що дало змогу проводити незалежну верифікацію кожного рівня.

Результати дослідження та їх обговорення / Research results and their discussion

Система книжкового маркетплейсу орієнтована на три основні ролі користувачів. Покупець визначається як фізична або юридична особа, що здійснює пошук, вибір, замовлення та оцінку книг, та взаємодіє з продавцями. Продавець – фізична або юридична особа, яка розміщує пропозиції щодо продажу книг, керує власними оголошеннями, опрацьовує замовлення та взаємодіє з покупцями, причому кожен продавець може також виступати в ролі покупця. Третьою ключовою роллю є адміністратор – уповноважена особа, відповідальна за управління застосунком і обліковими записами користувачів, за модерування контенту та забезпечення належного функціонування платформи.

Зважаючи на аналіз літературних джерел, можна зробити висновок про основні потреби цих груп користувачів. Покупцям потрібний зручний пошук книг з фільтрами та сортуванням, доступ до детальної інформації про товари (опис, ціни, відгуки) та продавців. Ключовими є можливість зберігати книги у списку ба-

жань, залишати відгуки, а також проста реєстрація, авторизація, безпечне оформлення й відстеження замовлень та керування власним профілем.

Продавцям потрібні інструменти для створення й адміністрування оголошень. Важливими є ефективне опрацювання замовлень, доступ до аналітики продажів через інформаційну панель, можливість отримувати відгуки та взаємодіяти з покупцями, зберігаючи при цьому функціонал покупця.

Адміністраторам потрібний функціонал для комплексного управління платформою, враховуючи адміністрування облікових записів користувачів, модерацію контенту (оголошень, відгуків) для дотримання правил та якості, а також інструменти для моніторингу, забезпечення стабільної та безпечної роботи маркетплейсу.

Враховуючи основні потреби користувачів, наведені вище, розроблено вебзастосунок для книжкового магазину-маркетплейсу.

Загальна архітектура вебзастосунку. Для побудови книжкового маркетплейсу використано тривірневу клієнт-серверну архітектуру, що містить клієнтський рівень, серверний рівень та рівень даних (рис. 1). Такий підхід забезпечує чіткий поділ відповідальностей, модульність та можливість незалежного масштабування кожного з рівнів. Взаємодію між клієнтським та серверним рівнями здійснюють за допомогою RESTful web API.

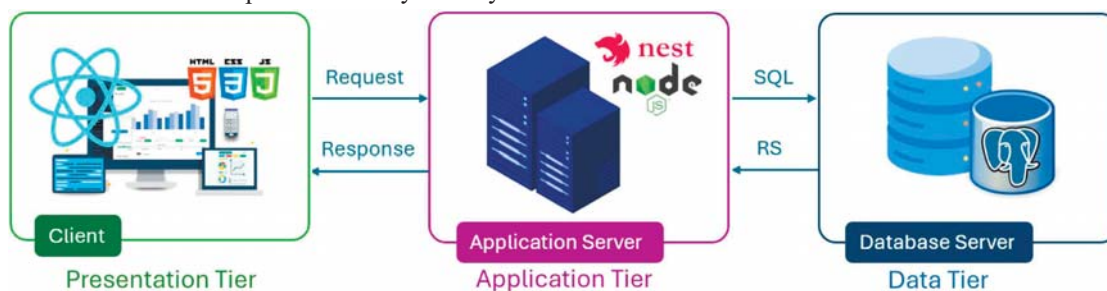


Рис. 1. Архітектура книжкового маркетплейсу / Book marketplace architecture

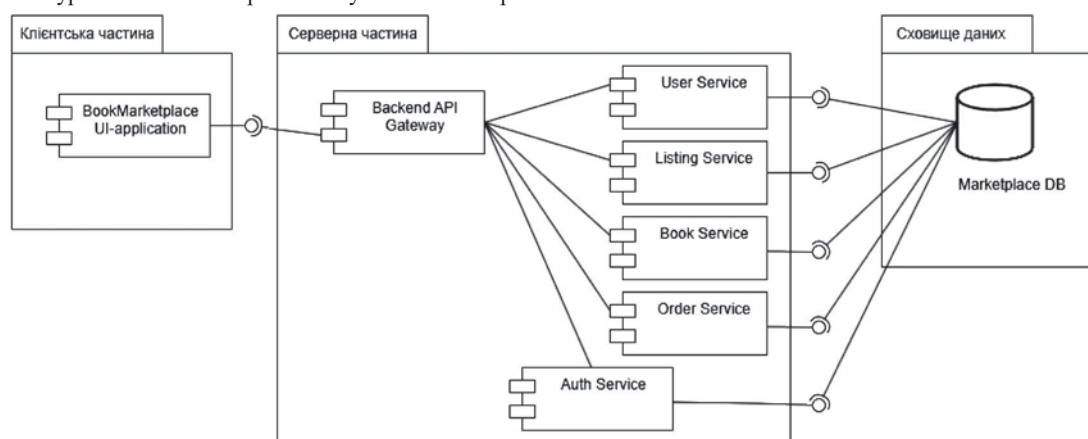


Рис. 2. Діаграма компонентів тривірневої клієнт-серверної архітектури вебзастосунку / Component diagram of a three-tier client-server architecture of a web application

Клієнтський рівень (Presentation Tier) реалізовано як односторінковий застосунок з використанням бібліотеки React.js. Це забезпечує динамічний рендеринг інтерфейсу та безперебійну навігацію (рис. 2). Управління станом здійснюють за допомогою Redux та React Context API. Стилiзацію виконано з використанням Material UI, а для розроблення використано мову TypeScript.

Серверний рівень (Application Tier) відповідає за бізнес-логіку, оброблення запитів, безпеку та взаємодію з рівнем даних. Він функціонує на платформі Node.js з

використанням фреймворку NestJS. NestJS забезпечує структуровану модульну архітектуру та ґрунтується на Dependency Injection. Взаємодію з базою даних PostgreSQL здійснюють через ORM Sequelize. Передбачено інтеграцію зі сторонніми API: PayPal для платежів, Google Maps для картографії, "Нова Пошта" для доставки та Nodemailer для email-сповіщень.

Рівень даних (Data Tier) відповідає за постійне зберігання даних. Використовують СУБД PostgreSQL, що гарантує ACID-сумісність транзакцій та цілісність даних.

Діаграма компонентів тривірневої клієнт-серверної архітектури вебзастосунку (див. рис. 2) демонструє її ключові модульні блоки та взаємозв'язки між ними: поділ на клієнтську (UI-застосунок) та серверну частини (з Backend API Gateway та набором спеціалізованих сервісів: User, Listing, Book, Order, Auth), і сховище даних (Marketplace DB). Клієнтський рівень взаємодіє з мікросервісами через API шлюз, а кожен сервіс має доступ до централізованої бази даних для виконання своїх функцій.

Для розгортання системи використовують три окремі фізичні сервери, що відповідають основним компонентам описаної тривірневої архітектури.

Розроблення програмного забезпечення для книжкового маркетплейсу. Реалізація функціональних вимог охоплює всі особливості взаємодії трьох класів користувачів (покупця, продавця, адміністратора). Особливу увагу приділено управлінню каталогом книг, враховуючи пошук та фільтрацію, відображенню пропозицій від різних продавців, системі відгуків, управлінню замовленнями та адмініструванню платформи. Ці функції відповідають найкращим практикам для маркетплейсів.

Діаграма прецедентів, подана на рис. 3, наглядно візуалізує функціональність системи з погляду користувача для ідентифікації ролей (акторів), формалізації вимог, полегшення комунікації між учасниками проєкту, підтримки аналізу та проєктування архітектури, а також як основа для створення тест-кейсів і забезпечення цілісного уявлення про систему. Система має чотири основні типи акторів, об'єднаних відношенням узагальнення: користувач (доступ до базових функцій: реєстрація, авторизація, перегляд і пошук книг, профіль), покупець, продавець і адміністратор.

Дотримання нефункціональних вимог, зокрема щодо безпеки (хешування паролів, захист від поширених вразливостей), надійності (резервне копіювання) та продуктивності (оптимізація завантаження сторінок), є критично важливим для успішної експлуатації платформи. Використання сучасних інструментів для забезпечення якості коду (ESLint, Prettier, TypeScript) сприяє зручності супроводу системи.

Основний сценарій використання застосунку полягає у взаємодії об'єктів під час виконання функцій системи, відображаючи послідовність викликів методів. У сценарії оформлення замовлення покупець додає товари до кошика, вводить адресу й платіжні дані, після чого інтерфейс передає цю інформацію серверу. Сервер валідує дані, створює записи замовлення, товарів і доставки, опційно обробляє платіж та надсилає підтвердження. У сценарії створення оголошення продавець заповнює форму, а сервер, за потреби оновивши довідкові дані, зберігає оголошення зі статусом "На модераторі" та додає зображення.

Логіка процесів системи полягає у послідовності дій та рішень і паралельності потоків з використанням вузлів дій, розгалуження, доріжок та потоків керування. Оформлення замовлення передбачає етапи взаємодії покупця із системою під час придбання книг: додавання товарів до кошика, перевірку, введення даних доставки й оплати, підтвердження замовлення. Система при цьому обробляє кошик, перевіряє наявність і ціни, створює записи замовлення, взаємодіє з платіжним сервісом і надсилає сповіщення.

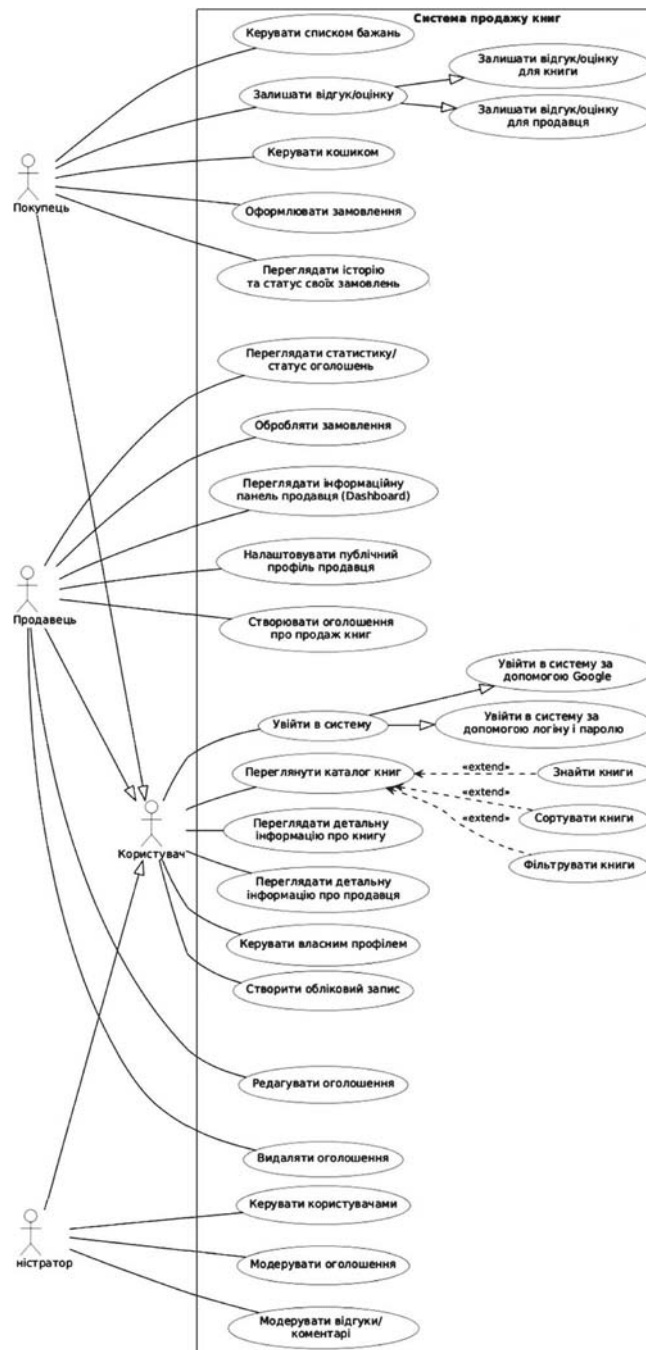


Рис. 3. Діаграма прецедентів функціональності системи з погляду користувача для ідентифікації ролей / Use case diagram of the system functionality from a user perspective to identify roles

Фронтенд застосунку книжкового маркетплейсу реалізовано на базі React v19 із використанням Redux – для управління станом, React Router – для маршрутизації сторінок та Material UI – для побудови сучасного інтерфейсу користувача. Бекенд створено з використанням Nest.js на платформі Node.js v22. Ця частина застосунку також містить основні інтеграції зі сторонніми програмними інтерфейсами. Для роботи з базою даних PostgreSQL застосовують ORM Sequelize, яка забезпечує зручну та ефективну взаємодію між серверною логікою та персистентними даними.

Унаслідок розроблення застосунку створено користувацький інтерфейс, що відображає інформацію, отриману зі створеного API. Розроблення користувацького інтерфейсу системи ґрунтувалося на специфікації функціональних вимог, діаграмі прецедентів та принципах юзабіліті, зокрема евристичних правилах Нільсена й

правилах ергономіки взаємодії людини з комп'ютером. Для підвищення зручності та інтуїтивності користувацького інтерфейсу враховано специфіку функціонування системи як багаторольового книжкового маркетплейсу (покупець, продавець, адміністратор), а також здійснено конкурентний аналіз UI провідних платформ (наприклад, Amazon, Goodreads, BookFinder). Особливу

увагу приділено структурі компонентів користувацького інтерфейсу, його lazy-loading сторінок, обробленню помилок через глобальні сповіщення та реалізації toast-нотифікацій про стан системи. Розроблено логотип системи, названої "Alexandria", обрано основну кольорову гаму, шрифти (Inter, Roboto) та іконки з бібліотеки Material Symbols (рис. 4).

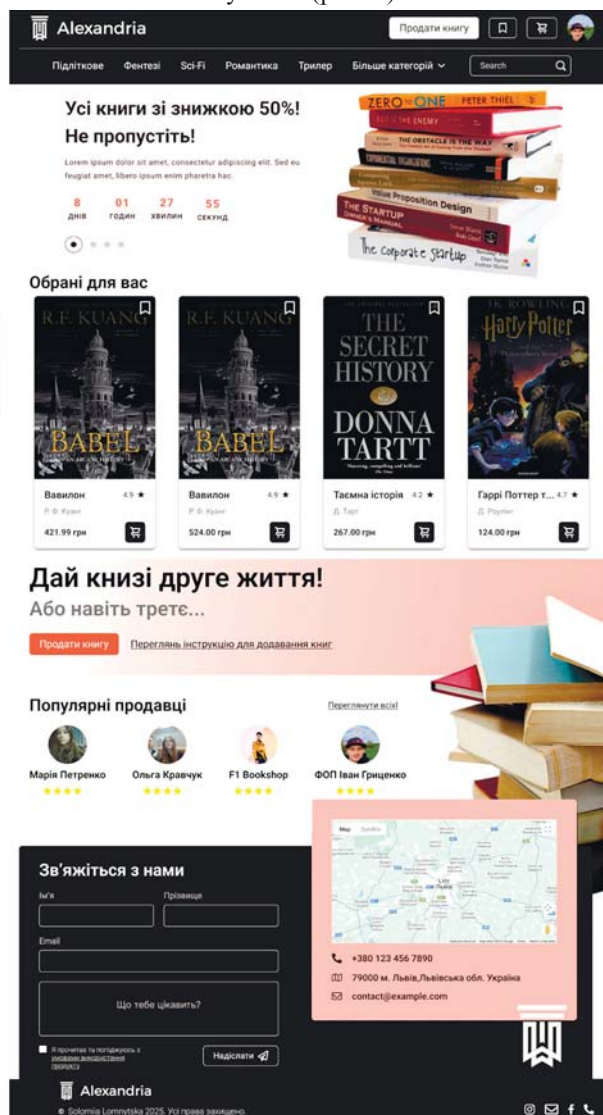
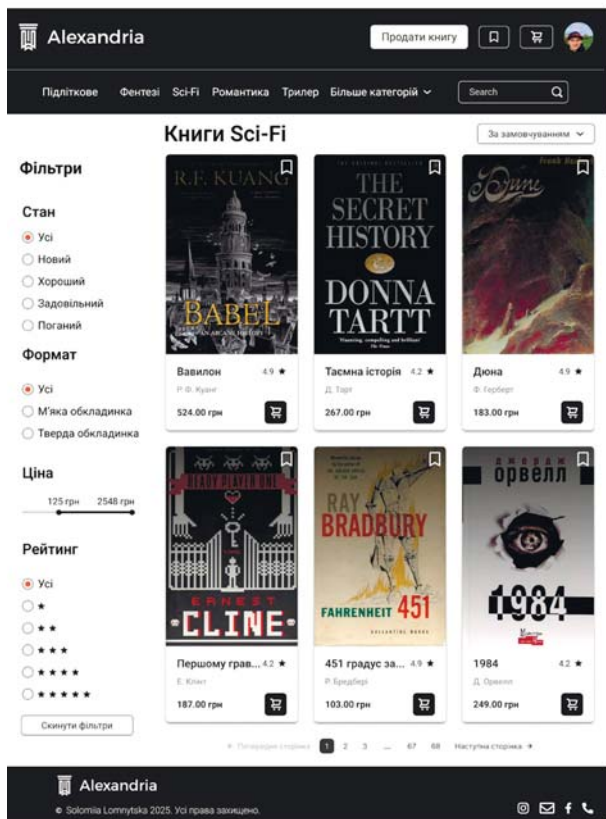


Рис. 4. Каталог книг (a) та головна сторінка маркетплейсу (b) / Books catalogue (a) and a marketplace main page (b)

Реалізація функціональних і нефункціональних ви-
мог системи забезпечила всебічне охоплення процесів
взаємодії ключових категорій користувачів (покупця,
продавця та адміністратора) у межах електронного кни-
жкового маркетплейсу. Трирівнева клієнт-серверна ар-
хітектура системи комунікації між користувачами кни-
жкового маркетплейсу, побудована на підставі сучас-
них тенденцій розроблення вебзастосунку у поєднанні з
ORM-рішенням для управління базою даних, забезпе-
чує надійність і масштабованість застосунку та відпо-
відність вимогам продуктивності цієї системи. Сформо-
ваний унаслідок розроблення графічний користуваць-
кий інтерфейс відповідає принципам зручності його ви-
користання, враховує основні евристичні UI та UX і ві-
дображає специфіку домену.

Однією з основних особливостей системи, яку мож-
на удосконалити в розвитку платформи, є впроваджен-
ня мікросервісної архітектури та системи рекомендацій
з використанням засобів штучного інтелекту.

Обговорення результатів дослідження. Сучасні
платформи електронної комерції тяжіють до мікросер-
вісної архітектури (MSA) для гнучкості та масштабован-
ості. У дослідженні Idris Oumoussa та Rajaa Saidi [8]
аналізують методології ідентифікації мікросервісів за
міграції від монолітів, вказуючи на переваги MSA для
цифрової продуктивності. Модульний підхід у NestJS
можна розглядати як крок до MSA. Такий підхід забез-
печує кращу супроводжуваність коду та створює потен-
ціал для незалежного масштабування окремих функці-
ональних блоків (управління каталогом книг, оброблен-
ня замовлень чи адміністрування користувачів). Це уз-
годжується з висновками щодо переваг гранулярності в
архітектурі.

У роботі Dāvis Kažemaks та Jérémie Decouchant [2]
розглядають надійність роботи мікросервісної архітек-
тури (MSA), зазначаючи, що її поділ на менші сервіси
сприяє масштабуванню системи, але вводить нові її
вразливості. Водночас фреймворк NestJS, будучи

придатним для модульних монолітів та мікросервісів, надає переваги масштабованості, але й ставить виклики щодо надійності та управління.

Продуктивність і масштабованість застосунків, розроблених на NestJS, значною мірою визначаються асинхронною природою платформи Node.js, на якій він базується. Це дає змогу ефективно обробляти велику кількість операцій введення-виведення, що є типовим для платформ електронної комерції (наприклад, численні запити до бази даних для отримання інформації про товари, оброблення замовлень, взаємодія з файловою системою для зберігання зображень тощо). Можливість горизонтального масштабування способом запуску декількох екземплярів NestJS-додатку за балансувальником навантаження є ключовим фактором для забезпечення високої доступності та продуктивності під час пікових навантажень. У дослідженні [11] вказано на важливість досягнення швидкого часу відгуку бекенду для забезпечення якісного користувацького досвіду.

У дослідженні [18] автори наголошують на центральній ролі персоналізованих користувацьких інтерфейсів для оптимізації ефективності онлайн-магазинів. Вони пропонують фреймворк, який передбачає збирання даних про поведінку клієнтів для подальшого надання "спеціалізованого користувацького інтерфейсу". React, завдяки своїй компонентній архітектурі, гнучким можливостям управління станом (наприклад, за допомогою Context API, Redux або Zustand) та декларативному підходу до побудови UI, є чудовим інструментом для реалізації таких персоналізованих і динамічних інтерфейсів. Розроблений з використанням React інтерфейс отримує технологічну базу для ефективного впровадження механізмів персоналізації (відображення рекомендованих книг на підставі попередніх покупок, історії переглядів або вподобань інших користувачів зі схожими інтересами). Це повністю відповідає тенденціям, окресленим у роботі [18], щодо підвищення релевантності пропозицій для кожного окремого користувача.

У роботі [13] наголошують на те, що в умовах такої великої різноманітності пропозицій на ринку підприємства електронної комерції повинні розробляти інноваційні способи утримання або залучення клієнтів і що надійні системи рекомендацій є критично важливими для досягнення цієї мети. React, як бібліотека для побудови користувацьких інтерфейсів, дає змогу легко інтегрувати такі системи рекомендацій (які можуть бути реалізовані на бекенді або як окремі сервіси) та ефективно відображати їх результати користувачеві в режимі реального часу без потреби перезавантаження всієї сторінки. Порівнюючи з висновками цього дослідження, розроблений книжковий маркетплейс, маючи React на фронтенді, володіє потрібною технологічною основою для впровадження складних рекомендаційних алгоритмів, що, безперечно, покращить користувацький досвід та підвищить конкурентоспроможність платформи.

Безпека є критичною для електронної комерції. У дослідженні [20] обговорюють релевантні загрози, зокрема, впровадження фальшивих даних або атаки з перехопленням. Це свідчить про нагальну потребу захисту API-ендпоінтів NestJS та каналів зв'язку (HTTPS). Дослідження акцентує на потребі мінімізації поверхні атаки для вебсерверів. Робота, розглядаючи мікросервіси в IoT, зазначає, що MSA може посилити безпеку через ізоляцію, але й вводить нові ризики. Стандартні заходи

безпеки для маркетплейсу охоплюють автентифікацію/авторизацію (JWT), захист від XSS, CSRF, SQLi (NestJS надає засоби, React екранує дані), HTTPS та валідацію вхідних даних. Вибір сучасних фреймворків (NestJS та React) часто надає розробникам певні вбудовані механізми захисту від поширених вразливостей. Наприклад, React автоматично екранує динамічні дані, що вставляються в DOM, що значно знижує ризик XSS-атак. NestJS завдяки своїй структурі та екосистемі полегшує інтеграцію різноманітних middleware для безпеки.

Аналіз обговорених результатів дослідження підтверджує доцільність проведеної роботи. Як видно з аналізу наявної науково-технічної літератури, автори розглядають тільки окремі особливості розроблення платформ електронної комерції, зокрема питання архітектури, їхню надійність, продуктивність серверних рішень на базі Node.js та NestJS, особливості створення персоналізованих користувацьких інтерфейсів за допомогою React та питання безпеки, однак комплексного дослідження з детальним описом проєктування та реалізації саме книжкового маркетплейсу як модульного моноліту з використанням специфічного стеку технологій (React.js, Node.js з NestJS, PostgreSQL) та з урахуванням потреб українського ринку не було виявлено. Тому представлена робота заповнює цю нішу, пропонуючи обґрунтоване архітектурно-технологічне рішення та його практичну реалізацію для вказаної предметної області, що може слугувати прототипом для аналогічних розроблень.

Отже, внаслідок виконаної роботи можна сформулювати такі наукову новизну та практичну значущість результатів дослідження.

Наукова новизна отриманих результатів дослідження – удосконалено методику розроблення спеціалізованих вебплатформ для електронної комерції типу маркетплейс, зокрема для ринку літературної продукції, яка, на відміну від наявних, науково обґрунтовує практичне застосування комплексної технологічної інтеграції сучасних вебтехнологій (React.js, Node.js з фреймворком NestJS, та PostgreSQL) з тривірневою клієнт-серверною архітектурою, використанням інтерфейсу RESTful API та принципів SPA, що забезпечує ефективну взаємодію користувачів різних ролей та закладає основу для подальшого розвитку функціональності й масштабування системи.

Практична значущість результатів дослідження – результати дослідження використано для створення функціонального прототипу книжкового маркетплейсу та можна застосувати як прикладну модель для розроблення комерційних електронних платформ продажу, що потребують реалізації функціоналу управління користувачами, каталогами товарів, системами замовлень та контентом.

Висновки / Conclusions

Спроектовано та реалізовано ефективний, масштабований та зручний щодо використання вебзастосунків для книжкового магазину-маркетплейсу з обґрунтуванням вибору технологій та архітектурних рішень, який забезпечує прямий продаж книг зацікавленим особам через мережу Інтернет. За результатами проведеного дослідження можна зробити такі основні висновки.

1. Проаналізовано сучасні тенденції ринку електронної комерції та онлайн-продажу книг, що підтвердило ак-

- туальність розроблення книжкового маркетплейсу, особливо для українського ринку. Визначено основні функціональні (реєстрація, каталогізація, пошук, опрацювання замовлень) та нефункціональні вимоги (продуктивність, надійність, масштабованість, безпека) до програмної системи.
- Обґрунтовано вибір сучасних технологій, що містить React.js для клієнтської та Node.js із фреймворком NestJS для серверної частин, а також PostgreSQL як систему управління базами даних. Такий вибір забезпечує високу продуктивність, масштабованість та зручність розроблення й підтримки вебзастосунку.
 - Розроблено та описано тривірневу клієнт-серверну архітектуру (Presentation Tier, Application Tier, Data Tier) для книжкового маркетплейсу. Взаємодію між рівнями реалізовано за допомогою інтерфейсу RESTful API. Запропонована архітектура системи сприяє модульній організації системи, легкості масштабування та супроводжуваності.
 - Описано реалізацію ключових функціональних модулів системи, таких як управління користувачами з різними ролями (покупець, продавець, адміністратор), каталог книг з фільтрацією та пошуком, система оформлення замовлень, управління оголошеннями та адміністрування платформи. Реалізований вебзастосунок "Alexandria" надає зручний інструмент для взаємодії на ринку книг, даючи можливість користувачам знаходити та купувати книги, а продавцям – ефективно керувати продажами.
 - Підтверджено, що інтеграція обраних вебтехнологій з тривірневою архітектурою є ефективним рішенням для створення багатofункціональних маркетплейсів, здатних підтримувати потреби сучасного ринку електронної комерції літературної продукції. Перспективи подальшого розвитку охоплюють розширення функціональності, інтеграцію з додатковими зовнішніми сервісами та впровадження елементів персоналізації.

References

- Global Book Market – Size, Trends, and Future Outlook. BookMandee. URL: <https://bookmandee.com/blog/global-book-market-size-trends-and-future-outlook/>
- Kažemaks, D., & Decouchant, J. (2025). SoK: Microservice Architectures from a Dependability Perspective. URL: <https://arxiv.org/abs/2503.03392>
- Khan Mohd, T., Thompson, J., Carmine, A., & Reuter, G. (2022). Comparative Analysis on Various CSS and JavaScript Frameworks. *Journal of Software*, 282–291. <https://doi.org/10.17706/jsw.17.6.282-291>
- Kryk, J., & Plechawska-Wójcik, M. (2025). Multi-aspect comparative analysis of JavaScript programming frameworks – React.js and Solid.js. *Journal of Computer Sciences Institute*, 34, 68–75. <https://doi.org/10.35784/jcsi.6712>
- Kryzhnia, M. (2023, March 2). "Publishing works by Ukrainian authors is a matter of principle." Publishing houses established during the full-scale war. *Suspilne Kultura*. URL: <https://suspilne.media/culture/394499-vidanna-ukrainskih-avtoriv-principove-pitanna-vidavnictva-zasnovani-pid-cas-povnomasstabnoi-vijni>
- Novela, S., Sihombing, Y., Hansopaheluwakan, S., & Aurellia, C. (2023). How Online Customer Review and Online Customer Rating Influence Customer Purchase Decision on E-Marketplace with Customer Trust as A Mediator. In: *Proceedings of the International Conference on Technology and Management (ICTMOD)*, 1–6. <https://doi.org/10.1109/ICTMOD59086.2023.10472906>
- Nyabuto, G. (2023). Client-server Architecture, a Review. *International Journal of Advanced Science and Computer Applications*, 3(1), 1–4. <https://doi.org/10.47679/ijasca.v3i1.48>
- Oumoussa, I., & Saidi, R. (2025). The Ontology-Based Mapping of Microservice Identification Approaches: A Systematic Study of Migration Strategies from Monolithic to Microservice Architectures. *Computers*, 14(4), article ID 133. <https://doi.org/10.3390/computers14040133>
- Overview – Material UI. MUI: The React component library you always wanted. URL: <https://mui.com/material-ui/getting-started/>
- Parwal, Harsh, R, Radha, & X, Anita. (2025). An Experimental Study on Website Optimization: Test Your Strategies. *Procedia Computer Science*, 259, 660–669. <https://doi.org/10.1016/j.procs.2025.04.016>
- Perri, D., Simonetti, M., & Gervasi, O. (2022). Deploying Efficiently Modern Applications on Cloud. *Electronics*, 11(3), article ID 450. <https://doi.org/10.3390/electronics11030450>
- React Reference Overview. React. URL: <https://react.dev/reference/react>
- Riaz, H., Baig, U., Meidute-Kavaliauskiene, I., & Ahmed, H. (2021). Factors Effecting Omnichannel Customer Experience: Evidence from Fashion Retail. *Information*, 13(1), article ID 12. <https://doi.org/10.3390/info13010012>
- Singh, N., Munjal, S., & Kundu, S. K. (2023). Marketplace platforms as game changers: Internationalization of smaller enterprises. *Journal of International Management*, article ID 101035. <https://doi.org/10.1016/j.intman.2023.101035>
- Taleb, M. (2025). Use and Design REST APIs. URL: https://www.researchgate.net/publication/389944668_Use_and_Design_REST_APIs
- The Business Research Company. (2025). E-commerce Market Report (Report ID: 5735260). URL: <https://www.researchandmarkets.com/reports/5735260/e-commerce-market-report>
- Wang, Y. B., & Chen, P. Y. (2024). Official online brand stores or online marketplaces? exploring innovation in consumer intentions. *Journal of Innovation & Knowledge*, 9(3), article ID 100541. <https://doi.org/10.1016/j.jik.2024.100541>
- Wasilewski, A. (2024). Functional Framework for Multivariant E-Commerce User Interfaces. *Journal of Theoretical and Applied Electronic Commerce Research*, 19(1), 412–430. <https://doi.org/10.3390/jtaer19010022>
- Xu, Wenqing. (2021). Benchmark Comparison of JavaScript Frameworks React, Vue, Angular and Svelte [Dissertation, University of Dublin]. URL: <https://publications.scss.tcd.ie/theses/diss/2021/TCD-SCSS-DISSERTATION-2021-020.pdf>
- Zhukabayeva, T., Zholshiyeva, L., Karabayev, N., Khan, S., & Alnazzawi, N. (2025). Cybersecurity Solutions for Industrial Internet of Things-Edge Computing Integration: Challenges, Threats, and Future Directions. *Sensors*, 25(1), article ID 213. <https://doi.org/10.3390/s25010213>

S. I. Lomnytska, L. M. Zhuravchak

Lviv Polytechnic National University, Lviv, Ukraine

IMPLEMENTATION OF A WEB APPLICATION FOR A BOOK MARKETPLACE

The research, design, and implementation of the Alexandria web application, a specialized book marketplace conceptualized to furnish an efficient, scalable, and user-centric platform for literary product commerce, was presented, thereby addressing the evolving needs of both consumers and vendors in the digital era. An exhaustive analysis of contemporary e-commerce ecosystems and existing online book retail solutions was conducted, which enabled the precise definition of both functional and non-functional system requirements. Furthermore, this paper supported the careful selection of a modern technology stack: React.js for the client side, Node.js with the NestJS framework for server-side logic, and PostgreSQL for database management. The subsequent architectural design focused on a three-tier client-server model, RESTful API integration, and Single-Page Application (SPA) principles. The

study established the growing significance of dedicated book marketplace platforms, particularly highlighting the increasing demand in the Ukrainian literary market for accessible digital infrastructure tailored to both publishers and individual sellers. Critical functional requirements were identified, including multi-role user authentication and management (buyers, sellers, administrators), advanced book catalog features with powerful search and filtering options, streamlined order processing workflows, comprehensive user profile and listing management, and also an integrated system for customer reviews and ratings. The chosen technology stack was justified as follows: React.js (v19), supported by Redux and Material UI, enabled a dynamic, responsive, and engaging presentation tier; simultaneously, Node.js (v22) with NestJS (v9) and TypeScript (v5) provided a modular, scalable, and maintainable application tier responsible for business logic and API services. PostgreSQL, accessed via the Sequelize object – relational mapper (ORM), formed the data tier, ensuring data integrity and ACID-compliant transactions. Implemented three-tier architecture, highlighting its inherent modularity, enhanced maintainability, and capacity for independent component scaling. The development encompassed key functional modules, such as user management systems, intuitive product (book) catalog interactions, seamless order and payment processing (with PayPal integration), logistics coordination and automated notification services via Nodemailer. Additionally, Alexandria platform's user interface (UI) and user experience (UX) design were showcased, demonstrating compliance with established usability heuristics and insights gleaned from competitive analyses. The system was implemented as a modular monolith, a strategic decision that provided the advantage of agile initial deployment while maintaining a clear pathway for potential migration to the Microservices Architecture (MSA) if future scalability demands such a transition. The benefits of this study also include a strong emphasis on security, with implemented measures encompassing JWT-based authentication, robust defences against prevalent web vulnerabilities (e.g., XSS, CSRF, SQL injection), and the mandatory use of HTTPS for all data transmission. The research provided a validated architectural and technological blueprint, adaptable for developing similar sophisticated marketplace applications across various domains. Future work could extend to incorporating advanced features like AI-driven recommendation engines and enhanced personalization to further enrich the user experience.

Keywords: web development; user interface; client-server architecture; e-commerce; literature market; single-page application (SPA).