Науковий вісник НЛТУ України
Scientific Bulletin of UNFU
ISSN 1994-7836 (print)
ISSN 2519-2477 (online)
https://doi.org/10.36930/40350310
Article received 13.05.2025
Article accepted 29.05.2025
UDC 004.[4'2+5+9]
Correspondence author: Kh. O. Hryshkanych khrystyna.hryshkanych.pz.2021@lpnu.ua

***Х. О. Гришканич, Л. М. Журавчак, Т. О. Муха***
*Національний університет "Львівська політехніка" м. Львів, Україна*

# РОЗРОБЛЕННЯ ВЕБЗАСТОСУНКУ ЗАСОБАМИ ТЕХНОЛОГІЇ MERN ДЛЯ УТВОРЕННЯ СПІЛЬНОТИ КНИЖКОВОГО СЕРЕДОВИЩА

Виявлено, що на більшості сучасних цифрових платформ, зокрема – маркетплейсів та електронних бібліотек, спостерігають недостатній рівень підтримки читацької взаємодії та обговорення змісту літературних творів. З'ясовано, що в таких середовищах функціонал рецензування має другорядне значення або зведений до технічного оцінювання товару, що ускладнює формування активної читацької спільноти. З огляду на цю проблему розроблено вебзастосунок "Форум обговорення книг", призначеного для обміну думками між користувачами щодо змісту та вражень від прочитаних книг. Реалізовано архітектуру за принципом клієнт-серверної моделі з використанням набору технологій MERN (MongoDB, Express.js, React.js, Node.js) у поєднанні з мовою програмування TypeScript, що забезпечило масштабованість, типобезпеку, швидкість розроблення та гнучкість модифікації функціоналу. Оцінено вплив REST-архітектури на зручність підтримки і взаємодії між клієнтом і сервером. Для підвищення роздільності логіки і спрощення підтримки бекенд реалізовано як набір сервісів, кожен з яких відповідає за окремі особливості бізнес-логіки: автентифікацію, управління книгами, рецензіями, оцінками та книжковими полицями. Охарактеризовано компонентну структуру інтерфейсу, засновану на React з використанням Redux та Context API для управління станом застосунку. Основною компонентою інтерфейсу є Layout, що охоплює Header, Sidebar і динамічну зону контенту, куди відображається відповідний екран залежно від маршруту. Особливу увагу приділено формуванню користувацького досвіду: застосовано принципи UI/UX-дизайну, враховуючи візуальну ієрархію, кольорову гармонію, адаптивність та інтуїтивну навігацію. Оцінено вплив оптимізаційних практик на продуктивність: використано React.memo, useCallback, React.lazy, WebP, а також налаштовано підтримку HTTP/2. Проведений аналіз засвідчив ефективність запропонованих рішень у контексті зниження тривалості завантаження, покращення взаємодії та зменшення відмов користувачів. Перспективами подальших досліджень є створення ефективних механізмів управління користувацьким контентом, зокрема, впровадження автоматизованих систем модерації, фільтрація деструктивних або спам-повідомлень, а також розроблення алгоритмів виявлення неконструктивної поведінки користувачів. Окрему увагу доцільно приділити застосуванню штучного інтелекту для персоналізації контенту й рекомендацій, що сприятиме покращенню користувацького досвіду та залученню до тематичних обговорень.

**Ключові слова:** обговорення змісту книг; клієнт-серверна архітектура; користувацький досвід; UI/UX-дизайн; цифрове рецензування.

## Вступ / Introduction

У сучасному світі цифрових технологій зростає потреба у створенні платформ, що сприяють взаємодії та об'єднанні користувачів за їхніми інтересами. Особливо актуальною є спільнота книжкового середовища, яке відіграє ключову роль у культурному та освітньому розвитку суспільства. Книги не тільки забезпечують доступ до нових знань, а й об'єднують людей, формуючи спільноти за жанрами, авторами чи тематиками.

Останніми роками активно популяризуються різноманітні онлайн-платформи, які розглядають книги, зокрема – маркетплейси та електронні бібліотеки. Проте більшість із таких ресурсів робить акцент саме на комерційній складовій або швидкому доступі до книги як товару. Можливості для обговорення змісту творів, зазвичай, обмежені другорядними функціями або повністю відсутні. Відгуки користувачів, якщо вони передбачені, зазвичай стосуються технічних аспектів (якості видання чи зручності сервісу), а не аналізу літературного змісту чи ідей автора. Це створює дефіцит спеціалізованих просторів для читачів, які прагнуть поглибленого обговорення прочитаного, що зумовлює потребу розроблення вебзастосунку, основний функціонал якого буде зосереджено саме на рецензуванні та обговоренні контенту книг, водночас доповненого додатковими різноманітними функціями для розширення можливостей надання пропозицій користувачами і відповідно підвищення їхньої зацікавленості.

*Об'єкт дослідження* – проєктування вебзастосунку

**Інформація про авторів:**
**Гришканич Христина Олегівна**, студент-бакалавр, кафедра програмного забезпечення.
**Email:** khrystyna.hryshkanych.pz.2021@lpnu.ua; https://orcid.org/0009-0005-6925-0424
**Журавчак Любов Михайлівна**, д-р техн. наук, професор, кафедра програмного забезпечення.
**Email:** liubov.m.zhuravchak@lpnu.ua; https://orcid.org/0000-0002-1444-5882
**Муха Тарас Орестович**, канд. техн. наук, ст. викладач, кафедра програмного забезпечення.
**Email:** taras.o.mukha@lpnu.ua; https://orcid.org/0009-0000-1188-3580
**Цитування за ДСТУ:** Гришканич Х. О., Журавчак Л. М., Муха Т. О. Розроблення вебзастосунку засобами технології Mern для утворення спільноти книжкового середовища. Науковий вісник НЛТУ України. 2025, т. 35, № 3. С. 96–107.
**Citation APA:** Hryshkanych, Kh. O., Zhuravchak, L. M., & Mukha, T. O. (2025). Development of a web application using Mern stack to create book environment community. *Scientific Bulletin of UNFU*, 35(3), 96–107. https://doi.org/10.36930/40350310

для формування спільноти книжкового середовища.

*Предмет дослідження* – методи та засоби для реалізації вебзастосунку, спрямованого на формування спільноти книжкового середовища, яке забезпечить набуття користувацького досвіду для ефективного рецензування, обговорення та оцінювання літературних творів.

*Мета роботи* – спроєктувати та реалізувати вебзастосунок для утворення спільноти книжкового середовища, який дасть змогу обговорювати зміст книг та обмінюватися думками та враженнями стосовно прочитаного, що сприятиме самовираженню його користувачів, полегшить пошук і вибір літератури за жанрами, авторами чи тематиками.

Для досягнення зазначеної мети визначено такі основні *завдання дослідження*:

- проаналізувати наукові праці, статті та книги щодо методів і засобів розроблення форуму для обговорення змісту книг, що дасть змогу сформувати ключові чинники, які необхідно врахувати під час розроблення власного вебзастосунку для забезпечення якісного користувацького досвіду через створення візуально привабливого інтерфейсу;
- дослідити можливі архітектурні підходи до вебзастосунків і спроєктувати таку архітектуру, яка буде водночас раціональною для впровадження та ефективною в роботі, що забезпечить гнучкість модифікації, масштабованість і легку інтеграцію функціоналу для рецензування книг та ведення дискусій;
- розробити вебзастосунок "Форум обговорення книг", використовуючи сучасні технології та інструменти, що дасть змогу підтримувати обмін даними в режимі реального часу.

**Аналіз останніх досліджень та публікацій.** Питання впливу типу платформи (соціальної мережі чи маркетплейсу) на стиль, глибину та активність рецензування книг стало предметом дослідження у праці [4]. У межах цієї роботи проведено масштабний аналіз оглядів книг на платформах Goodreads та Amazon із фокусом на жанрі біографій. Для цього проаналізовано понад 21 тис. книг і 2,5 млн рецензій (1,6 млн на Goodreads, 945 тис. на Amazon). Результати показали, що на Goodreads, де огляди не прив'язані до процесу купівлі, спільнота більш активна і схильна до детальних рецензій (у середньому 2,75 огляду на користувача), тоді як на Amazon, орієнтованому на продажі, користувачі залишають менше оглядів (1,51 на користувача). Водночас на Amazon понад 63 % оцінок становлять п'ять зірок, що вказує на високий рівень позитивного підкріплення покупцями: користувачі часто ставлять максимальні оцінки, щоб підтримати продавця або висловити своє загальне задоволення від покупки, навіть не заглиблюючись у зміст книги. Тоді як на Goodreads частіше виставляють чотири зірки, оскільки користувачі більше орієнтуються на змістовний аналіз твору, діляться враженнями про сюжет, персонажів, стиль написання та критичніше підходять до оцінювання книги загалом. У підсумку дослідження підтвердило, що контекст платформи істотно впливає на поведінку користувачів: у комерційно орієнтованих середовищах огляди здебільшого зводяться до технічного оцінювання продукту, тоді як на платформах із соціальним фокусом спільнота більше схильна до глибокого аналізу літературного твору як культурного явища.

У роботі [18] автор розглядає, як розвиток медіа та цифрових технологій змінив способи створення, поширення та споживання літератури. Автор зазначає, що до традиційних засобів передачі літературного контенту (книги, театр, радіо) раніше мала доступ тільки певна частина населення. Наразі цифрові платформи та соціальні мережі подолали ці бар'єри, зробивши літературу доступною для широкої глобальної аудиторії. У статті зазначають, що цифрове середовище сприяє появі нових форм літературної творчості: блогів, аудіовізуальних оповідей, інтерактивного контенту. Окрім цього, сучасні медіа відкривають авторам можливість самостійно публікувати й просувати свої твори, сприяючи культурному обміну та економічному зростанню.

У роботі [14] автори показують, що активна участь в онлайн-форумах має значні індивідуальні та соціальні переваги. Зокрема, ідентифікація користувачів з онлайн-спільнотою підвищує їхнє задоволення життям і стимулює участь у громадських ініціативах поза Інтернетом. Наприклад, обговорення змісту книги з проблем екології, може спонукати учасників форуму долучитися до екологічного руху або організувати власну акцію на захист довкілля. Окрім цього, важливою складовою частиною цього процесу є можливість певної анонімності, що особливо цінно для користувачів, які стикаються зі стигматизацією або побоюються засудження. Тому книжкові форуми, де учасники часто спілкуються під псевдонімами, стають важливим соціальним простором для читачів, які прагнуть відкритого і безпечного обговорення літератури.

Особливої уваги заслуговує досвід використання форумів у навчальному середовищі, наведений у роботі [13]. Тут досліджено вплив онлайн-форуму "CSC forum" на інтерактивне навчання та комунікацію між студентами й викладачами. Форум надає можливість обміну ідеями та враженнями, створення публікацій з модерацією адміністратором та ведення живих чатів для оперативної взаємодії. Аналіз результатів тестів показав, що студенти, які брали участь у форумі, мали кращі академічні показники (максимум 19 балів із 20), ніж ті, хто не використовував платформу (максимум 16 балів). Живі дискусії сприяли відкритішому спілкуванню з викладачами, зниженню стресу та пришвидшенню зворотного зв'язку. Обмеження успішності функціонування платформи пов'язані з низькою відданістю користувачів своїм вподобанням, технічними проблемами та ризиками появи неякісних або образливих постів.

У контексті проєктування інтерфейсів і врахування користувацьких очікувань важливою є стаття [9], де автори аналізують динамічність користувацького досвіду (UX) у контексті тривалого використання продуктів. Проведене чотиритижневе лонгітюдне дослідження показало, що на початкових етапах взаємодії користувачі приділяють більше уваги візуальній привабливості інтерфейсу, тоді як з часом визначальними стають функціональні та практичні особливості. Отримані результати підтверджують важливість врахування змін у сприйнятті UX під час проєктування цифрових продуктів, особливо тих, що передбачають тривалу взаємодію.

У технічному аспекті створення вебзастосунку важливим є забезпечення його продуктивності. У роботі [1] розглядають способи покращення роботи вебзастосунків, зосереджуючись на удосконаленні фронтенду для кращого досвіду користувачів. У межах дослідження створено інтернет-магазин, для якого налаштовано продуктивність роботи інтерфейсу. Зокрема, зменшено

кількість HTTP-запитів, зменшено розміри сторінок і графічних матеріалів, мінімізовано HTML-, CSS- та JavaScript-код, а також переконвертовано зображення у стислі формати (зокрема – JPEG). Реалізовано використання мережі доставки контенту (CDN) та об'єднано скрипти, що дає змогу скоротити тривалість оброблення. Для зменшення обсягу переданих даних застосовано ефективний алгоритм стискання GZIP (GNU zip), який зменшує розмір текстового контенту перед його передаванням мережею. Окрім цього, JavaScript-код обробляється відкладено, впроваджено кешування, усунуто зайві перенаправлення та зменшено кількість DNS-запитів. Для асинхронного завантаження даних без перезавантаження сторінки використано сучасну технологію AJAX (англ. *Asynchronous JavaScript and XML*), що забезпечує динамічність інтерфейсу та зменшує час очікування користувача. Порівняння роботи системи до і після удосконалення, включно з тестами навантаження на 3000, 5000 і 10000 одночасних користувачів, показало значне покращення швидкості і стабільності роботи. Автор зазначає, що оптимізація фронтенду має більший вплив на загальну швидкодію і є дешевшою, ніж оптимізація бекенду, оскільки саме тривалість завантаження й оброблення на стороні користувача найсильніше впливає на сприйняття роботи вебзастосунку.

Аналіз наукових джерел засвідчив наявність досліджень, у яких розглядають вплив цифрових технологій на літературу, особливості функціонування онлайн-форумів, динаміку користувацького досвіду та методи вдосконалення вебзастосунків. Водночас, серед публікацій відсутні роботи, які комплексно розглядають особливості створення спеціалізованої платформи саме для обговорення змісту книг – більшість наявних рішень орієнтовані на загальні особливості спілкування або комерційне поширення контенту й не сприяють активному розвитку читацької комунікації. Окрім цього, не досліджено адекватних рішень, спрямованих на забезпечення стабільності та швидкодії роботи вебзастосунків, що підтримують спілкування в режимі реального часу. Це зумовлює актуальність і новизну цього дослідження, яке комплексно поєднує такі особливості розроблення вебзастосунку, спрямованого на формування спільноти для обговорення книг.

**Матеріали та методи дослідження.** Дослідження проведено з використанням методів проєктування клієнт-серверної архітектури та UI/UX-аналізу; сучасних технологій вебпрограмування; мови програмування TypeScript; набору технологій MERN; бібліотеки управління станом Redux і механізму Context API; підходів до удосконалення продуктивності React-застосунків (React.memo, useCallback, React.lazy); форматів зображень WebP; протоколу HTTP/2; інструментів REST API; засобів юзабіліті-тестування та аналізу користувацького досвіду.

## Результати дослідження та їх обговорення / Research results and their discussion

***Опис архітектури вебзастосунку.*** У процесі дослідження можливих архітектурних рішень для розроблення вебзастосунку "Форум обговорення книг" проаналізовано сучасні підходи до побудови програмних систем, зокрема, монолітну та мікросервісну. Відповідні висновки ґрунтуються на результатах наукової праці [10], де проведено порівняльний аналіз цих двох моделей з погляду їх масштабованості, гнучкості розроблення, ізоляції або уникнення помилок, продуктивності роботи та складності супроводу.

Монолітна архітектура є традиційним рішенням, яке передбачає об'єднання всіх компонентів застосунку в єдину структуру. Основними її перевагами є простота розроблення, швидке розгортання, спрощене тестування та налагодження завдяки централізованій кодовій базі. Окрім цього, монолітна архітектура забезпечує високу продуктивність за рахунок внутрішньопроцесної взаємодії компонентів без потреби використання мережевих протоколів. Проте зростання складності застосунку значно ускладнює його масштабування та супровід: кожне оновлення вимагає повного перезбирання і розгортання системи, а тісна зв'язаність модулів створює ризики виникнення каскадних помилок.

На відміну від моноліту, мікросервісна архітектура забезпечує поділ застосунку на незалежні сервіси, кожен з яких виконує чітко визначену бізнес-функцію, має власний інтерфейс взаємодії та окрему базу даних. Серед її переваг варто відзначити можливість автономного масштабування окремих сервісів, підвищену гнучкість процесу розроблення, ізоляцію збоїв та незалежність вибору технологій для кожного компонента. Водночас мікросервіси мають й істотні недоліки: зростає складність інтеграції та управління консистентністю даних, з'являються додаткові накладні витрати на оркестрацію, моніторинг та налаштування міжсервісної взаємодії. Для вебзастосунку "Форум обговорення книг", де важливим є баланс між функціональністю та ефективністю впровадження, складність мікросервісної архітектури може бути невиправданою.

Враховуючи ці фактори, прийнято рішення використовувати класичну клієнт-серверну архітектуру вебзастосунку з розділенням на фронтенд і бекенд. Такий підхід забезпечує оптимальний баланс між складністю розроблення, масштабованістю та зручністю підтримки. Клієнтська частина буде відповідати за формування інтерфейсу користувача, серверна частина – за оброблення запитів, взаємодію з базою даних та реалізацію бізнес-логіки.

Для організації взаємодії клієнта і сервера використано архітектурний стиль REST (англ. *Representational State Transfer*). Він використовує стандартні HTTP-методи (GET, POST, PUT, PATCH, DELETE та інші) для доступу до ресурсів, що є об'єктами даних. Кожен ресурс має свій унікальний ідентифікатор (URI) і до нього можна дістатися через певний HTTP-метод, який визначає, яку саме операцію потрібно виконати щодо нього. Архітектурний стиль REST передбачає декілька ключових принципів, які забезпечують ефективність і гнучкість системи [13]. Один з таких принципів – відсутність стану, тобто кожен запит до сервера містить всю потрібну інформацію для його оброблення. З цього випливає, що сервер не зберігає дані про попередні запити, і саме це дає змогу зменшити складність системи. Інший важливий принцип – кешування. Часто запитувані дані зазвичай зберігаються на сервері клієнта для швидшого доступу та зниження навантаження на нього. Окрім цього, архітектурний стиль REST передбачає, що всі відповіді від сервера мають однакову структуру, а це робить роботу з даними зручною і цілком передбачуваною як для користувача.

У контексті статичної особливості архітектури вебзастосунку на рис. 1 наведено діаграму компонентів

пропонованої системи. Вона показує загальну структуру вебзастосунку та взаємозв'язки між його складовими компонентами. Програмне забезпечення складається з трьох ключових компонентів: клієнтської частини (фронтенду), реалізованої як React-застосунок у браузері, серверної частини (бекенду) на базі середовища Node.js/Express та бази даних, що функціонує на MongoDB – документо-орієнтованій СКБД.

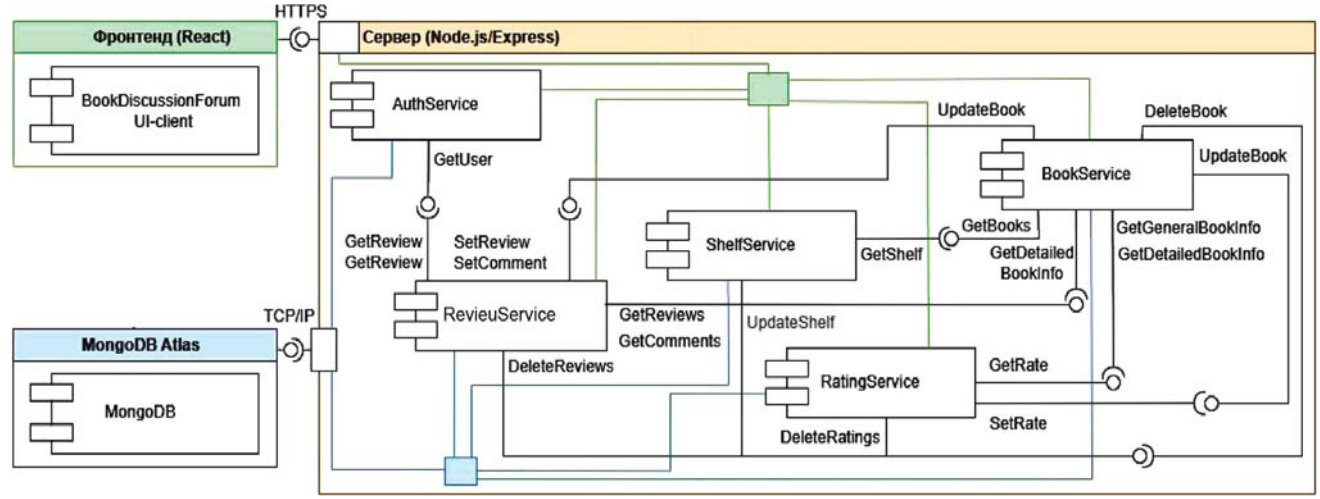


**Рис. 1.** Діаграма компонентів системи / System component diagram

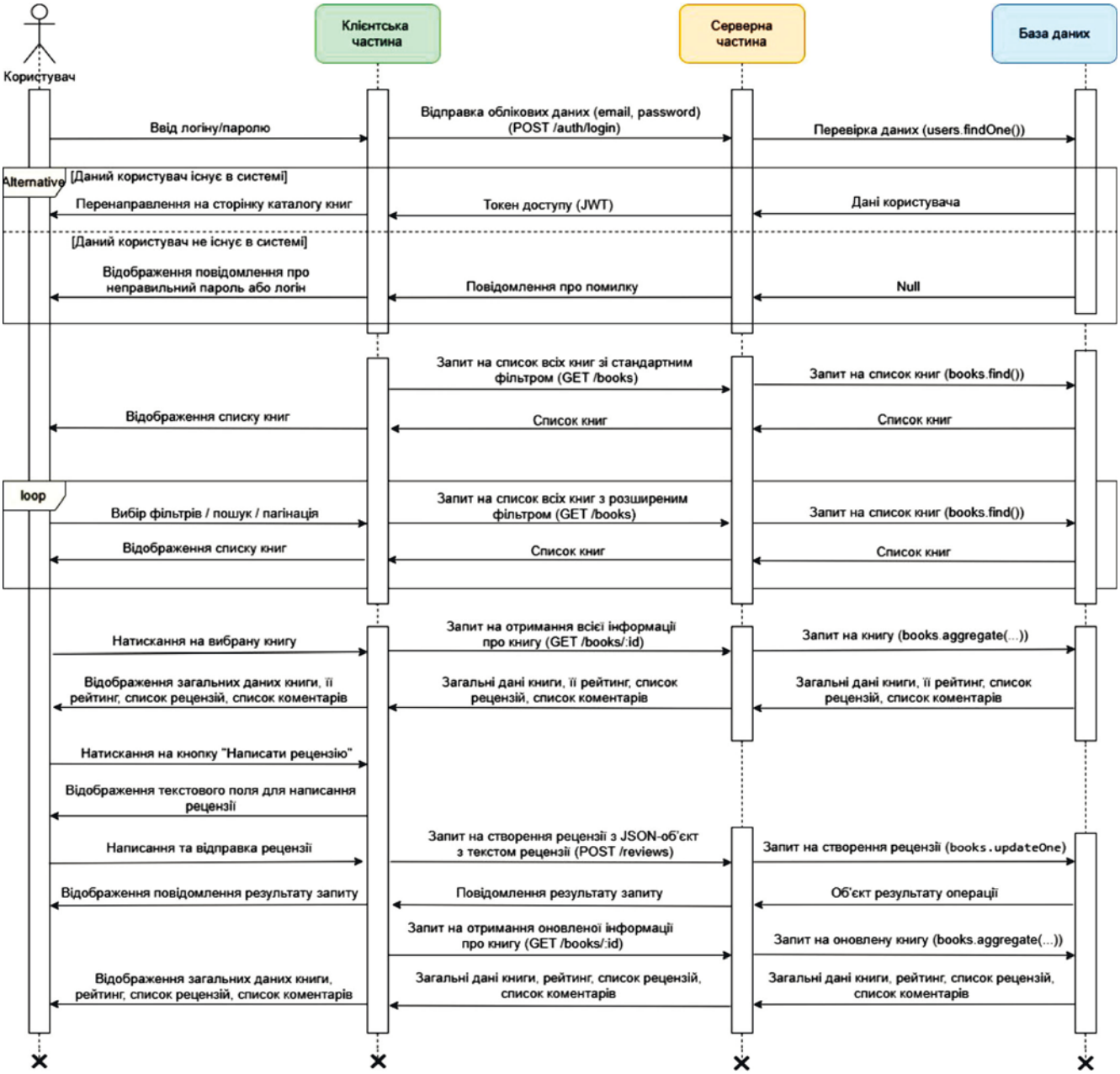


**Рис. 2.** Діаграма послідовності / Sequence diagram

Клієнтська частина взаємодіє із сервером через протокол HTTPS (англ. *HyperText Transfer Protocol Secure*), який працює на прикладному рівні. HTTPS, як безпечна версія HTTP, гарантує конфіденційність і цілісність даних завдяки шифруванню за технологією TLS (англ. *Transport Layer Security*).

Бекенд, водночас, для обміну даними з базою даних використовує протоколи TCP/IP. TCP (англ. *Transmission Control Protocol*) забезпечує надійну доставку даних у правильному порядку, тоді як IP (англ. *Internet Protocol*) відповідає за їхню адресацію та маршрутизацію між мережевими вузлами.

Описані складові частини системи охоплюють такі ключові компоненти:

- BookDiscussionForum UI Client – клієнтська частина, що відповідає за інтерфейс користувача та взаємодію з ним. Реалізована за допомогою React, вона обробляє дії користувача та здійснює обмін даними з сервером через HTTPS-запити з використанням бібліотеки Axios;
- AuthService – сервіс автентифікації, що відповідає за реєстрацію, вхід до системи та генерацію токенів доступу (JWT). Інші сервіси отримують і використовують ідентифікаційні дані, створені цим компонентом;
- BookService – модуль, який обробляє запити до книжкового каталогу, забезпечує сортування, пошук, фільтрацію та пагінацію. Для відображення розширеної інформації про книгу звертається до ReviewService, щоб отримати кількість рецензій і коментарів. У разі видалення книги також ініціює відповідні дії в ReviewService, RatingService (для очищення відгуків і оцінок), а також ShelfService (для видалення книги з полиць користувачів);
- ReviewService – сервіс, що відповідає за створення, оновлення та видалення рецензій і коментарів. Після публікації рецензії запитує у AuthService дані автора, а потім ініціює оновлення лічильника рецензій у BookService;
- RatingService – компонент для реалізації системи оцінювання книг за п'ятибальною шкалою. Після кожної оцінки звертається до BookService для оновлення середньої оцінки та кількості рецензентів;
- ShelfService – сервіс, що забезпечує керування віртуальними книжковими полицями користувача ("прочитано", "плануються прочитати"). Для відображення інформації про книги на полицях взаємодіє з BookService;

MongoDB – база даних, у якій зберігається основна інформація системи: облікові записи користувачів, дані про книги, рецензії, оцінки, полиці та коментарі.

Для відображення поведінки системи у часі використано діаграму послідовностей (див. рис. 2), яка відображає один із її основних сценаріїв – процес створення рецензії користувачем: від авторизації до успішного збереження введених даних.

Спочатку користувач вводить свої облікові дані для входу. Клієнтська частина надсилає ці дані на сервер, який звертається до бази даних для перевірки їхньої коректності. Якщо автентифікація проходить успішно, користувач переходить до перегляду каталогу книг; у разі помилки виводиться відповідне повідомлення. У діаграмі цей етап відображено як альтернативна гілка сценарію. Після входу користувач бачить список доступних книг. Він має змогу фільтрувати їх за заданими критеріями, здійснювати пошук за назвою або переходити між сторінками, за кожної дії клієнт формує новий запит до сервера, який отримує відповідні дані з бази та повертає оновлений список.

Під час вибору конкретної книги система формує запит на отримання детальної інформації, що охоплює загальні відомості, середню оцінку, рецензії та коментарі до них. Отримані дані завантажуються та відображаються на сторінці книги.

Далі користувач натискає кнопку для створення рецензії, вводить текст у запропоноване поле та надсилає його. Ця інформація передається на сервер, де створюється новий запис рецензії у базі даних, а також оновлюється загальна кількість рецензій для книги. Після цього клієнт повторно запитує оновлені дані книги, щоб відобразити нову рецензію разом з іншою актуальною інформацією.

***Особливості реалізації вебзастосунку.*** Для реалізації вебзастосунку "Форум обговорення книг" обрано набір технологій MERN. Такий вибір обґрунтовано результатами дослідження, наведеного у праці [3], де автори докладно аналізують можливості кожного з компонентів і демонструють їхню ефективність під час створення сучасних вебзастосунків. У статті наголошено, що MERN є зручною повноцінною JavaScript-екосистемою з трирівневою клієнт-серверною архітектурою: фронтендом на React, серверною логікою на Node.js + Express та базою даних MongoDB. Такий підхід дає змогу працювати над усіма рівнями застосунку єдиною мовою, що значно спрощує розроблення, налагодження і підтримання коду. Додаткова підтримка TypeScript ще більше підвищує стабільність і передбачуваність проєкту завдяки статичній типізації.

React.js забезпечує динамічну побудову інтерфейсу, ефективну роботу з віртуальним DOM (англ. *Document Object Model* – об'єктна модель документа, що описує структуру HTML-сторінки) та високу продуктивність завдяки оновленню тільки необхідних компонентів. У разі нашого форуму, де велика кількість дій користувачів пов'язана із взаємодією в реальному часі (додавання рецензій, коментарів, оцінок), це дає змогу забезпечити плавний і швидкий інтерфейс без зайвих перезавантажень.

Для реалізації серверної частини використано Node.js у поєднанні з фреймворком Express.js. Ефективність цієї технології підтверджують результати дослідження, проведеного у праці [15], у якій порівнювали продуктивність REST API, реалізованих на Node.js і PHP. У межах експерименту імітували до 1000 одночасних запитів до двох REST-ендпоінтів, а продуктивність оцінювали за середньою тривалістю відповіді та пропускною здатністю. Node.js показав стабільно кращі результати: середня тривалість відповіді становила 136-156 мс проти 244-266 мс у PHP, пропускна здатність Node.js залишалась на рівні 100 % до 1000 запитів, тоді як у PHP вона впала до 48,7 %. Отже, результати дослідження демонструють перевагу Node.js в умовах високого навантаження, що пояснюють його неблокуючою подієвою архітектурою. Саме тому цю технологію обрано для проєкту.

MongoDB як NoSQL база даних забезпечує гнучке зберігання документів, що чудово підходить для збереження рецензій, коментарів, профілів та книг.

Отже, застосування набору технологій MERN дає змогу ефективно забезпечити низку технічних завдань: швидкий відгук системи, масштабовану архітектуру, зручне керування асинхронними операціями та пришвидшене розроблення завдяки єдиній мовній екосистемі. Це робить його чудовим рішенням для створення функціонального, продуктивного та зручного вебзастосунку для обговорення книг.

Серверну частину системи розроблено за підходом Code First. Це означає, що спочатку створено Mongoose-моделі – шаблони, які визначають дані, що зберігатимуться в базі, поля кожного об'єкта, їхні типи даних та зв'язки між об'єктами. Після цього структура колекцій у базі даних MongoDB формувалась автоматично на підставі цих моделей. У проєкті реалізовано моделі для таких основних сутностей:

- User містить дані про користувачів;
- Book описує книги, наведені в каталозі;
- Author зберігає інформацію про авторів;
- Genre визначає жанри книг;
- Review містить рецензії користувачів на книги;
- Comment зберігає коментарі до рецензій;
- Rating відповідає за оцінки, поставлені книгам;
- Shelf описує книги, додані користувачами на власні віртуальні полиці.

На підставі створених моделей реалізовано відповідні контролери, які зосереджують бізнес-логіку роботи з кожною сутністю. Вони відповідають за оброблення запитів від клієнта: приймають вхідні дані, звертаються до необхідних сервісів або бази даних, обробляють результати та формують відповіді, які повертаються клієнту. У системі реалізовано такі контролери:

- AuthController забезпечує функціональність реєстрації, авторизації та видачі токенів після входу користувача. Також відповідає за перевірку прав доступу та ідентифікацію користувача, який надсилає запит;
- BookController обробляє запити, що стосуються книг: отримання списку, перегляд детальної інформації, пошук, фільтрація, сортування й пагінація. Окрім цього, саме тут оновлюється рейтинг книги після виставлення оцінки;
- ReviewController відповідає за створення, редагування та перегляд рецензій до книг. У ньому реалізовано логіку коментарів: додавання, перегляд і прив'язка коментарів до відповідних рецензій. Така структура дає змогу поєднати пов'язані функції в одному місці;
- RatingController реалізує додавання оцінок до книг і обчислення середнього рейтингу на підставі всіх оцінок. Також перевіряє, чи вже користувач оцінював книгу, і дає змогу змінити оцінку за потреби;
- ShelfController керує віртуальними книжковими полицями. Забезпечує додавання книг до розділів "Прочитано" та "Плануються прочитати", перегляд вмісту полиць і переміщення книг між ними.

У контролерах для взаємодії з базою даних застосовують бібліотеку Mongoose, яка спрощує роботу з колекціями MongoDB, використовуючи попередньо визначені моделі. Наприклад, для отримання переліку всіх книг у BookController викликають метод Book.find (), що забезпечує легке отримання масиву всіх документів без потреби у складних запитах. Такий спосіб дає змогу оперативно реалізовувати основні CRUD-операції – створення (Create), читання (Read), оновлення (Update) та видалення (Delete) даних. Проте, коли виникає потреба в отриманні складних пов'язаних даних (список рецензій із вкладеними коментарями) використовують механізм агрегаційних запитів (aggregation pipeline). Агрегація дає змогу здійснювати послідовне оброблення даних за допомогою таких операцій: об'єднання колекцій ($lookup), відбір потрібних записів ($match), впорядкування результатів ($sort), групування ($group) тощо.

У системі для автентифікації користувачів, перевірки прав доступу та валідації параметрів запитів застосовують middleware-функції, які забезпечують опрацювання запиту до його потрапляння в контролер і часто можуть використовуватися для багатьох маршрутів. Наприклад, authMiddleware перевіряє авторизацію користувача та надає доступ до захищених маршрутів тільки після успішної перевірки. Проте однієї тільки авторизації за токеном недостатньо для контролю прав доступу до конкретних дій. Наприклад, тільки адміністратор має можливість додавати, редагувати або видаляти книги. Для цього в системі реалізовано окремий middleware checkRole, який перевіряє, чи роль користувача відповідає очікуваній.

Окрім цього, у системі активно використовуються middleware-валідатори, що здійснюють перевірку вхідних даних перед їхнім обробленням. Наприклад, у запиті, що містить ідентифікатор певної сутності, проводиться перевірка на його відповідність формату MongoDB ID. Це дає змогу запобігти помилкам на рівні бази даних та оперативно інформувати клієнта про некоректно сформований запит.

Клієнтську частину реалізовано на підставі компонентного підходу, який передбачає створення інтерфейсу з окремих, незалежних елементів, кожен з яких відповідає за конкретну частину сторінки або функціонал. Це дає змогу легко змінювати, розширювати або перевикористовувати окремі частини інтерфейсу без потреби переписувати весь код.

Одним з основних компонентів застосунку є Layout. Він слугує загальною обгорткою для всіх сторінок і містить такі незмінні елементи: заголовок з навігацією та бічну панель, яку можна показувати або приховувати за потреби. Це забезпечує зручність для користувача, оскільки навігаційне меню та бічна панель доступні на всіх ключових сторінках, де це потрібно. Компонент Layout відображає вміст сторінок, що розміщені в папці screens, серед яких каталог книг, деталі про книгу, віртуальні полиці користувача та інші розділи.

Кожна сторінка (screen) будується з менших компонентів, таких як контейнери для відображення книги чи рецензії, кнопки для взаємодії із вмістом системи, модальні вікна для підтвердження дій тощо. Під час реалізації було враховано, що багато елементів будуть використовувати у різних місцях застосунку, тому їх створено як універсальні базові компоненти та винесено в спеціальну окрему директорію. Наприклад, компонент для відображення даних про книгу застосовують як на сторінці каталогу, так і на сторінці особистих віртуальних полиць користувача.

Для реалізації інтерфейсу активно застосовували бібліотеку Material UI, яка є одним із найпоширеніших рішень для створення сучасних та візуально привабливих React-застосунків. Ця бібліотека містить багато готових компонентів (кнопки, діалоги, меню, таблиці, списки та інші), які мають добрий вигляд "з коробки" та легко адаптуються під дизайн проєкту. За потреби їх можна стилізувати під свої потреби за допомогою властивості styled.

Для стилізації звичайних компонентів у проєкті використано технологію SCSS (англ. *Sassy CSS* – розширення синтаксису мови стилів CSS). Вона є зручнішою за стандартний CSS (англ. *Cascading Style Sheets* – каскадна таблиця стилів), оскільки дає змогу робити код чистішим та легшим для підтримки. SCSS дало можливість використовувати вкладеність селекторів, що допомогло краще структурувати стилі під HTML-розмітку

застосунку. Також у проєкті використано змінні для кольорів, шрифтів чи відступів, що виявилося дуже зручним: достатньо змінити значення в одному місці, і воно автоматично застосовувалося в інших частинах застосунку.

У проєкті навігацію між сторінками реалізовано за допомогою бібліотеки React Router. Вона дала змогу організувати маршрути так, аби вебсторінка не потребувала повного перезавантаження для відображення нового вмісту. Замість цього, після початкового завантаження, оновлення здійснюються шляхом динамічної заміни окремих частин сторінки. Такий підхід забезпечив швидку навігацію та створив для користувача позитивний досвід у користуванні застосунком.

Для забезпечення зв'язку між клієнтською частиною та сервером створено окрему директорію services, де розміщено всі API-запити. Кожна сутність має відповідний сервіс. Для виконання цих запитів у проєкті застосовували бібліотеку Axios, що надає простий та зрозумілий спосіб надсилання HTTP-запитів та опрацювання отриманих відповідей.

У застосунку для зберігання та керування даними обрано бібліотеку Redux Toolkit, яка є спрощеною та зручнішою альтернативою класичного Redux. Завдяки їй усі ключові дані (перелік книг, рецензії, користувацькі полиці або інформація про користувача) зберігаються в єдиному централізованому сховищі (store). Це доволі зручно, оскільки доступитись до даних можна з будь-якої частини застосунку, і немає потреби передавати їх від одного компонента до іншого через параметри, навіть якщо вони розміщені на різних рівнях дерева компонентів.

***Формування користувацького досвіду та дизайн інтерфейсу.*** Одним із ключових факторів формування позитивного користувацького досвіду у вебзастосунках є якісний UI/UX-дизайн, що поєднує привабливість, інтуїтивну навігацію та послідовну візуальну ідентичність. У праці [8] досліджено, як окремі компоненти інтерфейсу впливають на залучення та утримання користувачів. На підставі опитувань і юзабіліті-тестування автори дійшли висновку, що найбільший вплив мають такі особливості: дизайн головної сторінки, навігаційна зручність та гармонійне використання кольорів. Зокрема, наголошено, що приваблива й функціональна головна сторінка формує перше враження про платформу та виступає "вхідною точкою" до подальшої взаємодії. Простота навігації дає змогу користувачу швидко знаходити потрібні функції, що критично важливо для зручності та ефективності. Окрім цього, послідовне й добре продумане кольорове оформлення користувацького інтерфейсу підсилює емоційне його сприйняття й створює цілісне відчуття надійності роботи та комфорту під час його використання.

У контексті розроблення вебзастосунку для обговорення змісту книг ці висновки мають прикладне значення. На рис. 3 наведено головну сторінку застосунку.

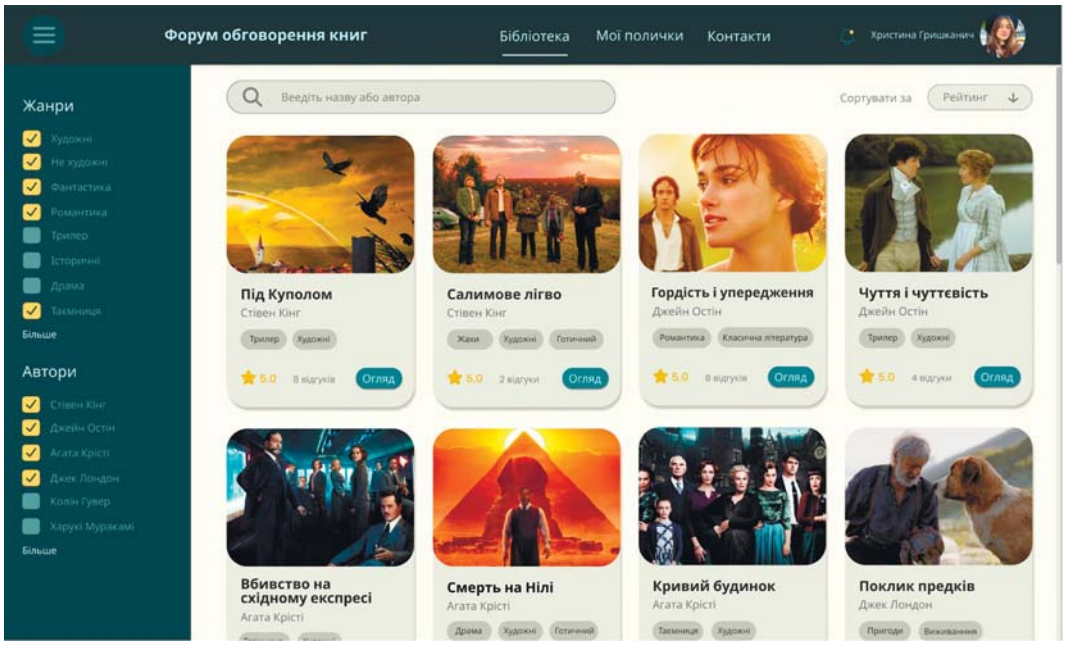


**Рис. 3.** Головна сторінка застосунку / Main page of the application

Дизайн інтерфейсу у стилі мінімалізму – жоден з елементів не перевантажує візуальне поле користувача. Картки для відображення інформації про книгу мають легку тінь, що дає змогу візуально виокремити їх на фоні світлої поверхні. Завдяки однаковим розмірам цих контейнерів і чіткому вертикальному та горизонтальному вирівнюванню досягають візуальної симетрії, що підвищує комфорт сприйняття. Поле пошуку "Enter name or author" має окреслену межу, яка однозначно сигналізує зону введення даних. Типографіка реалізована з акцентом на читабельність: наприклад, назви книг відображено жирним шрифтом, а додаткову інформацію (жанр, автор, кількість рецензій) подано звичайним, що створює ієрархію візуального сприйняття.

Окремої уваги заслуговують інтерактивні елементи користувацького інтерфейсу, доповнені іконками для візуальних підказок:

- іконка лупи у полі пошуку " Enter name or author ";
- стрілки вниз/вгору у випадному меню для сортування у різних порядках;
- зірочки для відображення рейтингу книг;
- гамбургер-меню для показу та приховування бокового меню.

Бічна панель фільтрації за жанрами та авторами також добре структурована: активні фільтри виділено кольоровими чекбоксами із жовтим маркером, що одразу привертає увагу та дає змогу швидко зорієнтуватись у вибраних параметрах.

Навігацію реалізовано так: у верхній частині інтерфейсу розміщене постійно закріплене горизонтальне меню, яке охоплює вкладки: Library, My Shelves, Contacts. Воно доступне на кожній сторінці застосунку, що гарантує швидке перемикання між основними розділами без потреби повертатися назад. Окрім цього, всі клікабельні елементи (наприклад, кнопки View, вкладки меню Library, My Shelves тощо) змінюють свій колір за наведення (hover), що забезпечує додатковий тактильний зв'язок і підсилює відчуття керованості інтерфейсом.

Для кольорової палітри користувацького інтерфейсу застосовано класичний принцип 60-30-10, який активно використовують в UI-дизайні, а саме:

- приблизно 60 % площини займає теплий відтінок слонової кістки, використаний як фон;
- 30 % становить темно-зелений – основний колір елементів меню, фільтрів, кнопок;
- 10 % припадає на акцентний жовтий, використаний для підсвічування важливих елементів, зокрема, рейтингу, активних чекбоксів тощо.

Варто зазначити, що в дизайні інших сторінок застосунку, зокрема, деталей книги, входу та полиць, також дотримано єдиних стилістичних і функціональних принципів. Так, усі кнопки мають однаковий стиль оформлення: фоновий колір, округлені кути, ефекти за наведення, що забезпечує послідовність взаємодії користувача з інтерфейсом.

На сторінках деталей книги (рис. 4) реалізовано чисту структуру з чітким акцентом на заголовок, рейтинг, рецензії та опис, що полегшує ознайомлення із вмістом.

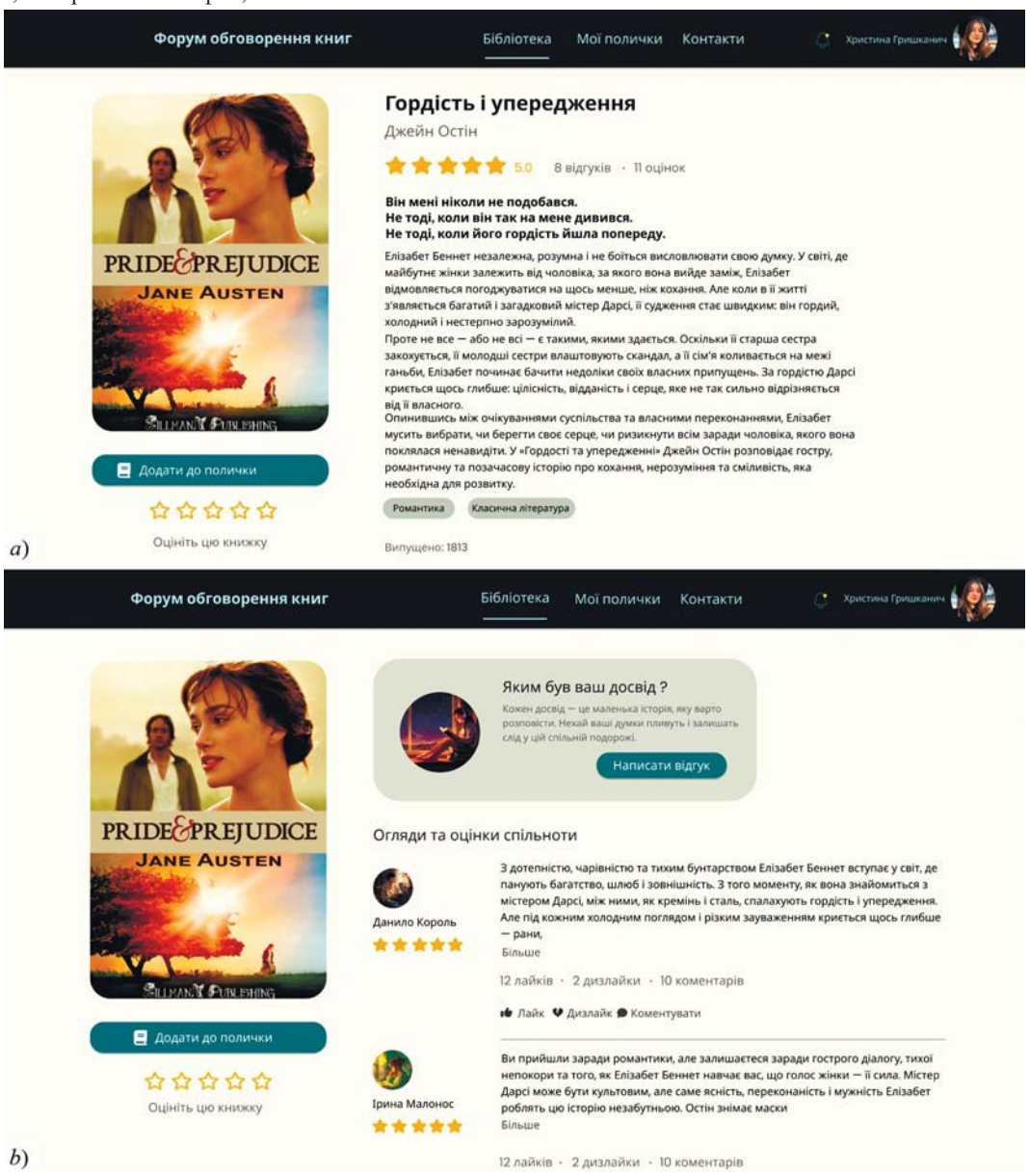


**Рис. 4.** Сторінки детальної інформації про книгу / Book details page

На сторінці входу користувацького інтерфейсу візуальне поле зосереджено на формі авторизації: поля введення мають ті ж самі стилі, що й на головній сторінці, що формує відчуття єдності в дизайні.

Особливої уваги заслуговує сторінка полиць (рис. 5). Тут реалізовано вкладки у вигляді дерева, що відображають розділення на "Прочитано" та "Плануються прочитати". Така структура є наближеною до фізичного уявлення книжкової полиці, що створює знайомий користувачеві ментальний образ і полегшує навігацію.

Отже, усі сторінки вебзастосунку побудовано на принципах узгодженості, передбачуваності та візуальної простоти, що сукупно забезпечує комфортний користувацький досвід.

Якість користувацького досвіду (UX) визначають не тільки дизайном користувацького інтерфейсу, а й про-

дуктивністю самого застосунку: швидкістю рендерингу, реакцією на взаємодії та стабільністю під час роботи. Висока затримка або повільне завантаження можуть негативно вплинути на залученість, задоволення та врешті-решт, на довіру користувача до самої системи. Саме тому у процесі реалізації вебзастосунку "Форум обговорення книг" приділено особливу увагу удосконаленню продуктивності його роботи.

У праці [19] проаналізовано сучасні методи удосконалення React-застосунків, що мають безпосередній вплив на UX. Дослідження базується на реальних випадках з електронної комерції та соціальних платформ і доводить, що продумана оптимізація знижує тривалість завантаження до 56 %, а споживання пам'яті до 40 %, що покращує показники метрик FCP (англ. *First Contentful Paint*) та TTI (англ. *Time to Interactive*).

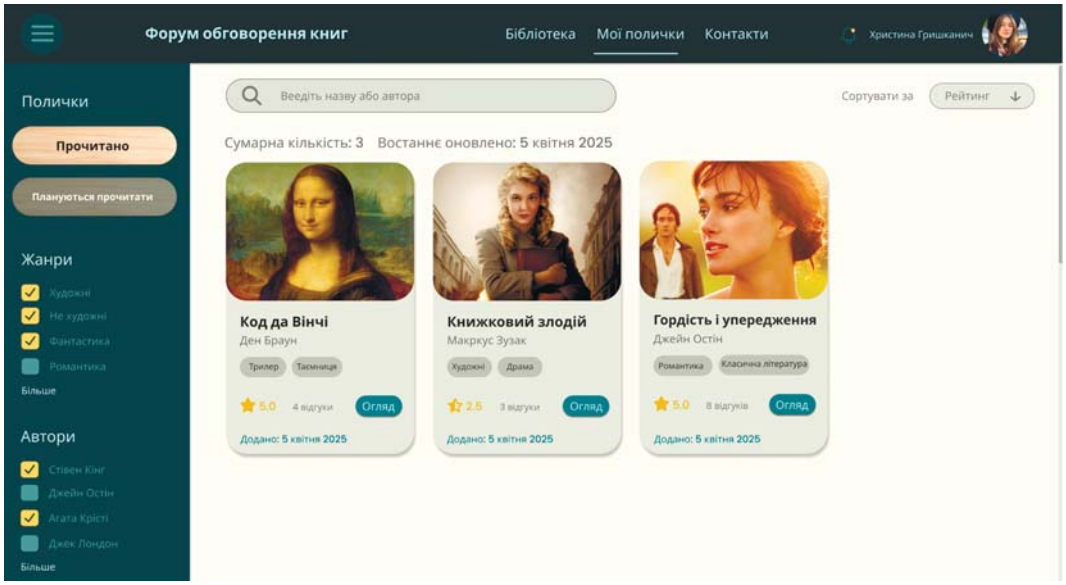


**Рис. 5.** Сторінка особистих віртуальних полиць користувача / User's personal virtual shelves page

Під час реалізації форуму обговорення змісту книг використано практичні рекомендації з цього дослідження:

- компонентну оптимізацію: застосовано React.memo для уникнення непотрібного повторного рендерингу за роботи зі списками книг, де дані не змінюються. Це дає змогу зменшити навантаження за фільтрації, сортування або зміни стану сторінки;
- оптимізацію хук-поведінки: в елементах керування, зокрема, в обробниках сортування, пошуку, перемикання між вкладками, використано useCallback, що дає змогу запам'ятовувати функції між рендерами й не створювати їх заново без потреби;
- віртуалізацію довгих списків: у майбутньому планують використання бібліотек типу react-window або react-virtualized для візуалізації великих списків рецензій або пошукових результатів, що дасть змогу підтримувати стабільну швидкодію за великої кількості елементів;
- лініве завантаження (React.lazy): деякі компоненти, наприклад, модальні вікна або редактор рецензій, завантажуються динамічно тільки за потреби. Це зменшує початковий обсяг групи зображень і пришвидшує перше завантаження інтерфейсу.

Під час розроблення вебзастосунку враховано висновки, наведені в аналітичному огляді [6], де автори наголошують на важливості технічного удосконалення як одного з ключових чинників покращення користувацького досвіду. Дослідження детально розглядає ефективність сучасних методів, зокрема, використання нових форматів зображень WebP і AVIF, а також мережевих протоколів HTTP/2 та HTTP/3, у пришвидшенні завантаження контенту, зменшенні навантаження на сервер та покращенні стабільності взаємодії. Зокрема, формати WebP та AVIF дають змогу значно зменшити вагу графічних файлів без втрати візуальної якості, що особливо критично для сайтів, де використовують численні обкладинки книг, ілюстрації та інтерфейсні іконки. У проєкті реалізовано підтримку формату WebP для зображень у списках книг і в картках деталей, що дало змогу зменшити тривалість завантаження зображень і покращити показник Largest Contentful Paint (LCP).

Окрім цього, серверну частину вебзастосунку налаштовано з урахуванням підтримки протоколу передавання даних HTTP/2, що забезпечує мультиплексування запитів (декілька файлів передаються одночасно через одне з'єднання) та ефективніше використання пропускної здатності. Це дає змогу зменшити затримку за завантаження сторінок та забезпечити плавніший досвід навігації.

Застосування зазначених рішень відповідає сучасним вимогам до продуктивності інтерфейсів і підтверджує висновки дослідження щодо їх впливу на підвищення задоволеності користувача, зниження показників відмов і загального покращення UX.

**Обговорення результатів дослідження.** У ширшому контексті роль цифрових платформ у трансформації культури письма розкриває автор у книзі [2]. У роботі проаналізовано, як різні етапи розвитку письмових технологій (від олівця та друкарської машинки до комп'ютера й мережі Інтернет) змінювали роль тексту у суспільстві та трансформували поведінку користувачів. Особливу увагу автор приділяє впливу застосунків (блоги, форуми, соціальні мережі) на залучення користувачів до створення контенту, обговорення текстів та формування спільнот. За результатами дослідження зроблено висновок, що цифрові технології сприяли демократизації процесу створення текстів і дали змогу широкій аудиторії брати участь у публічному обговоренні, однак поряд із цим постала нова проблема: зростання кількості користувачів й обсягів контенту призвело до складнощів в управлінні цими спільнотами. Зокрема, виникла потреба у розробленні нових підходів до керування, фільтрації контенту, боротьби з дезінформа-

цією та спамом, а також у забезпеченні етичних норм взаємодії користувачів. Автор зазначає, що без ефективного регулювання такі спільноти можуть втрачати свій культурний та освітній потенціали, а їхня комунікація перетворюється на хаотичний потік інформації без належної структурованості та змістовності.

У праці [11] окреслено проблему неефективного управління дискусійними форумами в онлайн-середовищах, що призводить до втрати фокусу обговорень та поширення дезінформації. Як вирішення, автори пропонують впровадження алгоритмів машинного навчання для автоматичної модерації – таких, що можуть виявляти відхилення від теми, неконструктивні повідомлення та підтримувати структурованість без постійної участі людини.

Автор праці [20] виступає на підтримку використання штучного інтелекту для персоналізації рекомендацій, наголошуючи, що такі інструменти спрощують пошук релевантної літератури, адаптуючи її під індивідуальні вподобання користувача. Це, водночас, підвищує зацікавленість, сприяє відкриттю нових авторів і розширює читацький досвід.

У праці [12] розглянуто переваги WebTransport – новітньої технології для двостороннього обміну даними на базі протоколу QUIC. Автор наголошує, що WebTransport забезпечує низьку затримку, підтримує одночасну передачу тексту, аудіо й відео та ефективно масштабується під велику кількість користувачів. У контексті форуму обговорення змісту книг ця технологія може слугувати інструментом у реальному часі для інтеграції чатів, стрімінгу презентацій або живого реагування на рецензії без втрати швидкодії та якості взаємодії.

У праці [16] наведено результати аналізу методів психологічної маніпуляції користувачами в UI/UX вебдизайні. Авори стверджують, що зростаюча залежність від вебінтерфейсів для онлайн-діяльності, такої як покупки, робота та спілкування, вказали на важливість етичного UI/UX-дизайну. Однак, зростання маніпулятивних методів вебдизайну створює актуальну проблему. Ці методи непомітно впливають на поведінку користувачів так, що це може призвести до порушення конфіденційності, небажаних покупок та інших негативних наслідків. Вони також можуть використовувати психологічні упередження користувачів, що ускладнює розпізнавання маніпуляцій. У своєму дослідженні автори зібрали, детально описали та систематизували методи маніпуляцій у вебдизайні UI/UX. Кожен метод детально описано та організовано за 10 різними категоріями. Також у цій роботі детально описано, які методи вимагають дій користувача або усвідомлення користувача, що призводить до ненавмисних дій або примушує їх до небажаних рішень шляхом маніпуляцій. Також було обговорено варіанти використання та можливі контрзаходи. Цей внесок слугує основою для майбутнього оцінювання впливу цих методів та розвитку етичних практик вебдизайну.

У роботі [7] описано UI-UX дизайн з використанням методу орієнтованого на користувача дизайну UCD (англ. *User-Centred Design*). Автори вважають, що оскільки компанії розширюють свої онлайн-пропозиції, попит на UI-UX та цифрові послуги зростає. Як внутрішні команди, так і агентства будуть під тиском, щоб пропонувати послуги вчасно, оскільки загальна кількість користувачів мережі Інтернет у всьому світі зросла на 350 мільйон з квітня 2022 року (We are Social). У своєму дослідженні для оброблення всього прототипування було використано метод дизайну, орієнтованого на користувача (UCD). Потік користувачів також обговорюється на основі трьох рівнів (екран запуску, створення та обліковий запис, головний екран). Результатом тестування інтерфейсу для зразка прототипування були елементи дизайну застосунку, де 4 бали були найпоширенішою відповіддю за 5-бальною шкалою, головна сторінка меню та її враження – 5 балів, а загальний UI-UX додатку – 5 балів.

У роботі [5] описано дослідження асинхронного форуму для електронного навчання. Автори спираються на теорії соціального конструктивізму та конективізму, які покладено в основу моделі форуму. Для аналізу рівня знань, взаємодії та залученості користувачів використано контент-аналіз, кластерний аналіз і соціальний мережевий аналіз. Дослідження вирізняється глибоким підходом до вивчення когнітивних і соціальних аспектів участі у форумах. Зокрема, встановлено статистично значущий зв'язок між рівнем побудови знань і показниками взаємодії, такими як кількість отриманих повідомлень (вхідна центральність) та роль учасника як посередника у комунікації (центральність посередництва): активні користувачі, які частіше комунікували з іншими та сприяли обміну інформацією, досягали вищих когнітивних результатів. Однак, система виявилася надто складною технічно й більше підходить для дослідницьких цілей, ніж для зручного використання звичайними користувачами.

У роботі [17] представлено чат і відеозастосунок, розроблений із використанням технології WebRTC для прямої передачі даних між браузерами та топології mesh, що передбачає взаємне об'єднання всіх користувачів у локальній мережі. Застосунок дає змогу обмінюватися повідомленнями, здійснювати відеодзвінки, демонструвати екран і створювати кімнати без потреби в додатковому програмному забезпеченні. Перевага дослідження – застосування сучасних вебтехнологій (WebRTC, React, Node.js) для забезпечення прямої комунікації. Водночас недоліком є відсутність уваги до масштабованості, безпеки й навантаження за великої кількості користувачів, що обмежує можливість широкого впровадження, зокрема в публічних цифрових платформах, таких як форуми для обговорення змісту книг.

Узагальнюючи обговорення результатів дослідження, можна зробити висновок, що жодна з проаналізованих систем не забезпечує повного вирішення актуальних проблем, пов'язаних із функціонуванням форумів для обговорення змісту книг. Це стосується і технічних аспектів (масштабованості, швидкодії та удосконалення), і якості організації дискусій, зокрема рівня структурованості, змістовного наповнення та релевантності інформації. Доцільність розробленого застосунку підтверджено його стабільною роботою за зростання кількості користувачів, ефективним опрацюванням запитів й забезпеченням якісної організації простору для змістовного та цілеспрямованого обговорення літератури.

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

*Наукова новизна отриманих результатів дослідження* – удосконалено метод розроблення інтерактивного середовища користувачів і формування їхнього пози-

тивного досвіду через забезпечення гнучкого управління контентом та збільшення їхнього залучення до тематичних літературних дискусій.

*Практична значущість результатів дослідження* – унаслідок масштабованості та швидкодії програмного забезпечення, які досягаються за рахунок інтеграції сучасних принципів UI/UX-дизайну з технічними методами фронтенд-удосконалення та віртуалізації компонентів, результати цього дослідження можна використати для створення інших функціональних вебзастосунків-форумів, призначених для обговорення, зокрема сайтів новин з можливістю оцінювання й коментування, освітніх платформ для взаємодії між учасниками навчального процесу та соціальних мереж з фокусом на тематичні дискусії.

## Висновки / Conclusions

Спроєктовано та реалізовано вебзастосунок для утворення спільноти книжкового середовища, який дає змогу проводити обговорення змісту книг та обмінюватися думками стосовно прочитаного, що сприятиме самовираженню його користувачів, полегшить пошук і вибір літератури за жанрами, авторами чи тематиками. За результатами проведеного дослідження можна зробити такі основні висновки:

1. Проаналізовано останні дослідження та публікації, встановлено, що більшість сучасних онлайн-платформ, які популяризують книги, зосереджені переважно на комерційних функціях або доступі до електронних видань, а можливості для глибокого обговорення змісту залишаються обмеженими. Виявлено дефіцит спеціалізованих середовищ для обміну літературними враженнями, що обґрунтувало потребу у створенні окремого вебзастосунку з відповідним функціональним спрямуванням.
2. Спроєктовано ефективну архітектуру вебзастосунку на підставі набору технологій MERN (MongoDB, Express.js, React.js, Node.js). Побудовано трирівневу клієнт-серверну модель з розподілом відповідальностей, що забезпечило масштабованість, гнучкість модифікації та можливість розширення. Реалізовано підтримку обміну даними в реальному часі та зручну взаємодію між компонентами системи.
3. Реалізовано функціональну частину платформи, що охоплює такі можливості: автентифікацію, написання рецензій, оцінювання рецензій, додавання коментарів, оцінювання книг, створення віртуальних книжкових полиць. Для ефективного обміну даними впроваджено архітектурний стиль REST API. Забезпечено безпечне оброблення запитів, модульну структуру проєкту та повторне використання компонентів.
4. Забезпечено якісний користувацький досвід та розроблено інтерфейс, що відповідає сучасним вимогам до вебдизайну. Інтерфейс побудовано з використанням Material UI, що дало змогу створити візуально привабливе середовище. Застосовано SCSS для зручного управління стилями та адаптивності. Проведено оптимізацію фронтенду за допомогою сучасних методів, що сприяло швидшому завантаженню інтерфейсу, зниженню навантаження на клієнта та забезпеченню стабільної роботи застосунку.

## References


1. Bada, A. B. (2021). Performance Optimization of Web-Based Application. International Journal of Computer Science Engineering (IJCSE). URL: https://www.researchgate.net/publication/353001318_Performance_Optimization_of_Web-Based_Application
2. Baron, D. (2009). A Better Pencil: Readers, Writers, and the Digital Revolution. – Oxford University Press, 272 p. URL: https://books.google.com.ua/books?hl=uk&lr=&id=VWzgtZMJCwgC
3. Bawane, M., Gawande, I., Joshi, V., Nikam, R., & Bachwani, S. A (2022). Review on Technologies used in MERN Stack. *International Journal for Research in Applied Science and Engineering Technology* (IJRASET), Vol. 10, Issue 1, 2321–9653. URL: https://www.ijraset.com
4. Dimitrov, S., Zamal, F., Piper, A., & Ruths, D. (2015). Goodreads Versus Amazon: The Effect of Decoupling Book Reviewing And Book Selling. *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 9, No. 1, 602–605. https://doi.org/10.1609/icwsm.v9i1.14662
5. Durairaj, K., Assarnurkutty, S. J. b., Ng, K. T., & Sabri, W. N. A. b. M. (2024). Development of a Framework for an Asynchronous Discussion Forum through an E-Learning Platform. *Dinamika Ilmu*, Vol. 14, No. 2. https://doi.org/10.30595/dinamika.v14i2.15274
6. Ekpobimi, H., Kandekere, R. C., & Fasanmade, A. (2024). Conceptual framework for enhancing front-end web performance: Strategies and best practices. *Global Journal of Advanced Research and Reviews*, Vol. 2, No. 1. URL: https://www.researchgate.net/publication/374979541
7. Goel, G., Tanwar, P., & Sharma, S. (2022). UI-UX Design Using User Centred Design (UCD) Method. In: 2022 *International Conference on Computer Communication and Informatics* (ICCCI), Coimbatore, India, pp. 1–8. https://doi.org/10.1109/ICCCI54379.2022.9740997
8. Hasan, T. I., Silalahi, C. I., Rumagit, R. Y., & Pratama, G. D. (2024). UI/UX Design Impact on E-Commerce Attracting Users. *Procedia Computer Science*, Vol. 245, 1075–1082. https://doi.org/10.1016/j.procs.2024.10.336
9. Hasan, Z., & Gope, R. (2013). Dynamics of User Experience (UX). *International Journal of Computer Applications*, Vol. 81, No. 16, 18–24. https://doi.org/10.5120/14207-2443
10. Kamisetty, A., Narsina, D., Rodriguez, M., Kothapalli, S., & Gummadi, J. C. (2022). Microservices vs. Monoliths: Comparative Analysis for Scalable Software Architecture Design. I*nternational Journal of Scientific Research in Computer Science, Engineering and Information Technology*. URL: https://ijsrcseit.com/CSEIT228143
11. Kilinc, H., & Altınpulluk, H. (2021). Use of Discussion Forums in Online Learning Environments. *International Journal of Higher Education Pedagogies*, Vol. 2, No. 1, 1–7. https://doi.org/10.33422/ijhep.v2i1.25
12. Mohapatra, L. (2023). *WebTransport: The Future of Real-Time Communication, Bridging the Gap Beyond WebRTC & Websockets*. URL: https://medium.com/@lakinmohapatra/webtransport-the-future-of-real-time-communication-bridging-the-gap-beyond-webrtc-websockets-99cf278e6aa0
13. Onyema, E. M., Deborah, E. C., Alsayed, A. O., Naveed, Q. N., & Sanober, S. (2019). Online Discussion Forum as a Tool for Interactive Learning and Communication. *International Journal of Recent Technology and Engineering* (IJRTE), Vol. 8, No. 4, 4852–4859. https://doi.org/10.35940/ijrte.D8062.118419
14. Pendry, L. F., & Salvatore, J. (2015). Individual and social benefits of online discussion forums. *Computers in Human Behavior*, Vol. 50, 211–220. https://doi.org/10.1016/j.chb.2015.03.067
15. Prayogi, A. A., Niswar, M., Indrabayu, & Rijal, M. (2020). Design and Implementation of REST API for Academic Information System. *IOP Conference Series: Materials Science and Engineering*, Vol. 875. https://doi.org/10.1088/1757-899X/875/1/012047
16. Riaz, R., Vasconcelos, A., & Pinto, P. (2024). An Overview of User Psychological Manipulation Techniques in UI/UX Web Design. In: 2024 *Cyber Awareness and Research Symposium* (CARS), Grand Forks, ND, USA, pp. 1–6. https://doi.org/10.1109/CARS61786.2024.10778887
17. Saeed, M. A., & Edan, N. M. (2022). Design and Implement Video and Chat Application Using Mesh Network. *International Journal of Applied Sciences and Technology*, Vol. 1, No. 12. Rimar Academy, Istanbul, Turkey. URL: https://www.minarjour-

nal.com/dergi/design-and-implement-video-and-chat-application-using-mesh-network20230130033253.pdf
18. Shahwan, Jamil S. (2023). The Impact of Social Media on Literature. *Arab World English Journal* (*AWEJ*). *Special Issue on Communication and Language in Virtual Spaces*, January 20 p. https://doi.org/10.2139/ssrn.4350373
19. Veeri, V. (2024). Performance Optimization Techniques in React Applications: A Comprehensive Analysis. *International Journal of Research In Computer Applications and Information Technology* (IJRCAIT), Vol. 7, Issue 2, 1165–1177. https://doi.org/10.5281/zenodo.14146734
20. Zul, M. (2025). The Future of Reading: How Technology, Society, and Innovation Are Redefining Literacy. PublishingState.com, URL: https://publishingstate.com/future-of-reading/

***Kh. O. Hryshkanych, L. M. Zhuravchak, T. O. Mukha***
*Lviv Polytechnic National University, Lviv, Ukraine*

## DEVELOPMENT OF A WEB APPLICATION USING MERN STACK TO CREATE BOOK ENVIRONMENT COMMUNITY

The paper deals with the developing a web application using MERN stack in creating book environment community. In the course of research, most modern digital platforms, particularly marketplaces and electronic libraries, are found to provide an insufficient level of support for reader interaction and in-depth discussion of literary content. It has been identified that review functionality in such services is of secondary importance or limited to evaluating the technical characteristics of publications, which complicates the formation of stable reading communities. The need for the creation of a specialized environment focused on the exchange of opinions between users, reflection on the reading experience, and the development of intellectual dialogue has been confirmed. In view of this, a web application titled "Book Discussion Forum" has been developed, with the main objective of providing a convenient platform for evaluating, reviewing, and discussing book content. The system architecture has been implemented according to a classical client-server model, using MERN stack (MongoDB, Express.js, React.js, Node.js) in combination with the TypeScript programming language, which ensured scalability, type safety, development speed, and flexibility for future system expansion. The impact of the REST architectural approach on maintainability and effective client-server interaction has been assessed. The backend of the project has been implemented as a service-oriented structure, where each controller is responsible for a specific part of the business logic, in particular, managing books, reviews, ratings, shelves, and user authentication. The structure of the interface is described as component-based, built with React and using Redux and Context API tools for state management, with a modular organization featuring a Layout component that dynamically loads content based on the current route. Special attention has been given to the user experience design, including the application of modern UI/UX principles such as visual hierarchy, colour harmony, responsiveness, and logical page structure. The impact of implemented optimizations, including React.memo, useCallback, React.lazy, WebP support, and HTTP/2 on system performance has been evaluated. A reduction in load time, improved interface responsiveness, and a decrease in user bounce rate have been confirmed as well. Future research perspectives include the implementation of automated moderation mechanisms, inappropriate content filtering, and the development of personalized recommendation systems based on artificial intelligence algorithms to enhance user engagement and the quality of reading interaction.

**Keywords:** book discussion; client-server architecture; user experience; UI/UX design; digital reviewing.