МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: **«РОЗРОБКА ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ ОРГАНІЗАЦІЇ ОНЛАЙН-КЛУБУ ЧИТАЧІВ З МОЖЛИВІСТЮ ОБМІНУ РЕЦЕНЗІЯМИ ТА ОБГОВОРЕННЯМИ»**

Виконав: студент 4 курсу, групи 6.1222
спеціальності 122 комп'ютерні науки
(шифр і назва спеціальності)
освітньої програми комп'ютерні науки
(назва освітньої програми)
В. В. Тварковський
(ініціали та прізвище)
Керівник доцент кафедри комп'ютерних наук, к.т.н., доцент Решевська К. С.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)
Рецензент завідувач кафедри фундаментальної та прикладної математики, професор, д.т.н. Гребенюк С.М.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя 2026

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра комп'ютерних наук

Рівень вищої освіти бакалавр

Спеціальність 122 комп'ютерні науки
(шифр і назва)

Освітня програма комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри комп'ютерних наук, к.пед.н., доцент

_____ Пшенична О.С.
(підпис)

" 08 " грудня 2025 р.

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Тварковському Владиславу Вікторовичу
(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка програмного комплексу для організації онлайн-клубу читачів з можливістю обміну рецензіями та обговореннями

керівник роботи Решевська Катерина Сергіївна, к.т.н., доцент
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 02 » грудня 2025 року № 1741-с

2. Строк подання студентом роботи 04.06.2026

3. Вихідні дані до роботи 1. Постановка задачі.
2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Теоретичні основи розробки онлайн-спільнот та систем обміну контентом.
2. Проєктування програмного комплексу онлайн-клубу читачів.
3. Реалізація та тестування програмного комплексу онлайн-клубу читачів.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 05.12.2025

К А Л Е Н Д А Р Н И Й П Л А Н

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	19.02.2026	
2.	Збір вихідних даних.	24.03.2026	
3.	Обробка методичних та теоретичних	28.03.2026	
	джерел.		
4.	Розробка першого та другого розділу.	23.04.2026	
5.	Розробка третього розділу.	07.05.2026	
6.	Оформлення та нормоконтроль	04.06.2026	
	кваліфікаційної роботи бакалавра.		
7.	Захист кваліфікаційної роботи.	20.06.2026	

Студент _____ В. В. Тварковський
(підпис) (ініціали та прізвище)

Керівник роботи _____ К. С. Решевська
(підпис) (ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____ О.Г.Спиця
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка програмного комплексу для організації онлайн-клубу читачів з можливістю обміну рецензіями та обговореннями»: 76 с., 17 рис., 12 табл., 20 джерел, 2 додатків.

АВТЕНТИФІКАЦІЯ, БАЗА ДАНИХ, ВЕБЗАСТОСУНОК, КОРИСТУВАЦЬКИЙ ІНТЕРФЕЙС, ОБГОВОРЕННЯ, ОНЛАЙН-СПІЛЬНОТИ, РЕЦЕНЗУВАННЯ, СЕРВЕРНІ ДІЇ, СИСТЕМА РОЛЕЙ, УПРАВЛІННЯ КОНТЕНТОМ, NEXT.JS.

Об’єктом дослідження є процеси організації онлайн-спільнот читачів та обміну літературними рецензіями в цифровому середовищі.

Предметом дослідження є методи та засоби розробки програмного комплексу для підтримки функціонування онлайн-клубу читачів.

Метою кваліфікаційної роботи є розробка програмного комплексу для організації онлайн-клубу читачів з можливістю обміну рецензіями та обговореннями літературних творів.

Методи дослідження включають системний і порівняльний аналіз онлайн-спільнот, методи проєктування реляційних баз даних, об’єктно-орієнтоване та компонентне проєктування вебзастосунків, а також функціональне тестування програмного забезпечення.

У роботі проаналізовано онлайн-спільноти читачів і визначено вимоги до системи. Спроєктовано архітектуру та базу даних, що забезпечує збереження інформації про користувачів, книги, рецензії та коментарі. Реалізовано основні модулі: реєстрацію й автентифікацію, каталог книг із пошуком і фільтрацією, рецензії з перерахунком рейтингу та систему обговорень. Окремо реалізовано адміністративний модуль із рольовим розмежуванням доступу. Тестування підтвердило коректність роботи системи та відповідність вимогам.

SUMMARY

Bachelor's Thesis «Development of A Software System for Organizing an Online Readers' Club with the Ability to Exchange Reviews and Participate in Discussions»: 76 pages, 17 figures, 12 tables, 20 sources, 2 appendices.

AUTHENTICATION, DATABASE, WEB APPLICATION, USER INTERFACE, DISCUSSIONS, ONLINE COMMUNITIES, REVIEWING, SERVER ACTIONS, ROLE SYSTEM, CONTENT MANAGEMENT, NEXT.JS.

The object of research is the processes of organizing online reading communities and exchanging literary reviews in a digital environment.

The subject of research is the methods and tools for developing a software system to support the functioning of an online book club.

The aim of the thesis is to develop a software system for organizing an online book club with the ability to exchange reviews and discuss literary works.

The research methods include systematic and comparative analysis of online communities, relational database design methods, object-oriented and component-based web application design, and functional software testing methods.

The work analyzes online reading communities and defines system requirements. The system architecture and database structure were designed to store information about users, books, reviews, and comments. The main modules were implemented, including user registration and authentication, a book catalog with search and filtering capabilities, a review system with dynamic rating calculation, and a discussion module. An administrative module with role-based access control was also implemented. Testing confirmed the correct operation of the system and compliance with the defined requirements.

ЗМІСТ

Завдання на кваліфікаційну роботу……..2

Реферат 4

Summary 5

Вступ..... 7

1 Теоретичні основи розробки онлайн-спільнот та систем обміну контентом 10

1.1 Поняття та особливості онлайн-спільнот читачів 10

1.2 Аналіз сучасних платформ для обміну рецензіями та обговореннями…..... 12

1.3 Визначення вимог до програмного комплексу онлайн-клубу читачів…. 16

2 Проєктування програмного комплексу онлайн-клубу читачів 20

2.1 Аналіз предметної області 20

2.2 Проєктування архітектури системи та бази даних 23

2.3 Проєктування користувацького інтерфейсу та вибір технологій розробки.. 32

3 Реалізація та тестування програмного комплексу онлайн-клубу читачів ... 39

3.1 Реалізація основного функціоналу системи 39

3.2 Розробка модулів рецензій, обговорень та користувацьких профілів…. 43

3.3 Тестування системи та аналіз результатів роботи 47

3.4 Інструкція користувача..... 50

Висновки 53

Перелік посилань..... 56

Додаток А Реалізація основного функціоналу системи..... 58

Додаток Б Інтерфейс вебзастосунку 71

ВСТУП

У сучасних умовах цифрової трансформації суспільства значно зростає роль онлайн-платформ, які забезпечують комунікацію користувачів, обмін інформацією та формування тематичних спільнот. Особливого розвитку набули онлайн-спільноти за інтересами, зокрема клуби читачів, які поєднують функції соціальної мережі, бібліотечного каталогу та платформи для обговорення літературних творів. Такі системи сприяють популяризації читання, розвитку критичного мислення та формуванню активного читацького середовища.

Актуальність теми зумовлена зростанням обсягів цифрового контенту та необхідністю створення спеціалізованих програмних комплексів, які дозволяють користувачам ефективно організовувати обмін рецензіями, брати участь у дискусіях та формувати персоналізовані читацькі добірки. Існуючі платформи, такі як Goodreads або локальні аналоги, не завжди повною мірою враховують потреби користувачів щодо зручності інтерфейсу, локалізації, персоналізації та гнучкості взаємодії, що створює потребу у розробці нових або удосконаленні існуючих рішень.

Метою кваліфікаційної роботи є розробка програмного комплексу для організації онлайн-клубу читачів з можливістю обміну рецензіями та обговореннями літературних творів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

– проаналізувати сучасні онлайн-спільноти читачів та платформи для обміну рецензіями;

– дослідити основні вимоги до подібних програмних систем;

– виконати аналіз предметної області та сформувати функціональні вимоги до системи;

– спроєктувати архітектуру програмного комплексу та структуру бази даних;

– розробити користувацький інтерфейс і обрати технології реалізації;

– реалізувати основні модулі системи, зокрема функції рецензування та обговорень;

– провести тестування програмного комплексу та проаналізувати отримані результати.

Об’єктом дослідження є процеси організації онлайн-спільнот читачів та обміну літературними рецензіями в цифровому середовищі.

Предметом дослідження є методи та засоби розробки програмного комплексу для підтримки функціонування онлайн-клубу читачів.

У процесі виконання роботи було враховано сучасні підходи до проєктування вебзастосунків, принципи побудови інтерактивних онлайн-спільнот та методи організації користувацького контенту. Серед основних напрямів розвитку подібних систем варто відзначити інтеграцію соціальних функцій, механізмів персоналізації та рекомендаційних алгоритмів, що дозволяють підвищити залученість користувачів.

Теоретичною основою дослідження стали наукові праці з розробки вебзастосунків, побудови інформаційних систем, UX/UI-дизайну та організації соціальних платформ. Також використано аналіз існуючих рішень у сфері онлайн-читання та літературних спільнот.

Методами дослідження є: системний аналіз для вивчення предметної області, порівняльний аналіз існуючих платформ, методи проєктування програмних систем, методи структурного та об’єктно-орієнтованого програмування, а також методи тестування програмного забезпечення.

Практичне значення роботи полягає у можливості використання розробленого програмного комплексу як готової платформи для створення онлайн-клубів читачів, навчальних спільнот або тематичних літературних сервісів. Запропоновані рішення можуть бути застосовані для розробки освітніх вебзастосунків, соціальних платформ та сервісів обміну контентом.

Наукова новизна роботи полягає у поєднанні базових механізмів соціальної взаємодії користувачів із функціоналом структурованого обміну

рецензіями та обговореннями, що дозволяє підвищити ефективність організації читацьких спільнот та покращити якість взаємодії між користувачами.

Отримані результати полягають у розробці концептуальної моделі програмного комплексу онлайн-клубу читачів, проєктуванні його архітектури та реалізації основних функціональних модулів, що забезпечують створення, перегляд та обговорення рецензій. Проведене тестування підтвердило працездатність системи та відповідність її основним функціональним вимогам.

1 ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ ОНЛАЙН-СПІЛЬНОТ ТА СИСТЕМ ОБМІНУ КОНТЕНТОМ

1.1 Поняття та особливості онлайн-спільнот читачів

Онлайн-спільноти читачів є важливим різновидом сучасних віртуальних соціальних утворень, які функціонують у цифровому середовищі та об’єднують користувачів навколо спільного інтересу до читання, обговорення літератури та обміну читацьким досвідом. У таких спільнотах основним чинником об’єднання виступає не територіальна або професійна належність, а спільна культурна практика – читання, що формує особливий тип соціальної взаємодії між користувачами. Онлайн-спільнота читачів у цьому контексті виступає як простір, де відбувається не лише обмін інформацією про книги, але й формування колективних інтерпретацій літературних творів, оцінок та рекомендацій, що впливають на читацькі вподобання учасників [1].

У ширшому соціологічному та комунікаційному значенні інтернет-спільноти розглядаються як віртуальні групи людей, які взаємодіють через мережу Інтернет, створюючи, поширюючи та обговорюючи інформаційний контент. Вони формуються на основі спільних інтересів і характеризуються регулярною взаємодією між учасниками, що може відбуватися у вигляді коментарів, повідомлень, публікацій або мультимедійного контенту. Саме тому онлайн-спільноти читачів можна визначити як спеціалізований підвид інтернет-спільнот, у якому спільний інтерес конкретизується літературною діяльністю та практикою читання, що робить такі платформи одночасно соціальними і культурно-освітніми системами.

Подальший розвиток онлайн-спільнот читачів тісно пов’язаний із цифровізацією культури та появою спеціалізованих платформ для читання й обговорення книг. Такі платформи створюють умови для формування нових моделей комунікації, де користувачі можуть не лише читати літературні твори,

але й одразу ділитися враженнями, оцінювати прочитане та брати участь у дискусіях. Це формує динамічне інформаційне середовище, у якому знання про літературу постійно оновлюється та доповнюється за рахунок користувацького контенту.

Однією з ключових особливостей онлайн-спільнот читачів є їхня віртуальна природа, яка дозволяє долати географічні та соціальні бар'єри. Завдяки цьому учасники з різних країн можуть взаємодіяти в межах одного інформаційного простору, обмінюватися думками та формувати спільні читацькі практики незалежно від фізичної відстані. Така глобальність є однією з головних переваг цифрових спільнот і значно розширює можливості культурної взаємодії [1; 2].

Важливою характеристикою таких спільнот є активна роль користувачів у створенні контенту. На відміну від традиційних медіа, де інформація подається односторонньо, онлайн-спільноти читачів базуються на принципі користувацької участі. Кожен учасник може створювати рецензії, залишати коментарі, формувати рейтинги книг та впливати на загальну інформаційну картину спільноти. Така модель взаємодії перетворює користувачів із пасивних споживачів контенту на активних співтворців інформаційного середовища.

Окрему роль у функціонуванні онлайн-спільнот читачів відіграють соціальні механізми взаємодії, які часто описуються як ритуали участі. До них належать реєстрація в системі, створення персонального профілю читача, ведення списків прочитаних книг, написання рецензій та участь у тематичних обговореннях. Ці дії формують відчуття приналежності до спільноти та сприяють розвитку соціальних зв'язків між учасниками, навіть якщо вони ніколи не зустрічаються в реальному житті [1; 3].

Ще однією суттєвою особливістю є інклюзивність таких платформ. Онлайн-спільноти читачів зазвичай не мають жорсткої ієрархії, що дозволяє кожному учаснику вільно висловлювати власну думку незалежно від його соціального статусу, досвіду чи професійної підготовки. Це сприяє формуванню відкритого середовища, де різні точки зору можуть співіснувати та взаємно

доповнювати одна одну, що підвищує якість дискусій та глибину аналізу літературних творів [2].

Не менш важливою характеристикою є мультимодальність комунікації, яка проявляється у використанні різних форматів взаємодії між користувачами. Окрім текстових повідомлень і форумів, сучасні платформи підтримують блоги, рейтингові системи, а також мультимедійні елементи, що дозволяє користувачам більш повно висловлювати свої думки та емоції щодо прочитаних творів. Це робить комунікацію більш гнучкою та адаптованою до різних стилів взаємодії.

Важливим аспектом функціонування таких спільнот є також наявність правил поведінки та модерації, які регулюють взаємодію між учасниками. Вони спрямовані на підтримання конструктивної атмосфери, запобігання конфліктам і забезпечення дотримання етичних норм спілкування [1; 3]. Таким чином, навіть у відкритому цифровому середовищі забезпечується певний рівень організованості та стабільності соціальної взаємодії.

Отже, онлайн-спільноти читачів можна розглядати як складні соціально-цифрові системи, що поєднують елементи комунікації, культурного обміну та інформаційних технологій. Вони виконують важливу роль у формуванні сучасної читацької культури, сприяють розвитку критичного мислення та забезпечують ефективний обмін літературним досвідом між користувачами з різних країн і соціальних груп.

1.2 Аналіз сучасних платформ для обміну рецензіями та обговореннями

Розвиток цифрових технологій та зростаюча популярність соціальних мереж докорінно змінили культуру читання. Сучасні читачі вже не обмежуються самостійним осмисленням прочитаного – вони активно діляться враженнями, обговорюють сюжетні лінії та рекомендують твори своєму оточенню. Це призвело до появи спеціалізованих платформ, орієнтованих на читацькі

спільноти, кожна з яких пропонує унікальний набір функцій для обміну рецензіями й дискусій.

Найбільшою та найвпливовішою платформою у цій сфері залишається Goodreads (goodreads.com), заснована у 2007 році. Після придбання компанією Amazon у 2013 році платформа стала ключовим гравцем книжкового ринку. Станом на 2024 рік Goodreads налічує понад 150 мільйонів зареєстрованих користувачів у всьому світі [4], а її рекомендаційний двигун аналізує понад 20 мільярдів точок даних. Аудиторія платформи демографічно неоднорідна: 61 % користувачів – жінки, а найчисленнішу вікову групу складають особи 25–34 роки (близько 30 % аудиторії), США – ключовий ринок із часткою 48 % трафіку [4].

Основний функціонал Goodreads охоплює ведення персональних книжкових полиць («прочитано», «читаю зараз», «хочу прочитати»), п’ятизіркову систему оцінювання, написання текстових рецензій, участь у тематичних групах-обговореннях, щорічний читацький виклик (Reading Challenge) та підписку на активність друзів. Однак попри масштаб платформи, користувачі та аналітики фіксують низку системних недоліків. Дослідники та публіцисти відзначають, що інтерфейс Goodreads практично не оновлювався з початку 2010-х і за сучасними стандартами UX виглядає застарілим [5]. Рекомендаційний алгоритм платформи також зазнає критики: замість персоналізованих порад на основі смаків конкретного читача він схильний просувати комерційно вигідні для Amazon видання [5]. Додаткова проблема – так зване «review bombing»: масштабне виставляння однозіркових оцінок книгам, які ще не вийшли, або авторам, що викликали соціальний резонанс [6]. Все це свідчить про те, що домінуюча позиція Goodreads не є синонімом якості клієнтського досвіду.

Прямою відповіддю на ці недоліки стала платформа The StoryGraph (thestorygraph.com), заснована у 2019 році розробником Надією Одунайо. Ця незалежна платформа стрімко зростає: якщо у квітні 2025 року вона подолала позначку в 4 мільйони користувачів, то вже у січні 2026 року їх кількість досягла 5 мільйонів [7]. The StoryGraph позиціонується як «розумна» альтернатива

Goodreads з акцентом на аналітику читання та персоналізацію. Ключова відмінність – машиннонавчальний рекомендаційний двигун, що враховує не лише жанрові вподобання, а й настрій читача, темп та емоційний тон твору (через систему «mood-based filters»)

Серед конкурентних переваг The StoryGraph варто виділити: можливість виставляти оцінки з кроком у чверть зірки (замість лише цілих значень на Goodreads); детальну статистику читання у вигляді діаграм і графіків; щорічний «Reading Wrap-up»; систему попереджень про контент (content warnings); ведення читацького щоденника з приватними нотатками [8]. Платна підписка The StoryGraph Plus надає розширену аналітику і право голосу щодо майбутніх функцій. Водночас соціальний функціонал платформи помітно скромніший за Goodreads – функція коментування відгуків та повноцінні книжкові клуби перебувають у стадії розробки [8].

Принципово іншу модель пропонує LibraryThing (librarything.com), засноване у 2005 році. Ця платформа від початку орієнтувалася не на соціальну мережу, а на каталогізацію – ретельний облік особистих книжкових колекцій. База даних LibraryThing охоплює понад 140 мільйонів бібліографічних записів з таких джерел, як Library of Congress та Amazon, що робить її особливо цінною для бібліофілів, колекціонерів та дослідників [9]. Унікальна функція «Common Knowledge» дозволяє учасникам спільноти наповнювати сторінки книг фактичними метаданими: порядком у серії, персонажами, нагородами, анотаціями. Рекомендаційна система LibraryThing працює за принципом перетину бібліотек – порівняння колекції конкретного користувача з колекціями інших читачів для виявлення спільних інтересів [9]. Основний недолік – менша кількість соціальних функцій та менш сучасний мобільний додаток порівняно з конкурентами.

Платформа BookLikes (booklikes.com) є прикладом гібридного підходу, що поєднує блог-сервіс і соціальну мережу для читачів. Вона дозволяє створювати персоналізовані блоги з темами та макетами, ділитися рецензіями у вільній формі, стежити за іншими читачами та авторами [10]. Проте платформа має

обмежену аудиторію порівняно з лідерами ринку, а окремі користувачі повідомляють про технічні проблеми зі стабільністю та оновленнями. Відсутність продвинутого рекомендаційного алгоритму та застарілий інтерфейс суттєво знижують конкурентоспроможність BookLikes.

Окремої уваги заслуговує BookWyrm (joinbookwyrm.com) – відкрита децентралізована соціальна мережа для читачів, побудована на протоколі ActivityPub (тому ж, що використовують Mastodon та PeerTube). Платформа функціонує як федерація незалежних серверів-інстансів, між якими можлива безперешкодна взаємодія [11]. BookWyrm надає всі базові функції для читачів – ведення списків, написання рецензій, відстеження прогресу, читацькі цілі – і при цьому є повністю безкоштовною та позбавленою реклами. Ключова перевага: повний контроль користувача над власними даними і відсутність корпоративних інтересів у формуванні стрічки [11]. Недоліки – нижча масштабованість окремих інстансів при стрімкому зростанні, менш розвинений пошук за жанрами, а також необхідність технічних знань для самостійного розгортання сервера.

Дослідження практики використання онлайн-читацьких спільнот виявляє характерну тенденцію: незважаючи на широкий вибір платформ, жодна з них повною мірою не задовольняє потребу в організованому читацькому клубі з структурованими обговореннями. Більшість платформ побудовані навколо індивідуального читача і його трекінгу, тоді як групова взаємодія є лише допоміжним функціоналом. Дані досліджень підтверджують, що 72 % віртуальних читацьких клубів втрачають половину учасників протягом перших трьох місяців [12], що свідчить про системну проблему утримання аудиторії в наявних інструментах.

Порівняльний аналіз розглянутих платформ дозволяє виявити загальні тенденції та прогалини. По-перше, більшість сервісів орієнтовані на англомовну аудиторію і не мають локалізації для менших мовних ринків, зокрема українського. По-друге, інтеграція рецензій та асинхронних групових обговорень на більшості платформ розрізнена: рецензія і дискусія існують як окремі сутності без чіткого зв’язку між собою. По-третє, функціонал книжкового

клубу (розклад читання, розподіл розділів, модерація дискусій) або відсутній, або реалізований на рівні мінімального MVP. Нарешті, алгоритмічні рекомендації навіть у найрозвиненішій реалізації (The StoryGraph) орієнтовані на пошук нових книг, а не на підбір читачів зі схожими смаками для спільного обговорення.

Таким чином, аналіз сучасного ринку платформ для обміну рецензіями демонструє наявність незайнятої ніші: спеціалізованого програмного комплексу для організації онлайн-клубу читачів, що органічно поєднує ведення персональних рецензій, структуровані обговорення у форматі клубних сесій та інструменти спільного читання в рамках однієї когезивної системи. Саме заповнення цієї ніші є метою розробки, описаної в даній роботі.

1.3 Визначення вимог до програмного комплексу онлайн-клубу читачів

Формування вимог до програмного комплексу онлайн-клубу читачів є ключовим етапом проєктування системи, оскільки саме на цьому рівні визначається, які функції повинна виконувати система, якими властивостями вона має володіти та в яких умовах повинна працювати. Вимоги формуються на основі аналізу предметної області, очікувань користувачів та сучасних підходів до розробки вебзастосунків соціального типу. Результатом цього етапу є структурована специфікація, яка слугує основою для подальшого проєктування архітектури та реалізації програмного продукту.

У межах даного проєкту вимоги до програмного комплексу визначаються як сукупність функціональних можливостей і нефункціональних характеристик, необхідних для забезпечення повноцінної роботи онлайн-клубу читачів. Функціональні вимоги описують, що саме система повинна робити, тоді як нефункціональні визначають якість її роботи, включаючи продуктивність, безпеку, зручність використання та надійність.

Такий підхід дозволяє розділити логіку системи на окремі групи вимог, що спрощує їх подальшу реалізацію, тестування та супровід. Крім того, це забезпечує відповідність між очікуваннями користувачів і технічними можливостями системи.

Перед визначенням конкретних вимог було проаналізовано основні групи користувачів системи та їхні очікування. Онлайн-клуб читачів орієнтований на кілька категорій учасників: гостей, зареєстрованих користувачів, адміністраторів та власників системи.

Гості повинні мати можливість ознайомлюватися з каталогом книг та переглядати публічні рецензії без необхідності авторизації. Зареєстровані користувачі отримують розширений функціонал, який включає можливість створення рецензій, участі в обговореннях та оцінювання книг. Адміністратори відповідають за модерацію контенту, керування користувачами та підтримку актуальності даних. Власники системи зацікавлені у стабільній роботі платформи, її безпеці та можливості масштабування.

Таким чином, система повинна підтримувати багаторівневу модель доступу та забезпечувати різні сценарії взаємодії залежно від ролі користувача.

Функціональні вимоги (табл. 1.1) визначають основні можливості системи онлайн-клубу читачів і описують її поведінку з точки зору взаємодії з користувачами.

Таблиця 1.1 – Функціональні вимоги до програмного комплексу онлайн-клубу читачів

Код	**Вимога**
FR-01	Система повинна забезпечувати реєстрацію нових користувачів
FR-02	Система повинна забезпечувати авторизацію користувачів за email і паролем
FR-03	Система повинна підтримувати розмежування ролей користувачів (гість, користувач, адміністратор)
FR-04	Система повинна надавати доступ до каталогу книг
FR-05	Система повинна забезпечувати пошук і фільтрацію книг за основними параметрами

FR-06	Система повинна дозволяти перегляд детальної інформації про книгу
FR-07	Система повинна забезпечувати створення та перегляд рецензій
FR-08	Система повинна дозволяти користувачам додавати коментарі до рецензій
FR-09	Система повинна автоматично обчислювати середній рейтинг книги на основі оцінок користувачів
FR-10	Система повинна забезпечувати функції адміністрування контенту та користувачів

У таблиці наведено функціональні вимоги до програмного комплексу онлайн-клубу читачів, які визначають основні можливості системи. Зокрема, вони охоплюють процеси реєстрації та авторизації користувачів, розмежування прав доступу відповідно до ролей, а також роботу з каталогом книг, включаючи пошук, фільтрацію та перегляд детальної інформації.

Крім того, вимоги регламентують функціонал взаємодії користувачів із контентом, а саме створення та перегляд рецензій, додавання коментарів до них і автоматичне обчислення середнього рейтингу книг. Окремо виділено адміністративні функції, які забезпечують керування контентом і користувачами системи, що дозволяє підтримувати порядок і актуальність даних у платформі.

Нефункціональні вимоги (табл. 1.2) визначають якісні характеристики системи, які впливають на її стабільність, продуктивність, безпеку та зручність використання.

Таблиця 1.2 – Нефункціональні вимоги до програмного комплексу онлайн-клубу читачів

Код	**Характеристика**	**Вимога**
NFR-01	Продуктивність	Час відгуку системи на основні операції не повинен перевищувати 1–2 секунди
NFR-02	Безпека	Паролі користувачів повинні зберігатися у хешованому вигляді
NFR-03	Безпека	Система повинна забезпечувати захист від несанкціонованого доступу до адміністративних функцій
NFR-04	Зручність використання	Інтерфейс повинен бути інтуїтивно зрозумілим та україномовним
NFR-05	Адаптивність	Система повинна коректно працювати на різних типах пристроїв

NFR-06	Надійність	Система повинна забезпечувати збереження цілісності даних
NFR-07	Супровідність	Код системи повинен бути модульним та легко розширюваним

У таблиці наведено нефункціональні вимоги до програмного комплексу онлайн-клубу читачів, які визначають якісні характеристики його роботи. Зокрема, вони регламентують продуктивність системи, забезпечуючи швидкий час відгуку на основні операції, а також вимоги до безпеки, що включають захист паролів шляхом їх хешування та обмеження несанкціонованого доступу до адміністративних функцій.

Окрім цього, вимоги охоплюють аспекти зручності використання, зокрема інтуїтивно зрозумілий україномовний інтерфейс, та адаптивність системи до різних пристроїв. Також визначено вимоги до надійності, що забезпечують цілісність даних, і супровідності, яка передбачає модульну структуру коду та можливість його подальшого розширення без суттєвих змін у системі.

Таким чином, було сформовано повний набір вимог до програмного комплексу онлайн-клубу читачів, що включає функціональні та нефункціональні аспекти. Функціональні вимоги визначають основні можливості системи, тоді як нефункціональні задають якісні параметри її роботи. Отримана специфікація є основою для подальшого проєктування архітектури системи та реалізації програмного продукту.

2 ПРОЄКТУВАННЯ ПРОГРАМНОГО КОМПЛЕКСУ ОНЛАЙН-КЛУБУ ЧИТАЧІВ

2.1 Аналіз предметної області

Предметною областю даної кваліфікаційної роботи є онлайн-клуб читачів, який представляє собою веборієнтовану соціальну платформу для взаємодії користувачів навколо літературних творів. Основна ідея подібної системи полягає не лише у наданні доступу до каталогу книг, а й у створенні середовища для активного обміну думками, рецензіями та оцінками.

На відміну від класичних бібліотечних або електронних каталогів, онлайн-клуб читачів має виражений соціальний характер. Його цінність полягає у формуванні колективної думки щодо кожного літературного твору, що реалізується через систему рецензій, коментарів та рейтингових оцінок. Таким чином, користувачі виступають не лише споживачами контенту, а й його активними генераторами.

Аналіз предметної області дозволяє виділити основні сутності, які формують інформаційну модель системи (табл. 2.1). Вони відображають як об’єкти реального світу (книги, користувачі), так і процеси взаємодії між ними (рецензії, коментарі, сесії авторизації).

Таблиця 2.1 – Основні сутності предметної області онлайн-клубу читачів

Сутність	Опис	Ключові атрибути
Користувач (Reader)	Зареєстрований учасник платформи, який взаємодіє з книгами	ім’я, email, пароль (хеш), роль
Адміністратор	Користувач із розширеними правами управління системою	усі атрибути користувача + роль admin
Книга (Book)	Літературний твір у каталозі системи	назва, автор, жанр, опис, рік, обкладинка
Рецензія (Review)	Користувацький відгук із оцінкою книги	текст, оцінка, автор, книга
Коментар (Comment)	Повідомлення в обговоренні книги або рецензії	текст, автор, дата створення
Сесія (Session)	Активний сеанс авторизованого користувача	токен, користувач, термін дії

У таблиці наведено основні сутності предметної області, які формують інформаційну модель онлайн-клубу читачів. Кожна сутність відображає окремий об’єкт системи, що бере участь у процесах зберігання, обробки та взаємодії даних. Наприклад, сутність «Користувач» відповідає за збереження інформації про зареєстрованих учасників, тоді як «Книга» є центральним елементом каталогу, навколо якого будуються рецензії та обговорення. Сутності «Рецензія» та «Коментар» забезпечують соціальний компонент системи, а «Сесія» відповідає за механізм автентифікації та безпечного доступу.

У системі онлайн-клубу читачів передбачено чітке розмежування користувачів за рівнями доступу (табл. 2.2). Це необхідно для забезпечення безпеки, керованості системи та дотримання принципу мінімально необхідних привілеїв (least privilege). У межах системи визначено три основні ролі: гість, зареєстрований користувач та адміністратор.

Гість є неавторизованим відвідувачем, який має обмежений доступ і може лише переглядати каталог книг та публічну інформацію. Зареєстрований користувач отримує розширений функціонал, що включає можливість залишати рецензії, оцінки та коментарі. Адміністратор має повний контроль над системою, включаючи управління книгами, користувачами та модерацію контенту.

Таблиця 2.2 – Порівняння можливостей користувачів системи

Можливість	**Гість**	**Користувач**	**Адміністратор**
Перегляд каталогу книг	+	+	+
Перегляд рецензій та обговорень	+	+	+
Реєстрація та авторизація	+	–	–
Створення рецензій та оцінок	–	+	+
Додавання коментарів	–	+	+
Видалення власного контенту	–	+	+
Керування книгами	–	–	+
Модерація контенту	–	–	+
Управління користувачами	–	–	+

У таблиці представлено порівняльний аналіз функціональних можливостей різних типів користувачів системи. Вона демонструє логіку

розмежування прав доступу відповідно до ролей. Найбільш обмежені можливості має гість, тоді як зареєстрований користувач отримує доступ до створення контенту. Адміністратор виконує управлінську та модераційну функцію, що забезпечує стабільність та контроль якості інформації в системі.

Діаграма варіантів використання (рис. 2.1) відображає взаємодію між акторами системи та основними функціональними можливостями онлайн-клубу читачів. Вона дозволяє візуально представити, як різні категорії користувачів взаємодіють із системою та які дії вони можуть виконувати.



Рисунок 2.1 – Діаграма варіантів використання онлайн-клубу читачів

На діаграмі наведено узагальнене порівняння рівня доступу різних типів користувачів у системі. Найменший набір функцій має гість, який обмежений переглядом інформації. Зареєстрований користувач отримує значно ширший функціонал, включаючи створення контенту. Адміністратор має максимальний рівень доступу, що охоплює всі можливості системи, включаючи керування даними та модерацію.

Отже, у межах аналізу предметної області було досліджено особливості функціонування онлайн-клубу читачів як інтерактивної вебплатформи, що поєднує функції електронного каталогу книг та соціального середовища для обміну читацьким досвідом. Визначено ключові сутності системи, зокрема користувачів, адміністраторів, книги, рецензії, коментарі та сесії, які формують інформаційну основу програмного комплексу.

Також проаналізовано ролі користувачів і розподіл їхніх прав доступу, що дозволило встановити логіку взаємодії різних категорій учасників із системою. Результати аналізу стали основою для подальшого формування функціональних вимог та проєктування архітектури програмного комплексу, забезпечуючи узгодженість між потребами користувачів і реалізацією системи.

2.2 Проєктування архітектури системи та бази даних

Архітектура програмного комплексу онлайн-клубу читачів побудована за багаторівневим принципом (layered architecture), що забезпечує чітке розділення відповідальності між компонентами системи та спрощує її подальший супровід і масштабування. У якості технологічної основи використано фреймворк Next.js (App Router), який дозволяє поєднувати серверну та клієнтську логіку в межах єдиної архітектури. Особливістю реалізації є використання React Server Components, що дає змогу виконувати частину логіки безпосередньо на сервері, зменшуючи навантаження на клієнтську сторону та мінімізуючи кількість

мережевих запитів. Для виконання операцій зміни даних застосовано Server Actions, що дозволяє реалізувати взаємодію з базою даних без необхідності побудови окремого REST API.

У структурі системи виділено чотири основні рівні, кожен з яких виконує чітко визначені функції (табл. 2.3).

Таблиця 2.3 – Рівні архітектури програмного комплексу онлайн-клубу читачів

Рівень системи	**Призначення**	**Реалізація в проєкті**
Рівень представлення	Забезпечує взаємодію користувача з системою через інтерфейс	`app/`, `components/`
Рівень бізнес-логіки	Обробка даних, перевірка правил, авторизація	`lib/actions.ts`, `lib/auth.ts`
Рівень доступу до даних	Виконання запитів до бази даних та обробка результатів	`lib/data.ts`
Рівень зберігання даних	Фізичне зберігання інформації та ініціалізація БД	`lib/db.ts`, `club.db` (SQLite)

Рівень представлення відповідає за взаємодію користувача із системою. На цьому рівні передбачено сторінки каталогу книг, сторінки окремих книг, форми авторизації та реєстрації, а також інтерфейси для перегляду та створення рецензій. Основною метою цього рівня є забезпечення зручного та інтуїтивного доступу до функціоналу системи без необхідності розуміння внутрішніх процесів.

Рівень бізнес-логіки є центральним у структурі системи, оскільки саме тут реалізуються правила роботи онлайн-клубу читачів. До таких правил належать перевірка коректності введених даних, обробка процесів реєстрації та авторизації, контроль доступу до функцій системи, а також логіка формування та оновлення рейтингів книг на основі користувацьких оцінок.

Рівень доступу до даних забезпечує взаємодію між бізнес-логікою та сховищем інформації. Його функція полягає у виконанні операцій читання, додавання, оновлення та видалення даних відповідно до запитів системи. На цьому рівні важливим є дотримання цілісності даних та мінімізація надлишкових операцій.

Рівень зберігання даних відповідає за фізичне розміщення інформації у структурованому вигляді. Він забезпечує довготривале збереження даних про користувачів, книги, рецензії, коментарі та сесії доступу.

Архітектурна модель взаємодії компонентів системи відображає основні потоки даних між користувачем, серверною частиною та базою даних (рис. 2.2).

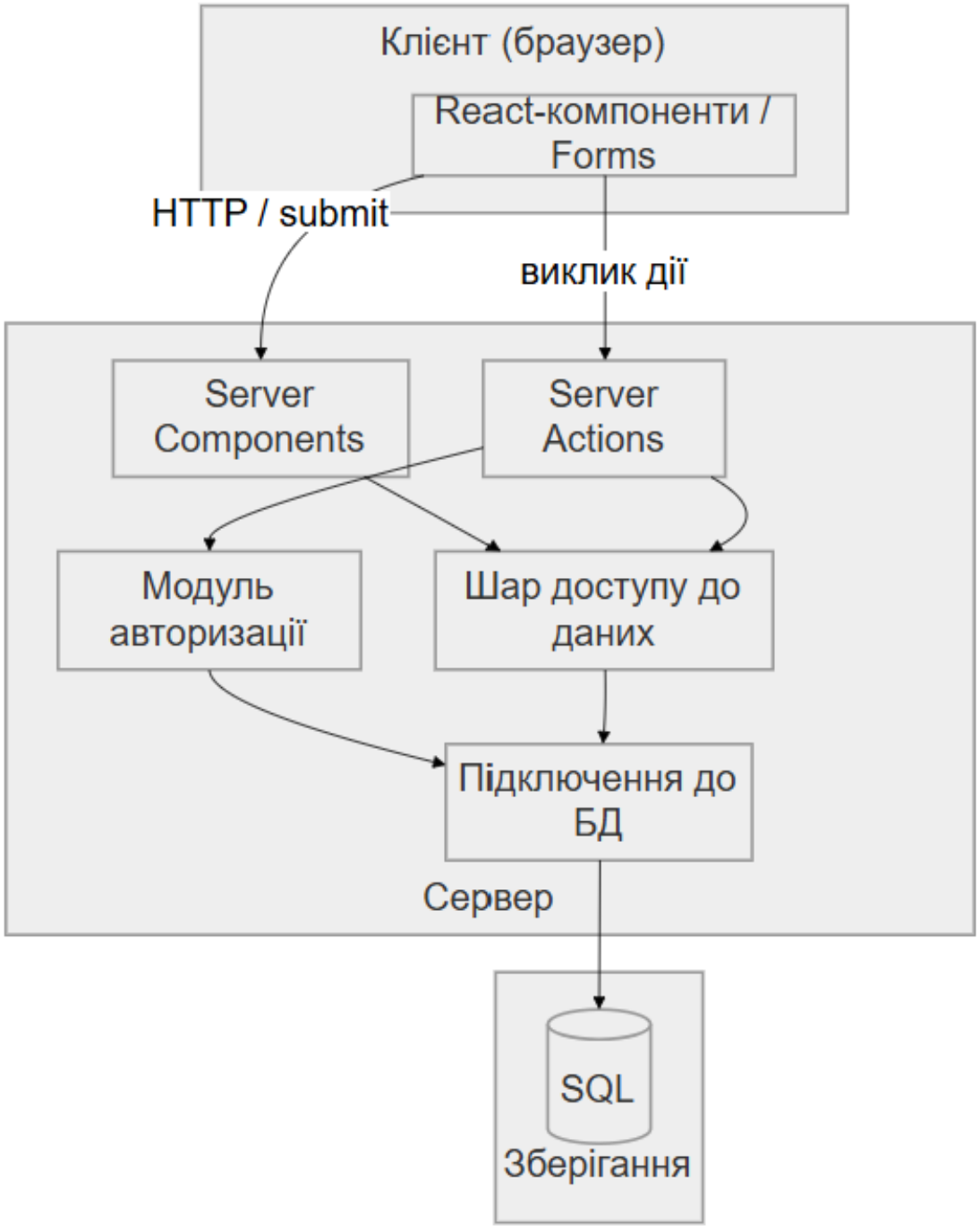


Рисунок 2.2 – Архітектура взаємодії компонентів програмного комплексу онлайн-клубу читачів

Діаграма ілюструє загальний принцип роботи системи: користувач взаємодіє з інтерфейсом у браузері, після чого запити надходять до рівня представлення. Далі управління передається на рівень бізнес-логіки, де відбувається обробка правил системи, перевірка автентифікації та виконання серверних дій. Рівень доступу до даних формує SQL-запити та взаємодіє з базою даних, після чого результат повертається назад до користувача через усі рівні системи. Така архітектура забезпечує контрольований потік даних і чітке розділення відповідальності між компонентами.

Перед побудовою фізичної структури бази даних доцільним є створення її концептуальної моделі (рис. 2.3), яка відображає основні сутності предметної області та взаємозв’язки між ними на логічному рівні. Такий підхід дозволяє на ранньому етапі проєктування узгодити структуру даних із функціональними вимогами системи та забезпечити коректну організацію інформаційних потоків.

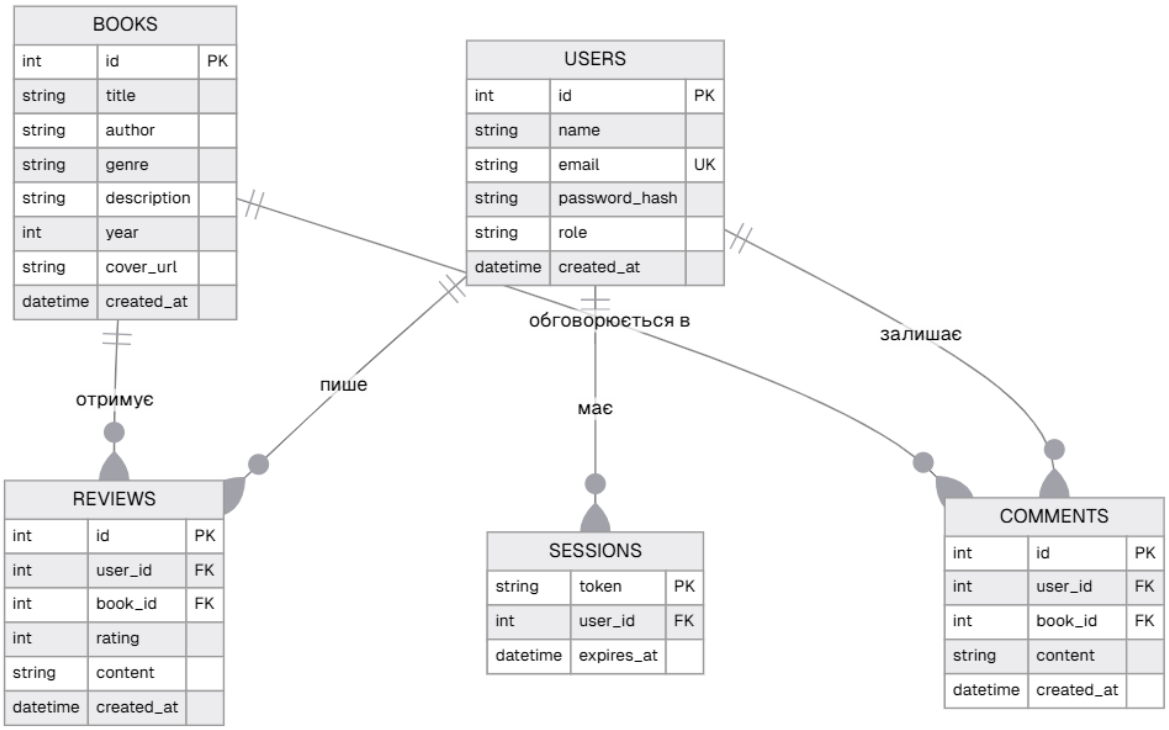


Рисунок 2.3 – Концептуальна ER-модель бази даних онлайн-клубу читачів

Дана ER-діаграма відображає концептуальну модель даних системи онлайн-клубу читачів. У центрі моделі знаходиться сутність BOOKS, яка є ключовим елементом предметної області, оскільки саме навколо книг будується основна взаємодія користувачів. Вона пов’язана з іншими сутностями через механізми рецензій та коментарів, що формують соціальний та інтерактивний рівень платформи.

Користувачі системи виступають активними учасниками взаємодії, оскільки можуть створювати рецензії та коментарі, оцінювати книги та брати участь в обговореннях. Це реалізує класичний зв’язок типу «один-до-багатьох», де один користувач може бути автором багатьох записів, але кожен запис належить лише одному користувачу.

Окрему роль у моделі відіграє сутність SESSIONS, яка відповідає за управління авторизацією користувачів та зберігання інформації про активні сеанси доступу до системи. Вона забезпечує контроль входу в систему та підтримує безпечну взаємодію користувача з платформою.

Таким чином, запропонована концептуальна модель забезпечує логічну цілісність даних, відображає основні процеси предметної області та створює основу для подальшого проєктування фізичної структури бази даних.

Реляційна модель бази даних онлайн-клубу читачів побудована у вигляді взаємопов’язаних таблиць, що відображають основні сутності предметної області та забезпечують логічну цілісність інформації. Кожна таблиця містить первинні та зовнішні ключі, що дозволяє реалізувати зв’язки між сутностями та уникнути дублювання даних. Така структура є основою для стабільної роботи системи, оскільки забезпечує коректне зберігання, обробку та вибірку інформації.

Таблиця users (табл. 2.4) є базовою сутністю системи, оскільки саме вона забезпечує збереження інформації про всіх зареєстрованих учасників онлайн-клубу читачів. Через цю таблицю реалізуються процеси автентифікації та авторизації користувачів, а також визначаються їхні права доступу до функціоналу системи. Поле email використовується як унікальний ідентифікатор

для входу, а password_hash забезпечує безпечне зберігання пароля в захищеному вигляді. Поле role дозволяє розмежовувати звичайних користувачів і адміністратора, що є важливим для реалізації системи доступу

Таблиця 2.4 – Структура таблиці users

Поле	Тип даних	Обмеження	Опис
id	INTEGER	PK, AUTOINCREMENT	Унікальний ідентифікатор користувача
name	TEXT	NOT NULL	Ім’я користувача
email	TEXT	UNIQUE, NOT NULL	Електронна пошта (використовується як логін)
password_hash	TEXT	NOT NULL	Захешований пароль користувача
role	TEXT	DEFAULT ‘user’	Роль користувача в системі (звичайний користувач або адміністратор)
created_at	DATETIME	DEFAULT CURRENT_TIMESTAMP	Дата та час створення облікового запису

Таблиця books (табл. 2.5) призначена для зберігання інформації про літературні твори, доступні в каталозі онлайн-клубу читачів. Вона є центральною інформаційною сутністю предметної області, оскільки саме навколо книг формується вся взаємодія користувачів. Поля таблиці містять як основні бібліографічні дані (назва, автор, рік), так і додаткові атрибути, що забезпечують зручність використання системи, зокрема опис та обкладинку.

Таблиця 2.5 – Структура таблиці books

Поле	Тип даних	Обмеження	Опис
id	INTEGER	PK	Унікальний ідентифікатор книги
title	TEXT	NOT NULL	Назва літературного твору
author	TEXT	NOT NULL	Автор книги
genre	TEXT	–	Жанрова належність книги
description	TEXT	–	Короткий опис або анотація книги
year	INTEGER	–	Рік видання
cover_url	TEXT	–	Посилання на обкладинку книги
created_at	DATETIME	DEFAULT CURRENT_TIMESTAMP	Дата додавання книги до каталогу

Таблиця reviews (табл. 2.6) використовується для збереження користувацьких рецензій, які формують основу соціальної взаємодії в системі. Вона пов’язує користувачів із книгами через механізм оцінювання та текстових відгуків. Поле rating дозволяє формувати агрегований середній рейтинг книги, що використовується для її оцінки в каталозі, а поля user_id і book_id забезпечують зв’язок із відповідними сутностями.

Таблиця 2.6 – Структура таблиці reviews

Поле	Тип даних	Обмеження	Опис
id	INTEGER	PK	Унікальний ідентифікатор рецензії
user_id	INTEGER	FK – users(id)	Ідентифікатор користувача-автора
book_id	INTEGER	FK – books(id)	Ідентифікатор книги
rating	INTEGER	1–5	Оцінка книги користувачем
content	TEXT	NOT NULL	Текст рецензії
created_at	DATETIME	DEFAULT CURRENT_TIMESTAMP	Дата створення рецензії

Процес автентифікації є одним із ключових механізмів функціонування онлайн-клубу читачів, оскільки саме він забезпечує контроль доступу до персоналізованих функцій системи. Для його формалізації використано діаграму послідовності (рис. 2.4), яка дозволяє наочно відобразити взаємодію між користувачем, серверною частиною та базою даних у часовій послідовності.

У межах даного процесу користувач спочатку вводить свої облікові дані у форму входу (email та пароль), після чого система передає ці дані на сервер для перевірки. Сервер виконує пошук відповідного запису в базі даних за email-адресою та, у разі його наявності, перевіряє коректність пароля шляхом порівняння з захешованим значенням. У випадку успішної перевірки створюється сесія користувача, яка зберігається у системі та використовується для подальшої ідентифікації. Якщо ж перевірка не проходить, користувач отримує повідомлення про помилку авторизації.

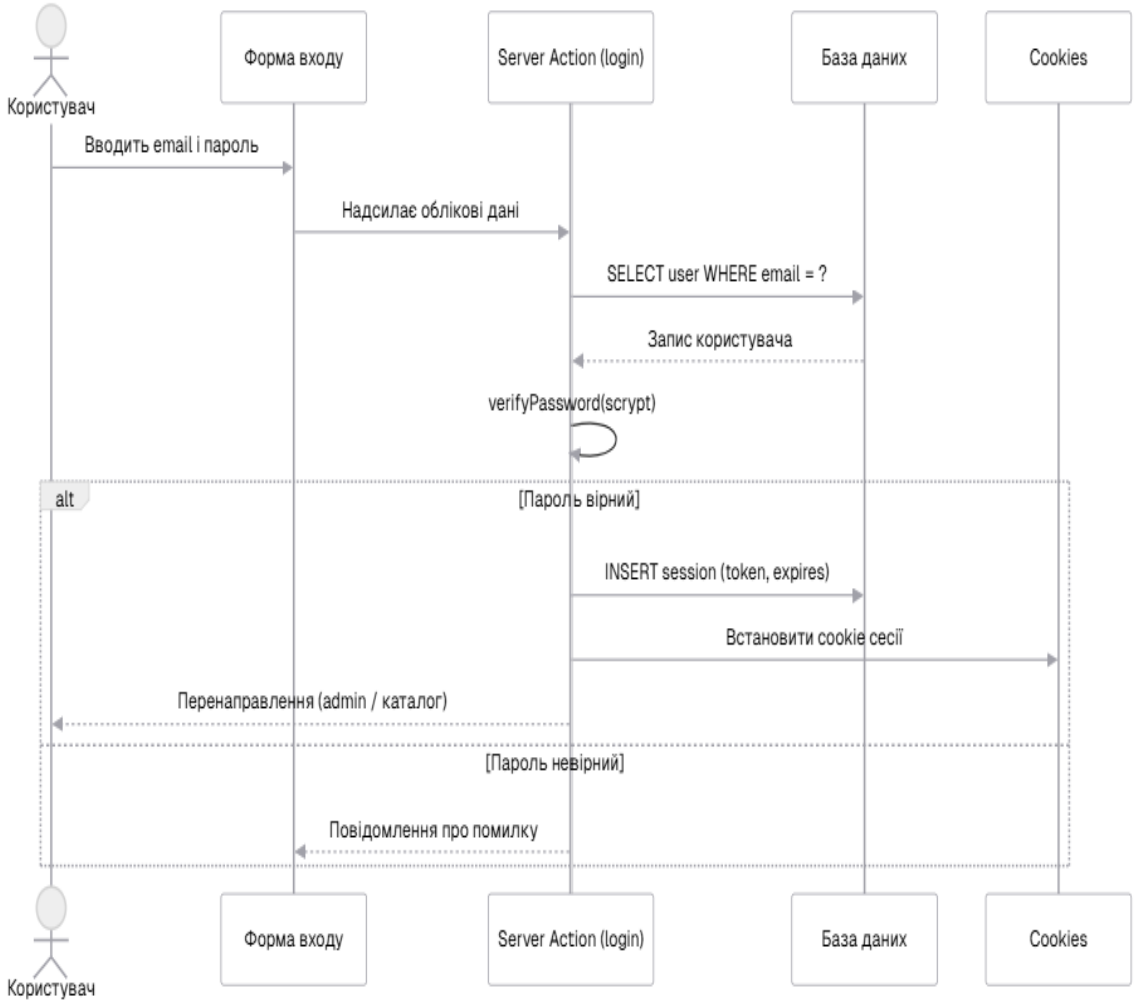


Рисунок 2.4 – Послідовність процесу входу користувача в систему

Описана діаграма відображає повний цикл автентифікації користувача, починаючи з введення облікових даних і завершуючи прийняттям рішення про доступ. Така структура дозволяє чітко відокремити етапи перевірки даних і забезпечує зрозумілу логіку обробки як успішних, так і неуспішних спроб входу.

З метою забезпечення надійності та захищеності програмного комплексу онлайн-клубу читачів на етапі проєктування передбачено комплекс заходів інформаційної безпеки (табл. 2.7). Вони охоплюють захист облікових даних, контроль доступу до функціоналу системи, а також запобігання поширеним типам атак на вебзастосунки.

Окрему увагу приділено захисту персональних даних користувачів, оскільки система передбачає зберігання облікової інформації та історії

активності. Крім того, важливим аспектом є забезпечення цілісності даних і недопущення несанкціонованого доступу до адміністративних функцій.

Таблиця 2.7 – Механізми захисту системи

Загроза	**Метод захисту**
Перехоплення паролів	Хешування паролів із використанням алгоритму scrypt та унікальної солі
SQL-ін’єкції	Використання параметризованих запитів до бази даних
Несанкціонований доступ	Перевірка ролей користувачів на рівні серверної логіки
Підробка запитів	Захист серверних операцій через контрольовані механізми обробки запитів
Викрадення сесії	Використання захищених cookie з обмеженим терміном дії (HttpOnly, Secure)

У таблиці наведено основні потенційні загрози безпеці системи та відповідні механізми їх запобігання. Запропоновані рішення формують базовий, але достатній рівень захисту, який дозволяє мінімізувати ризики несанкціонованого доступу, втрати даних або їх модифікації.

У процесі проєктування архітектури програмного комплексу онлайн-клубу читачів було сформовано цілісну структурну модель системи, що базується на принципах багаторівневої організації. Визначено чіткий поділ на рівні представлення, бізнес-логіки, доступу до даних та їх зберігання, що забезпечує логічну ізоляцію компонентів і спрощує подальший розвиток системи.

Окремо розроблено концептуальну модель даних, яка відображає основні сутності предметної області та зв’язки між ними, а також логічну структуру реляційної бази даних. Це дозволяє забезпечити узгоджене зберігання інформації та ефективну взаємодію між елементами системи.

Крім того, спроєктовано процес автентифікації користувачів, який визначає механізм безпечного доступу до функціоналу системи, а також сформовано перелік базових заходів інформаційної безпеки. У сукупності отримані результати створюють надійну основу для подальшої реалізації програмного комплексу.

2.3 Проєктування користувацького інтерфейсу та вибір технологій розробки

Проєктування програмного комплексу онлайн-клубу читачів передбачає визначення технологічного стеку, який забезпечує ефективну реалізацію функціональних вимог, підтримує масштабованість системи та дозволяє створити зручний і сучасний користувацький інтерфейс. Вибір технологій здійснювався з урахуванням критеріїв продуктивності, простоти підтримки, популярності інструментів та їх відповідності сучасним практикам веб-розробки.

Основний стек технологій подано в таблиці 2.8.

Таблиця 2.8 – Вибір технологічного стеку

Категорія	**Технологія**	**Обґрунтування вибору**
Мова програмування	TypeScript [14]	Забезпечує статичну типізацію, зменшує кількість помилок на етапі розробки та підвищує надійність коду
Фреймворк	Next.js (App Router) [15]	Підтримує серверний рендеринг, маршрутизацію на основі файлової структури та сучасні підходи до побудови веб-застосунків
Бібліотека UI	React [16]	Використовує компонентну архітектуру, що спрощує повторне використання елементів інтерфейсу
Стилізація	Tailwind CSS [17]	Дозволяє швидко створювати адаптивний інтерфейс через утилітарні класи
UI-компоненти	shadcn/ui [18]	Надає готові, але гнучко налаштовувані компоненти інтерфейсу
СУБД	SQLite [19]	Легка реляційна база даних, яка не потребує окремого серверного розгортання
Іконки	Lucide React [20]	Уніфікований набір сучасних іконок для інтерфейсу

Наведена таблиця відображає вибір основних технологій, що використовуються для реалізації програмного комплексу. Поєднання зазначених інструментів дозволяє забезпечити баланс між продуктивністю розробки, зручністю підтримки коду та якістю користувацького досвіду.

Візуальне оформлення інтерфейсу є важливим елементом проєктування, оскільки безпосередньо впливає на зручність використання системи та

сприйняття контенту користувачем. У межах даного проєкту було обрано стриману, «книжкову» кольорову палітру, яка асоціюється з читанням, комфортом і концентрацією уваги (табл. 2.9).

Таблиця 2.9 – Колірна схема інтерфейсу

Назва кольору	**HEX**	**Призначення**
Air superiority blue	#78A1BB	Основний акцентний колір (кнопки, посилання, активні елементи)
Mint cream	#EBF5EE	Основний фон сторінок
Khaki	#BFA89E	Додаткові акценти та декоративні елементи
Нейтральні відтінки	#FFFFFF, сірі тони	Фони карток, текст, розділювачі

Застосована кольорова схема забезпечує візуальну гармонійність інтерфейсу та не перевантажує користувача зайвими контрастами. Це особливо важливо для систем, орієнтованих на тривале читання текстового контенту.

Типографічна система базується на поєднанні двох шрифтів: Playfair Display для заголовків, що створює асоціацію з друкованими літературними виданнями, та Geist для основного тексту, який забезпечує високу читабельність. Оптимізований міжрядковий інтервал (1.4–1.6) сприяє комфортному сприйняттю великих обсягів тексту.

Діаграма відображає логічну структуру веб-застосунку (рис. 2.5) та взаємозв’язки між основними сторінками системи. Вона демонструє повний маршрут користувача в межах платформи – від головної сторінки до перегляду каталогу, окремої книги, а також взаємодії з функціоналом автентифікації та адміністративною частиною системи.

Особливістю побудови інтерфейсу є чітке розділення на публічну частину (доступну всім користувачам) та адміністративний модуль (доступний лише користувачам із відповідними правами доступу). Це забезпечує безпеку, логічну організацію функцій та зручність навігації.

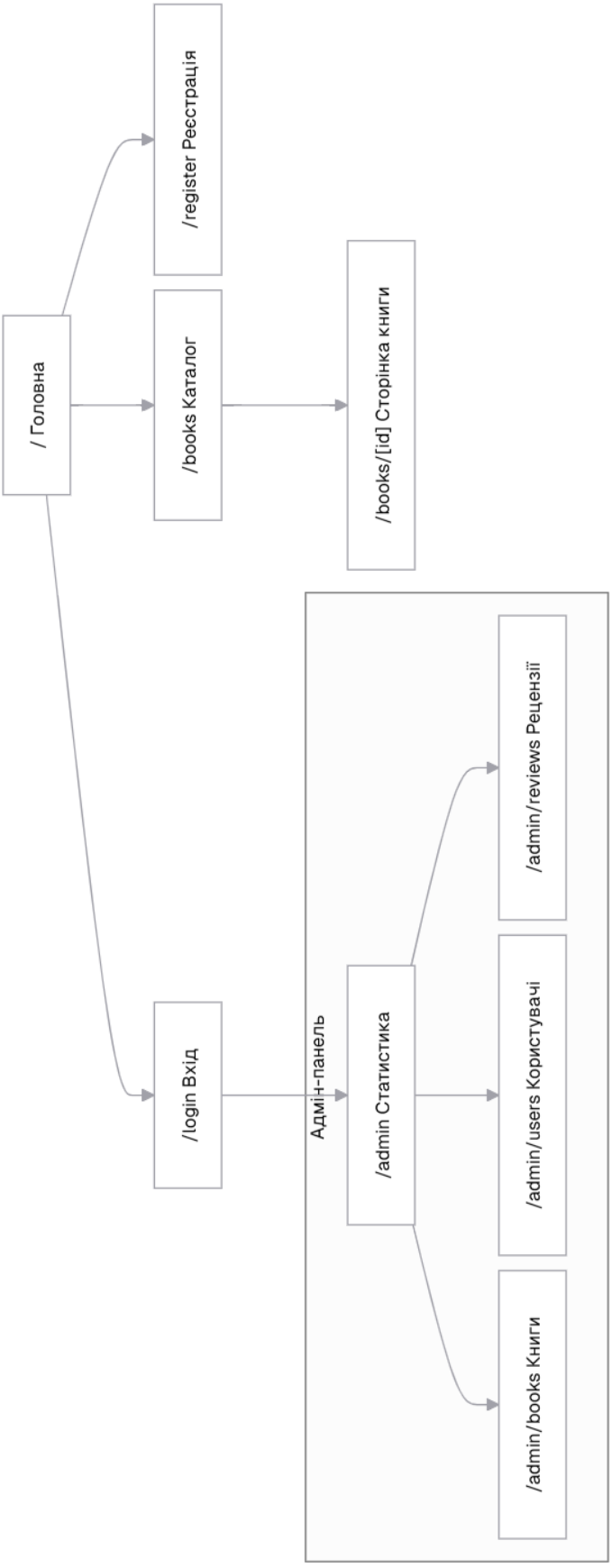


Рисунок 2.5 – Структура навігації користувацького інтерфейсу онлайн-клубу читачів

Публічна частина включає головну сторінку, каталог книг, сторінку окремої книги, а також сторінки входу та реєстрації.

Адміністративна частина містить модулі керування книгами, користувачами, рецензіями та перегляд статистики системи.

Таким чином, структура навігації побудована за принципом ієрархічної організації, де центральною точкою є каталог книг, а всі інші сторінки логічно пов’язані з ним.

На основі визначених технологічних рішень доцільним є розгляд внутрішньої організації користувацького інтерфейсу як набору взаємопов’язаних компонентів. Саме тому наступним етапом проєктування є аналіз його компонентної структури, яка визначає принципи побудови та повторного використання елементів інтерфейсу (рис. 2.6).

Дана діаграма відображає компонентну архітектуру користувацького інтерфейсу системи онлайн-клубу читачів та демонструє принцип модульного поділу UI на повторно використовувані елементи. Основною метою такого підходу є забезпечення узгодженості інтерфейсу, спрощення його супроводу та підвищення масштабованості системи.

У верхньому рівні структури знаходиться Root Layout, який виступає базовим шаблоном усієї системи та визначає загальну структуру сторінок. Від нього логічно відгалужуються дві основні частини: публічна зона та адміністративна панель.

Публічна частина охоплює сторінки, доступні всім користувачам або авторизованим учасникам. Вона побудована на основі повторного використання ключових UI-компонентів, таких як картки книг, форми рецензій, форми коментарів та елементи рейтингу. Це дозволяє забезпечити єдиний візуальний стиль і однакову логіку взаємодії на всіх сторінках, пов’язаних із переглядом та обговоренням книг.

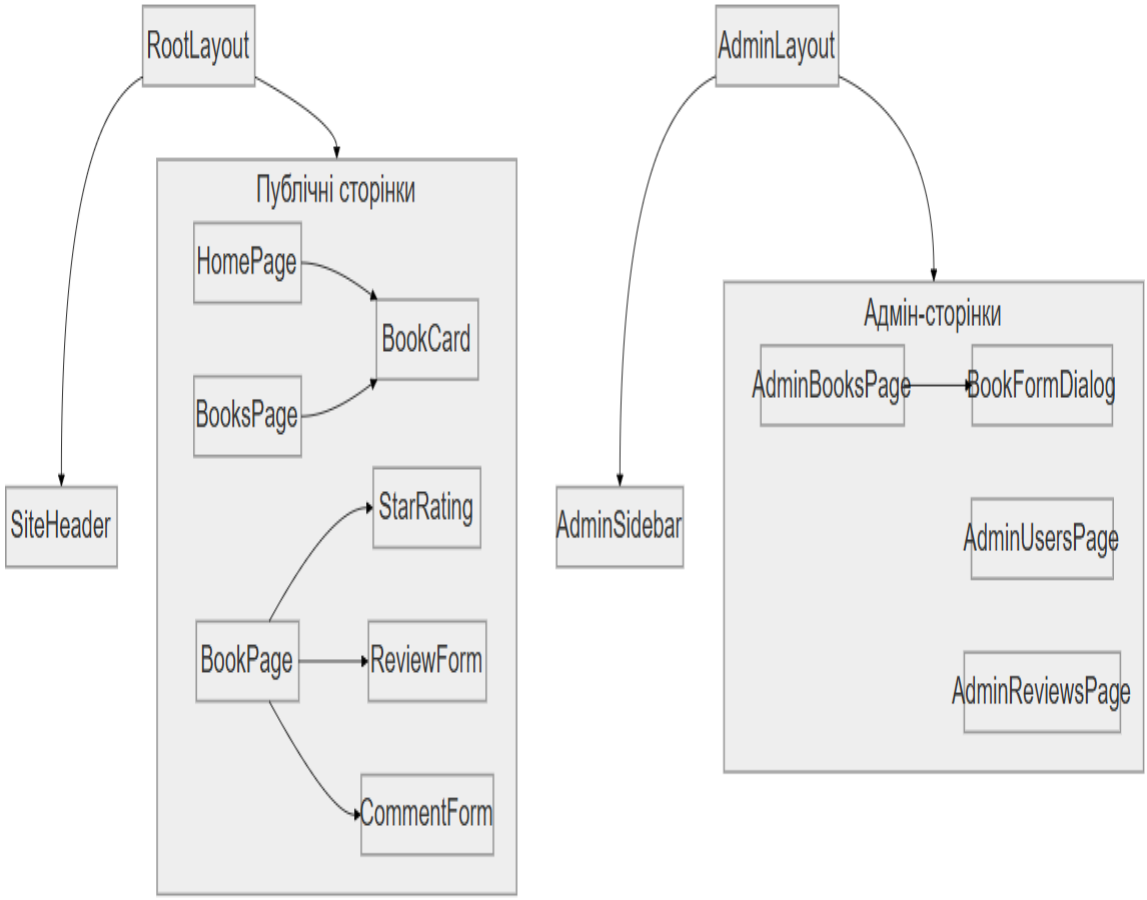


Рисунок 2.6 – Компонентна структура користувацького інтерфейсу

Адміністративна частина інтерфейсу виділена в окрему підсистему, яка має власний макет (Admin Layout) та навігаційну структуру. Вона включає спеціалізовані компоненти для керування даними системи: таблиці користувачів, модулі редагування книг, панелі модерації рецензій та статистичні блоки. Такий поділ дозволяє чітко відокремити користувацький та адміністративний функціонал і знизити ризик змішування логіки різних рівнів доступу.

Завершуючи аналіз компонентної структури інтерфейсу, доцільно перейти до розгляду конкретних елементів користувацької взаємодії, а саме ключових екранів системи, які реалізують описані компоненти на практичному рівні.

Інтерфейс системи спроєктовано з орієнтацією на просту навігацію, логічну структуру сторінок та швидкий доступ до основних функцій онлайн-клубу читачів. Кожен екран виконує чітко визначену роль у загальній взаємодії користувача з платформою (табл. 2.10).

Таблиця 2.10 – Ключові екрани інтерфейсу

Екран	Призначення	Основні елементи
Головна сторінка	Початкове ознайомлення з платформою та популярними книгами	Промо-блок, добірка книг, навігаційні елементи
Каталог книг	Перегляд і пошук літературних творів	Пошук, фільтри, сітка карток книг
Сторінка книги	Детальна інформація про книгу та взаємодія з нею	Опис, рейтинг, рецензії, коментарі
Вхід / реєстрація	Автентифікація користувачів	Форми входу та реєстрації з валідацією
Адміністративна панель	Загальне керування системою	Статистичні блоки, навігація по розділах
Керування книгами	Управління каталогом (CRUD-операції)	Таблиці даних, форми додавання та редагування
Користувачі	Керування обліковими записами	Список користувачів, зміна ролей
Модерація рецензій	Контроль користувацького контенту	Перелік рецензій, інструменти видалення

Після визначення структури компонентів інтерфейсу та основних екранів системи логічним є розгляд принципів, які були покладені в основу проєктування користувацького інтерфейсу.

Проєктування користувацького інтерфейсу здійснювалося з урахуванням сучасних принципів UX/UI-дизайну, що спрямовані на забезпечення зручності використання, доступності та логічної узгодженості системи.

Першим базовим підходом є Mobile-first дизайн, який передбачає розробку інтерфейсу спочатку для мобільних пристроїв із подальшою адаптацією до більших екранів. Це дозволяє забезпечити коректне відображення та зручність використання на всіх типах пристроїв.

Другим важливим принципом є семантична структура інтерфейсу, що реалізується через використання логічно структурованих елементів розмітки. Це покращує не лише доступність системи, а й її семантичну зрозумілість для пошукових систем та допоміжних технологій.

Також у межах проєктування застосовано принцип єдиної системи стилів, який передбачає використання узгоджених дизайн-токенів для кольорів,

типографіки та відступів. Це забезпечує візуальну цілісність інтерфейсу та зменшує ймовірність стилістичних розбіжностей між компонентами.

Окрему увагу приділено принципу зворотного зв’язку з користувачем, що реалізується через повідомлення про результати дій (успішні або помилкові операції), індикатори завантаження та інтерактивні елементи взаємодії.

Завершальним етапом розгляду користувацького інтерфейсу є узагальнення принципів його побудови, що дозволяє оцінити цілісність і ефективність прийнятих проєктних рішень.

Отже, здійснено проєктування користувацького інтерфейсу та обґрунтовано вибір технологій розробки програмного комплексу онлайн-клубу читачів. Визначено оптимальний технологічний стек, який забезпечує ефективну реалізацію інтерфейсу, його масштабованість та зручність підтримки.

Також спроєктовано колірну схему та типографіку, що формують єдиний візуальний стиль системи та сприяють комфортному сприйняттю контенту. Розроблено структуру навігації та компонентну архітектуру інтерфейсу, що забезпечує модульність і повторне використання UI-елементів.

Крім того, визначено ключові екрани системи та сформульовано основні принципи UX/UI-проєктування, які забезпечують зручність, логічність та узгодженість користувацької взаємодії. Отримані результати створюють основу для подальшої реалізації інтерфейсу програмного комплексу.

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОГО КОМПЛЕКСУ ОНЛАЙН-КЛУБУ ЧИТАЧІВ

3.1 Реалізація основного функціоналу системи

Розроблений програмний засіб реалізовано із використанням фреймворку Next.js, який забезпечує підтримку серверного рендерингу, маршрутизації та сучасних механізмів взаємодії між клієнтською і серверною частинами застосунку. Для підвищення зручності супроводу та подальшого масштабування проєкту його структура організована за модульним принципом. Кожна директорія відповідає за окремий аспект функціонування системи, що дозволяє зменшити зв’язність між компонентами та спростити процес внесення змін.

Основні сторінки застосунку розміщені у каталозі app/, який реалізує концепцію App Router. Адміністративна частина системи винесена в окрему директорію app/admin/, що дозволило ізолювати функції керування користувачами, книгами та модерації контенту від загальнодоступної частини ресурсу. Багаторазові елементи інтерфейсу згруповано у директорії components/, тоді як базові компоненти бібліотеки shadcn/ui зосереджені у каталозі components/ui/.

Окрему групу становлять службові модулі директорії lib/. Зокрема, файл lib/db.ts містить логіку підключення до бази даних, створення таблиць та реалізації механізмів захисту паролів. У модулі lib/data.ts реалізовано функції доступу до даних, які виконують SQL-запити та повертають підготовлені об’єкти. Модуль lib/auth.ts забезпечує керування процесами автентифікації та підтримки користувацьких сесій, а файл lib/actions.ts містить серверні дії для обробки запитів на зміну даних. Використання окремого файлу lib/types.ts дало змогу централізовано описати типи даних, що застосовуються у різних частинах системи.

Така організація програмного коду сприяє покращенню його читабельності, забезпечує повторне використання окремих компонентів та відповідає сучасним підходам до проєктування вебзастосунків.

Функціонування системи базується на використанні вбудованої системи керування базами даних SQLite, доступ до якої реалізовано через модуль node:sqlite. Логіку взаємодії з базою даних зосереджено у файлі lib/db.ts, фрагменти якого наведено у додатку А.1.

Під час запуску застосунку виконується перевірка наявності каталогу для зберігання файлу бази даних. У разі його відсутності директорія створюється автоматично. Після цього відбувається встановлення з’єднання з базою даних та активація спеціальних режимів роботи SQLite, зокрема режиму журналювання WAL та механізму підтримки зовнішніх ключів (foreign_keys).

Для уникнення створення декількох одночасних з’єднань під час гарячого перезавантаження застосунку використано глобальний об’єкт globalThis, у якому зберігається екземпляр відкритої бази даних. Такий підхід є особливо важливим під час розробки, оскільки дозволяє запобігти виникненню помилок блокування файлу бази даних.

Створення структури бази даних здійснюється функцією initDb(), яка виконується лише один раз завдяки використанню логічного прапорця initialized. У межах цієї функції створюються таблиці users, sessions, books, reviews та comments. Їхня структура забезпечує зберігання інформації про користувачів системи, активні сеанси роботи, каталог літературних творів, рецензії користувачів та коментарі до книг відповідно.

Після успішного створення таблиць виконується процедура початкового наповнення бази даних тестовими записами. Функція seed() перевіряє наявність користувачів та книг у системі і, за потреби, створює обліковий запис адміністратора, стандартного користувача, а також додає приклади літературних творів. Такий підхід забезпечує можливість тестування функціоналу застосунку одразу після його запуску без необхідності ручного введення початкових даних.

Однією з важливих вимог до розробленої системи є забезпечення захисту облікових даних користувачів. Для цього реалізовано механізм хешування паролів із використанням криптографічного алгоритму scrypt, який входить до стандартного набору засобів Node.js.

Під час реєстрації нового користувача функція hashPassword() генерує випадкову послідовність байтів, яка використовується як сіль (salt). Далі на основі введеного пароля та сформованої солі обчислюється хеш довжиною 64 байти. У базі даних зберігається лише рядок, що містить сіль та результат хешування, розділені символом двокрапки. Відкритий текст пароля при цьому не зберігається.

Під час авторизації користувача виконується функція verifyPassword(), яка розділяє збережений рядок на складові, повторно обчислює хеш для введеного пароля та здійснює порівняння отриманих значень. Для захисту від атак, заснованих на аналізі часу виконання операцій, застосовано функцію timingSafeEqual(), яка забезпечує безпечне порівняння двох масивів байтів.

Використання алгоритму scrypt дозволяє підвищити стійкість системи до атак методом повного перебору та ускладнює отримання паролів навіть у випадку компрометації бази даних.

Основний функціонал системи пов’язаний із роботою з каталогом книг, рецензіями та коментарями користувачів. Відповідна логіка реалізована у модулі lib/data.ts, фрагменти якого подано у додатку А.2. Також код роботи з каталогом наведено в додатку А.3.

Для отримання переліку літературних творів використовується функція listBooks(). Вона підтримує можливість пошуку за назвою книги або ім’ям автора, а також фільтрацію за жанром. SQL-запит формується динамічно залежно від переданих параметрів, що забезпечує гнучкість механізму пошуку. Додатково під час вибірки виконується обчислення середнього рейтингу книги та кількості залишених рецензій за допомогою агрегатних функцій AVG() та COUNT().

Для перегляду детальної інформації про окремий твір реалізовано функцію getBook(), яка повертає основні характеристики книги разом зі статистичними показниками. Відгуки користувачів отримуються функцією getReviews(), що виконує об’єднання таблиць reviews та users для відображення імен авторів рецензій.

Механізм додавання та редагування рецензій реалізовано за допомогою функції upsertReview(). Особливістю цього підходу є використання конструкції ON CONFLICT, яка дозволяє оновлювати існуючу рецензію користувача у випадку повторного надсилання відгуку на ту саму книгу. Видалення рецензій здійснюється функцією deleteReview().

Робота з коментарями реалізована через функції getComments(), addComment() та deleteComment(). Вони забезпечують перегляд дискусій щодо літературних творів, додавання нових повідомлень та видалення недоречних коментарів.

Окремий блок функцій призначений для адміністративної панелі системи. Зокрема, реалізовано можливості створення, редагування та видалення записів про книги (createBook(), updateBook(), deleteBook()), перегляду статистичних показників роботи системи (adminStats()), отримання списку зареєстрованих користувачів (listUsers()), зміни ролей (setUserRole()) та видалення користувачів (deleteUser()).

Також передбачено функцію listAllReviews(), що дозволяє адміністраторам здійснювати модерацію опублікованих рецензій.

Таким чином, реалізований набір функцій забезпечує повноцінну роботу вебзастосунку клубу читачів, підтримуючи всі основні сценарії взаємодії користувачів із системою: перегляд каталогу, пошук книг, оцінювання літературних творів, участь в обговореннях та адміністрування інформаційного наповнення ресурсу.

3.2 Розробка модулів рецензій, обговорень та користувацьких профілів

У межах розробленої інформаційної системи «книжкового клубу» особливу роль відіграють модулі рецензій, коментарів та користувацьких профілів, оскільки саме вони формують соціальну взаємодію між користувачами та забезпечують контентну наповненість платформи. Реалізація цих модулів базується на архітектурному підході, що передбачає чітке розділення клієнтської логіки, серверних операцій та рівня доступу до даних, що дозволяє забезпечити масштабованість, безпеку та передбачуваність роботи системи.

Модуль рецензій є центральним елементом користувацької взаємодії, оскільки дозволяє формувати якісну оцінку літературних творів на основі колективного досвіду користувачів. Його функціональність передбачає можливість залишення одного структурованого відгуку на кожну книгу, який включає текстову частину та числову оцінку в діапазоні від 1 до 5.

З технічної точки зору реалізація модуля базується на взаємодії клієнтського компонента ReviewForm, серверної дії submitReview та функцій доступу до даних у шарі lib/data.ts. При цьому клієнтський компонент відповідає виключно за взаємодію з користувачем, включаючи вибір рейтингу та введення тексту, тоді як усі операції з перевірки, збереження та оновлення даних виконуються на серверному рівні.

Процес створення рецензії починається з того, що користувач вводить відповідні дані у формі, після чого інформація передається до серверної дії. На цьому етапі виконується перевірка автентифікації користувача, що унеможливлює виконання операцій неавторизованими особами. У разі успішної перевірки дані зберігаються або оновлюються у таблиці reviews бази даних SQLite. Після цього система ініціює оновлення інтерфейсу через механізм повторної валідації сторінки, що забезпечує відображення актуального стану даних без необхідності ручного перезавантаження сторінки.

Схематично процес взаємодії учасників системи під час створення рецензії представлено на діаграмі 3.1.

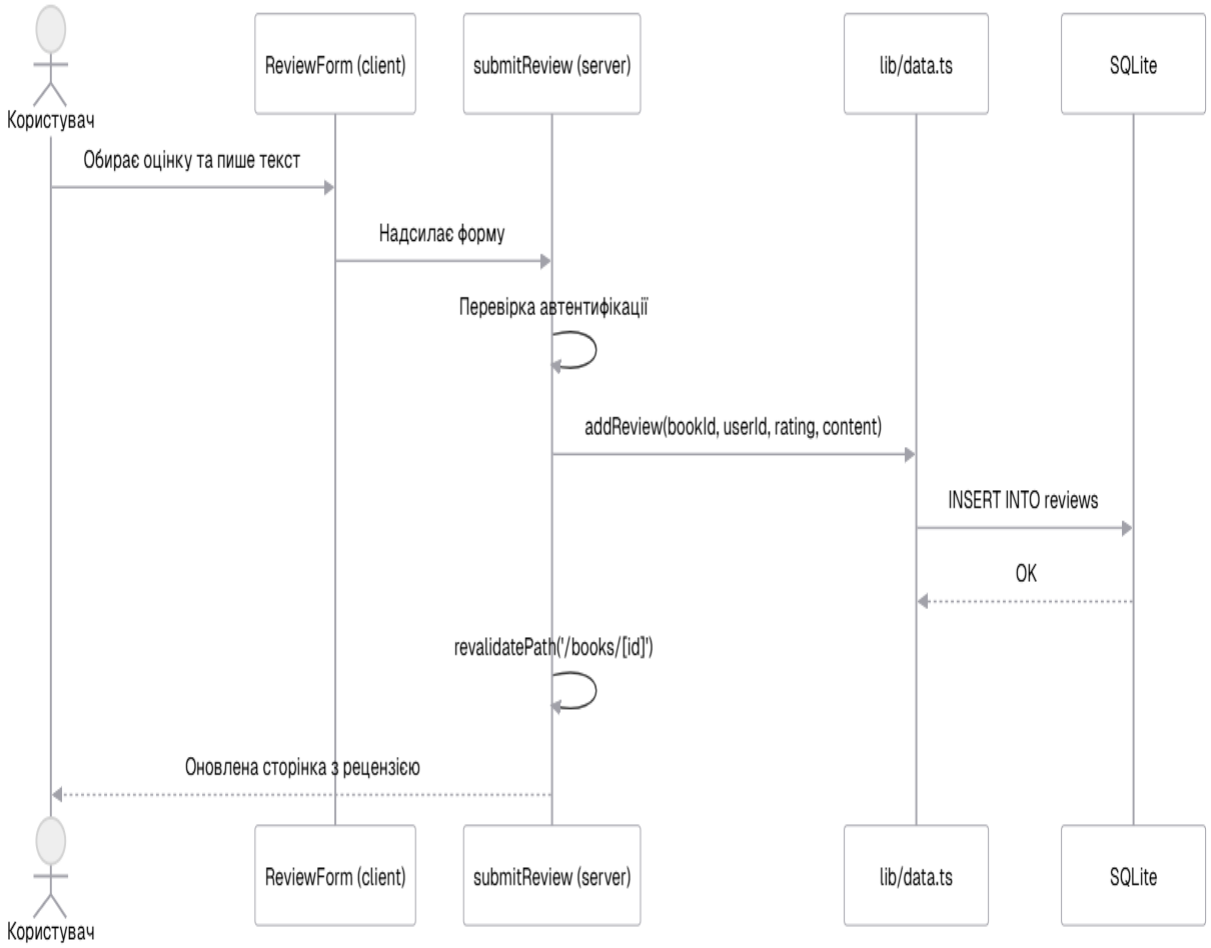


Рисунок 3.1 – Процес створення рецензії

На діаграмі зображено, що користувач починає взаємодію з інтерфейсом форми рецензії, де вводить текст і обирає оцінку. Після натискання кнопки надсилання дані передаються до серверної функції submitReview, яка спочатку виконує перевірку автентифікації користувача. У випадку успішної перевірки відбувається виклик функції доступу до даних addReview (у реалізації – SQL INSERT або UPSERT операція в таблиці reviews). Після успішного запису в базу даних сервер ініціює оновлення кешу сторінки через механізм revalidatePath, що забезпечує автоматичне оновлення списку рецензій без ручного перезавантаження сторінки. Завершальним етапом є повторне відображення оновленого списку рецензій для користувача.

Особливістю реалізації рейтингової системи є те, що середня оцінка книги не зберігається як окреме поле, а обчислюється динамічно під час кожного запиту до бази даних. Це реалізовано за допомогою SQL-агрегаційної функції AVG, що дозволяє отримувати актуальне значення рейтингу на основі всіх наявних рецензій у таблиці reviews.

Такий підхід забезпечує узгодженість даних, оскільки будь-яка зміна (додавання, редагування або видалення рецензії) автоматично впливає на результат обчислення без необхідності додаткового оновлення окремих полів. Крім того, це зменшує ризик виникнення розсинхронізації між фактичними рецензіями та їх агрегованим представленням.

Модуль обговорень реалізує систему коментарів, яка забезпечує більш неформальну взаємодію користувачів під сторінками книг. На відміну від рецензій, коментарі не містять оцінювання, не обмежуються кількістю на одного користувача та формують послідовну стрічку обговорення.

Структурно цей модуль реалізований через таблицю comments, яка містить зв’язки з користувачами та книгами. Отримання коментарів здійснюється функцією getComments, яка виконує SQL-запит із приєднанням таблиці користувачів для відображення авторів повідомлень. Додавання нового коментаря реалізується функцією addComment, яка виконує вставку запису в базу даних.

Користувацькі профілі в системі реалізовані через таблицю users, яка містить основну інформацію про користувача, включаючи ім’я, електронну пошту, роль та додатковий опис. Профіль користувача не є окремою сторінкою редагування складної структури, а інтегрований у загальну систему автентифікації та відображення інтерфейсу.

Візуальне представлення профілю здійснюється через компонент верхньої панелі SiteHeader, де користувач бачить свій аватар (ініціал імені), ім’я та доступ до функції виходу з системи. Розмежування функціональності залежить від ролі користувача, яка зберігається в полі role. Якщо користувач має роль адміністратора, йому додатково відкривається доступ до адміністративного

інтерфейсу, включаючи керування книгами, користувачами та рецензіями (рис. 3.2).

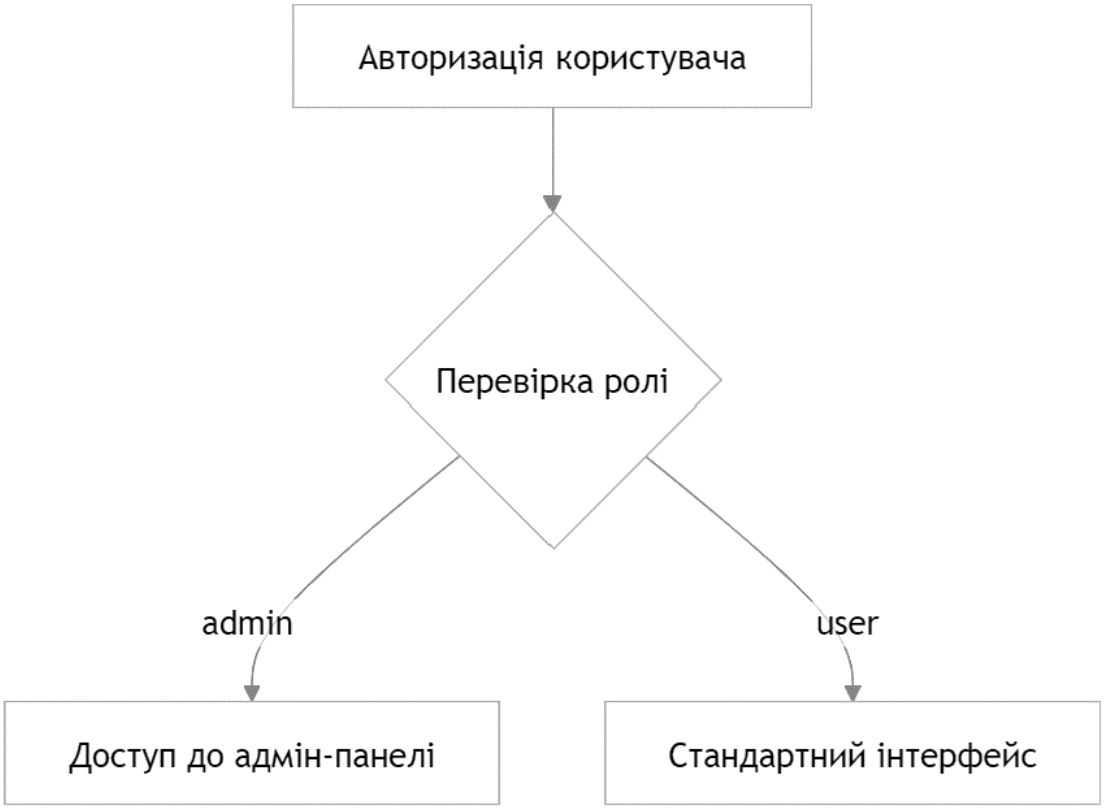


Рисунок 3.2 – Розмежування доступу за ролями

На діаграмі, що відображає структуру доступу на основі ролей, показано розподіл користувачів на дві категорії: звичайний користувач та адміністратор. Після автентифікації система визначає роль користувача та формує відповідний набір доступних функцій. Для звичайного користувача доступними є перегляд книг, створення рецензій та коментарів, тоді як адміністратор отримує розширений набір функцій, що включає управління контентом та користувачами.

Окремим елементом архітектури є захист адміністративної частини системи, реалізований через layout-рівень маршрутизації. Перед рендерингом будь-якої сторінки адміністративного розділу виконується перевірка ролі користувача. У випадку відсутності прав адміністратора система виконує автоматичне перенаправлення на головну сторінку, що унеможливлює доступ до внутрішніх функцій системи через URL.

Отже, реалізовані модулі рецензій, обговорень та користувацьких профілів утворюють ядро соціальної взаємодії системи. Завдяки використанню серверних дій, динамічної агрегації даних та рольової моделі доступу вдалося забезпечити як функціональність, так і безпеку платформи, що є критично важливим для подібних інформаційних систем.

3.3 Тестування системи та аналіз результатів роботи

Тестування програмного комплексу є завершальним етапом його розробки та спрямоване на перевірку коректності реалізації функціональних вимог, стабільності взаємодії між модулями, а також відповідності нефункціональним вимогам, зокрема безпеці, адаптивності та цілісності даних. У межах даного проєкту було застосовано комплексний підхід до тестування, що поєднує функціональну перевірку методом «чорної скриньки», інтеграційне тестування, а також перевірку інтерфейсу та безпекових механізмів у реальних умовах використання.

Основою обраної стратегії стало функціональне тестування методом чорної скриньки, при якому внутрішня реалізація системи не враховується, а перевіряється лише відповідність вхідних і вихідних даних очікуваній поведінці системи. Такий підхід є доцільним для вебзастосунків, оскільки дозволяє оцінити реальний користувацький досвід незалежно від внутрішньої архітектури.

Окрему увагу приділено ручному тестуванню в реальному браузері, що дозволило перевірити коректність роботи інтерфейсу, Server Actions та взаємодії з базою даних у наскрізних сценаріях. Додатково було проведено перевірку граничних випадків, а також сценаріїв із некоректними вхідними даними для оцінки стійкості системи до помилок користувача.

Для перевірки коректності роботи системи було сформовано набір тестових сценаріїв, які охоплюють ключові функціональні можливості застосунку: реєстрацію та автентифікацію користувачів, роботу з каталогом

книг, публікацію рецензій і коментарів, а також функції адміністративного управління.

Тестових сценаріїв – 12 основних перевірок (ТС-01 – ТС-12), кожна з яких відповідає окремому функціональному потоку. Зокрема, сценарії ТС-01–ТС-04 перевіряють процес реєстрації та входу в систему з коректними та некоректними даними, ТС-05 підтверджує правильність перенаправлення адміністратора до відповідної панелі, ТС-06–ТС-07 оцінюють роботу пошуку та фільтрації книг, а ТС-08–ТС-09 перевіряють функціональність створення рецензій і коментарів. Окремо сценарії ТС-10–ТС-12 охоплюють питання безпеки доступу до адміністративної частини, а також коректність каскадного видалення даних.

За результатами тестування всі сценарії були успішно виконані, що свідчить про стабільну роботу основних функціональних модулів системи. Особливо важливим результатом є коректне функціонування механізмів авторизації та розмежування доступу, а також правильне оновлення даних після виконання Server Actions.

Окремим етапом тестування стало дослідження граничних випадків та поведінки системи при виникненні помилкових або нестандартних умов використання. Саме цей етап дозволив виявити низку технічних недоліків, які не проявлялися у стандартних сценаріях.

Зокрема, було зафіксовано проблему рендерингу кнопкових компонентів, що виникала через невідповідність API компонента Button та параметрів його використання. Дефект проявлявся у вигляді некоректної обробки кліку, що впливало на взаємодію користувача з інтерфейсом. Проблему було усунуто шляхом коригування властивостей компонента та додавання параметра nativeButton=false, що забезпечило правильну поведінку елемента.

Іншою суттєвою проблемою стала помилка серіалізації даних, яка виникала під час передачі результатів запитів з бази даних node:sqlite до клієнтських компонентів. Оскільки результати запитів мають об’єкти з null-прототипом, вони не можуть бути напряму передані в React Server Components.

Для усунення цієї проблеми було реалізовано функції plain() та plainAll(), які перетворюють такі об’єкти у звичайні JavaScript-структури.

Також було виявлено помилку завантаження модуля бази даних, пов’язану з тим, що збирач (bundler) намагався обробити вбудований Node.js модуль sqlite, що призводило до помилки виконання require is not defined. Для вирішення цієї проблеми було налаштовано параметр serverExternalPackages, що дозволило виключити відповідний модуль із клієнтської збірки та забезпечити його виконання лише на сервері.

Усі виявлені дефекти були усунуті, після чого повторне тестування підтвердило стабільність роботи системи.

Загальний аналіз результатів тестування свідчить про те, що програмний комплекс функціонує відповідно до визначених вимог і забезпечує коректну взаємодію всіх основних модулів. Особливу увагу було приділено перевірці наскрізних сценаріїв, які охоплюють повний цикл взаємодії користувача із системою – від реєстрації до створення контенту та адміністрування.

Окремо було проведено перевірку нефункціональних характеристик системи, які визначають її якість з точки зору безпеки, продуктивності та зручності використання. Зокрема, перевірка безпеки показала, що паролі користувачів у базі даних зберігаються виключно у вигляді криптографічних хешів із використанням алгоритму scrypt, що унеможливлює їх відновлення у відкритому вигляді.

Перевірка захисту від SQL-ін’єкцій продемонструвала, що використання параметризованих запитів повністю виключає можливість впливу користувача на структуру SQL-команд. Тестування адаптивності інтерфейсу на різних роздільних здатностях екранів (від мобільних до десктопних) підтвердило коректне масштабування компонентів та збереження функціональності.

Додатково перевірено доступність інтерфейсу, зокрема використання семантичної HTML-структури та базових ARIA-атрибутів, що забезпечує базову сумісність із допоміжними технологіями.

У підсумку можна зробити висновок, що проведене тестування підтвердило функціональну повноту та стабільність розробленої системи, а також її відповідність основним вимогам до сучасних вебзастосунків.

3.4 Інструкція користувача

Онлайн-клуб читачів реалізовано як вебзастосунок, доступ до якого здійснюється через сучасний браузер без необхідності встановлення додаткового програмного забезпечення. Інтерфейс системи повністю україномовний, що забезпечує зручність використання та інтуїтивну навігацію між основними функціональними розділами. Користувачі системи поділяються на дві категорії: звичайні користувачі (читачі) та адміністратори, причому кожна категорія має власний набір доступних функцій і рівень прав у межах системи.

Початок роботи з системою для звичайного користувача передбачає відкриття головної сторінки вебзастосунку через браузер. Після завантаження інтерфейсу користувач може ознайомитися з каталогом книг у режимі гостя, однак повноцінна взаємодія з системою стає доступною лише після проходження реєстрації або авторизації. Для створення облікового запису користувач натискає кнопку «Приєднатися», після чого заповнює реєстраційну форму, вказуючи ім’я, електронну пошту та пароль. Після успішного завершення реєстрації система автоматично виконує вхід користувача, створюючи сесію та надаючи доступ до персоналізованого функціоналу. У подальшому для повторного входу використовується кнопка «Увійти», де користувач вводить свої облікові дані, після чого отримує доступ до всіх основних можливостей платформи.

Після авторизації користувач переходить до основного функціонального розділу – каталогу книг, який містить перелік усіх доступних видань, збережених у базі даних. Книги відображаються у вигляді карток із базовою інформацією, включаючи назву, автора та середній рейтинг. Для полегшення пошуку реалізовано механізм текстового пошуку, який дозволяє знаходити книги за

назвою або автором, а також фільтрацію за жанром, що дає змогу звузити вибірку відповідно до інтересів користувача. Після вибору конкретної книги користувач переходить на її детальну сторінку, де відображається повний опис видання, рік публікації, середній рейтинг та відгуки інших читачів, що формує повноцінне інформаційне середовище навколо кожної книги.

Одним із ключових елементів взаємодії користувача із системою є функціонал рецензій. На сторінці книги авторизований користувач отримує доступ до форми створення рецензії, де він може обрати оцінку за п’ятибальною шкалою у вигляді зіркового рейтингу, а також додати текстовий коментар, що відображає його особисту думку щодо прочитаного твору. Після надсилання форми дані передаються на серверну частину системи, де зберігаються у базі даних, а середній рейтинг книги автоматично перераховується з урахуванням нової оцінки. Важливою особливістю є те, що кожен користувач може залишити лише одну рецензію на одну книгу, що забезпечує цілісність даних та запобігає дублюванню оцінок.

Поряд із рецензіями реалізовано модуль обговорень, який дозволяє користувачам брати участь у дискусіях під кожною книгою. Коментарі відображаються у вигляді хронологічної стрічки, що забезпечує природний перебіг обговорення. Для додавання коментаря користувач вводить текст у відповідне поле та надсилає його, після чого повідомлення одразу з’являється у загальному списку. На відміну від рецензій, коментарі не обмежені за кількістю та не впливають на рейтинг книги, що дозволяє підтримувати відкриту та вільну форму спілкування між користувачами.

Окремим елементом системи є користувацький профіль, який створюється автоматично під час реєстрації та містить основні дані користувача, зокрема ім’я, електронну пошту та роль у системі. Профіль відображається в інтерфейсі у верхній частині сторінки та дозволяє користувачу контролювати стан своєї сесії, зокрема виконувати вихід із системи. Розмежування доступу до функціоналу реалізовано на основі ролей, тому звичайні користувачі бачать лише функції

взаємодії з контентом, тоді як адміністраторам додатково відкривається доступ до адміністративної панелі.

Після входу користувач із роллю адміністратора автоматично отримує доступ до розширеної адміністративної панелі, яка є окремим захищеним розділом системи. У цьому розділі реалізовано інструменти керування основними сутностями платформи, зокрема можливість перегляду статистики, додавання нових книг до каталогу, редагування та видалення існуючих записів, а також керування користувачами системи. Окремо адміністратор має доступ до перегляду всіх рецензій, що дозволяє здійснювати модерацію контенту та підтримувати належний рівень якості інформації.

Для тестування та демонстрації функціоналу в системі передбачено адміністративний обліковий запис, який створюється автоматично під час ініціалізації бази даних, тоді як звичайні користувачі створюють власні облікові записи самостійно через форму реєстрації. У процесі експлуатації системи можуть виникати типові ситуації, пов’язані з авторизацією або доступом до функцій, однак більшість із них пояснюється некоректним введенням даних або обмеженнями прав доступу. Зокрема, неможливість входу зазвичай пов’язана з помилками у введенні електронної пошти чи пароля, відсутність можливості залишити рецензію свідчить про необхідність авторизації, а недоступність адміністративної панелі є передбаченим обмеженням для користувачів без відповідної ролі. Крім того, система дозволяє залишати лише одну рецензію на одну книгу від одного користувача, що є частиною логіки забезпечення коректності та узгодженості даних у межах платформи.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було спроєктовано, реалізовано та протестовано програмний комплекс онлайн-клубу читачів, який забезпечує функціонування вебплатформи для взаємодії користувачів навколо літературного контенту. Основною метою роботи було створення цілісної системи, що поєднує каталог книг, механізми рецензування, коментування та адміністративного керування контентом, а також забезпечує зручний користувацький інтерфейс і надійну серверну логіку. У процесі виконання роботи ця мета була повністю досягнута, а поставлені завдання – реалізовані в повному обсязі.

У першому розділі було здійснено теоретичне дослідження предметної області онлайн-спільнот та систем обміну контентом. Розглянуто сутність онлайн-спільнот читачів як сучасного соціального явища, що виникло на перетині цифрових технологій та культурної взаємодії користувачів. Проаналізовано ключові особливості таких систем, зокрема орієнтацію на користувацький контент, важливість механізмів оцінювання та рецензування, а також роль комунікації між учасниками спільноти. Додатково було проведено аналіз сучасних платформ для обміну рецензіями та обговореннями, що дозволило виявити їх типові функціональні можливості, такі як рейтингові системи, персоналізовані профілі, соціальна взаємодія та модерація контенту. На основі цього аналізу були сформульовані вимоги до розроблюваного програмного комплексу, включаючи функціональні вимоги (робота з книгами, рецензіями, коментарями, користувачами) та нефункціональні вимоги (безпека, зручність інтерфейсу, продуктивність та цілісність даних).

У другому розділі виконано проєктування програмного комплексу онлайн-клубу читачів. Було проведено детальний аналіз предметної області, в межах якого визначено основні сутності системи – користувачі, книги, рецензії, коментарі та сесії, а також встановлено зв’язки між ними. На основі цього

сформовано структуру бази даних, яка забезпечує цілісність інформації, підтримує каскадні зв’язки та дозволяє ефективно виконувати запити до даних. Також було розроблено архітектуру програмного забезпечення із чітким поділом на шари: інтерфейс користувача, шар бізнес-логіки та шар доступу до даних. Такий підхід забезпечує модульність системи, спрощує її супровід та подальше розширення. Окрему увагу приділено проєктуванню користувацького інтерфейсу, який орієнтований на інтуїтивну взаємодію та мінімізацію когнітивного навантаження на користувача. Обґрунтовано вибір технологій реалізації, зокрема використання Next.js та вбудованої SQLite-бази даних, що дозволило створити легку, але функціонально повноцінну систему.

У третьому розділі представлено практичну реалізацію програмного комплексу. Реалізовано основні функціональні модулі системи, включаючи каталог книг із можливістю пошуку та фільтрації, систему автентифікації та сесій, а також механізми взаємодії користувачів із контентом. Особливу увагу приділено реалізації соціальних компонентів системи – рецензій та обговорень, які формують ядро користувацької активності. Рецензії реалізовано з урахуванням обмеження «одна рецензія на книгу від одного користувача» та з динамічним перерахунком середнього рейтингу, що забезпечує актуальність даних. Коментарі реалізовано як хронологічну стрічку, що підтримує природний формат дискусії. Окремо реалізовано систему користувацьких профілів та рольову модель доступу, що дозволяє розмежувати функціональність між звичайними користувачами та адміністраторами. У межах тестування системи застосовано функціональний та інтеграційний підходи, що дозволило перевірити коректність взаємодії між клієнтською частиною, серверними діями та базою даних. За результатами тестування підтверджено стабільність роботи системи, коректність реалізації основних сценаріїв використання, а також відповідність нефункціональним вимогам, зокрема щодо безпеки та цілісності даних.

Отже, розроблений програмний комплекс є повноцінною вебплатформою для організації онлайн-клубу читачів і забезпечує реалізацію всіх ключових функцій, необхідних для взаємодії користувачів із літературним контентом.

Система має модульну архітектуру, що дозволяє її подальше розширення без суттєвих змін у базовій структурі.

Перспективами подальшого розвитку можуть бути впровадження рекомендаційних алгоритмів на основі користувацької активності, розширена аналітика поведінки користувачів, інтеграція із зовнішніми літературними API, а також покращення соціальної складової шляхом додавання системи підписок, рейтингу користувачів і розширених механізмів модерації контенту.

ПЕРЕЛІК ПОСИЛАНЬ

1. Інтернет-спільноти. URL: https://cutt.ly/St9Ui7jz (дата звернення: 28.03.2026).

2. Коропецька А. О. Практики формування спільнот в українських медіа : магістерська робота : освітній ступінь магістр, спец. 061 Журналістика / Нац. ун-т «Києво-Могилянська академія», Ф-т соц. наук та соц. технологій, Могилян. шк. журналістики ; керівник Д. Р. Дуцик. Київ, 2025. 71 с.

3. Demchenko M., Myronenko V. Особливості онлайн-комунікації сучасної української бібліотеки (регіональний аспект). Communications and Communicative Technologies. 2023. № 23. С. 30–40. DOI: https://doi.org/10.15421/292304.

4. Goodreads Choice Awards 2024: Statistics and Community Facts. Expanded Ramblings. URL: https://expandedramblings.com/index.php/goodreads-facts-and-statistics/ (дата звернення: 28.03.2026).

5. The Decline of Goodreads: A Missed Opportunity in the Book-Loving Community. eReadersForum. URL: https://www.ereadersforum.com/threads/the-decline-of-goodreads-a-missed-opportunity-in-the-book-loving-community.3306/ (дата звернення: 28.03.2026).

6. James J. Is Goodreads Good? The Catalyst. 2024. 26 лют. URL: https://recatalyst.org/4044/arts/is-goodreads-good/ (дата звернення: 28.03.2026).

7. The StoryGraph. Wikipedia, the free encyclopedia. URL: https://en.wikipedia.org/wiki/The_StoryGraph (дата звернення: 28.03.2026).

8. The StoryGraph: Features and Roadmap. The StoryGraph Roadmap. URL: https://roadmap.thestorygraph.com/features (дата звернення: 28.03.2026).

9. LibraryThing Review: Is It Still the Best Book Cataloging Tool? BookWise. URL: https://bookwiseapp.com/blog/librarything-review-is-it-still-the-best-book-cataloging-tool (дата звернення: 28.03.2026).

10. BookLikes vs LibraryThing: Platform Comparison. SaaSHub. URL: https://www.saashub.com/compare-booklikes-vs-librarything (дата звернення: 28.03.2026).

11. A Beginner’s Guide to Decentralized Social Media – BookWyrm. GelbPhoenix Blog. URL: https://blog.gelbphoenix.de/a-beginners-guide-to-decentralized-social-media-bookwyrm/ (дата звернення: 28.03.2026).

12. How to Start a Virtual Book Club That People Actually Attend. Readfeed Blog. URL: https://www.readfeed.com/blog/start-virtual-book-club/ (дата звернення: 28.03.2026).

13. Goodreads vs. StoryGraph: Which is Better in 2024? Reedsy Discovery. URL: https://reedsy.com/discovery/blog/goodreads-vs-storygraph (дата звернення: 28.03.2026).

14. TypeScript: JavaScript with Syntax for Types. URL: https://www.typescriptlang.org/ (дата звернення: 23.03.2026).

15. Next.js – the React framework. Vercel. URL: https://nextjs.org/ (дата звернення: 28.03.2026).

16. React. React documentation. URL: https://react.dev/ (дата звернення: 28.03.2026).

17. Tailwind CSS Documentation. URL: https://tailwindcss.com/ (дата звернення: 28.03.2026).

18. shadcn/ui Documentation. URL: https://ui.shadcn.com/ (дата звернення: 28.03.2026).

19. SQLite Documentation. URL: https://sqlite.org/docs.html (дата звернення: 28.03.2026).

20. Lucide Icons. Lucide. URL: https://lucide.dev/guide/react/ (дата звернення: 28.03.2026).

ДОДАТОК А

РЕАЛІЗАЦІЯ ОСНОВНОГО ФУНКЦІОНАЛУ СИСТЕМИ

Лістинг А.1 – Ініціалізація бази даних та схеми

```
import { DatabaseSync } from "node:sqlite"
import { randomBytes, scryptSync, timingSafeEqual } from "node:crypto"
import path from "node:path"
import fs from "node:fs"

type DatabaseSyncType = InstanceType<typeof DatabaseSync>

// ---------------------------------------------------------------------------
// SQLite connection (Node.js built-in module, no native compilation required)
// ---------------------------------------------------------------------------

const DATA_DIR = path.join(process.cwd(), "data")
const DB_PATH = path.join(DATA_DIR, "club.db")

// Reuse a single connection across hot-reloads in development.
const globalForDb = globalThis as unknown as { __readersClubDb?:
DatabaseSyncType }

function createConnection(): DatabaseSyncType {
  if (!fs.existsSync(DATA_DIR)) {
    fs.mkdirSync(DATA_DIR, { recursive: true })
  }
  const connection = new DatabaseSync(DB_PATH)
  connection.exec("PRAGMA journal_mode = WAL;")
  connection.exec("PRAGMA foreign_keys = ON;")
  return connection
}

export const db: DatabaseSyncType = globalForDb.__readersClubDb ??
createConnection()
if (process.env.NODE_ENV !== "production") {
  globalForDb.__readersClubDb = db
}
```

```
// node:sqlite returns rows with a null prototype, which cannot be serialized
// and passed from Server Components to Client Components. Convert them into
// plain objects before they leave the data layer.
export function plain<T>(row: T): T {
  return row == null ? row : ({ ...row } as T)
}

export function plainAll<T>(rows: T[]): T[] {
  return rows.map((r) => ({ ...r }) as T)
}

// ---------------------------------------------------------------------
// Schema
// ---------------------------------------------------------------------

let initialized = false

export function initDb() {
  if (initialized) return
  db.exec(`
    CREATE TABLE IF NOT EXISTS users (
      id          INTEGER PRIMARY KEY AUTOINCREMENT,
      name        TEXT NOT NULL,
      email       TEXT NOT NULL UNIQUE,
      password    TEXT NOT NULL,
      role        TEXT NOT NULL DEFAULT 'user',
      bio         TEXT NOT NULL DEFAULT '',
      created_at  TEXT NOT NULL DEFAULT (datetime('now'))
    );

    CREATE TABLE IF NOT EXISTS sessions (
      token       TEXT PRIMARY KEY,
      user_id     INTEGER NOT NULL,
      created_at  TEXT NOT NULL DEFAULT (datetime('now')),
      FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
    );

    CREATE TABLE IF NOT EXISTS books (
      id          INTEGER PRIMARY KEY AUTOINCREMENT,
      title       TEXT NOT NULL,
      author      TEXT NOT NULL,
      description TEXT NOT NULL DEFAULT '',
      genre       TEXT NOT NULL DEFAULT '',
      cover_url   TEXT NOT NULL DEFAULT '',
      year        INTEGER,
```

```
    created_at  TEXT NOT NULL DEFAULT (datetime('now'))
  );

  CREATE TABLE IF NOT EXISTS reviews (
    id         INTEGER PRIMARY KEY AUTOINCREMENT,
    book_id    INTEGER NOT NULL,
    user_id    INTEGER NOT NULL,
    rating     INTEGER NOT NULL,
    body       TEXT NOT NULL DEFAULT '',
    created_at  TEXT NOT NULL DEFAULT (datetime('now')),
    FOREIGN KEY (book_id) REFERENCES books(id) ON DELETE CASCADE,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE,
    UNIQUE (book_id, user_id)
  );

  CREATE TABLE IF NOT EXISTS comments (
    id         INTEGER PRIMARY KEY AUTOINCREMENT,
    book_id    INTEGER NOT NULL,
    user_id    INTEGER NOT NULL,
    body       TEXT NOT NULL,
    created_at  TEXT NOT NULL DEFAULT (datetime('now')),
    FOREIGN KEY (book_id) REFERENCES books(id) ON DELETE CASCADE,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE CASCADE
  );
 `)
 initialized = true
 seed()
}

// ---------------------------------------------------------------------------
// Password hashing (scrypt via node:crypto – no external dependency)
// ---------------------------------------------------------------------------

export function hashPassword(password: string): string {
 const salt = randomBytes(16).toString("hex")
 const hash = scryptSync(password, salt, 64).toString("hex")
 return `${salt}:${hash}`
}

export function verifyPassword(password: string, stored: string): boolean {
 const [salt, hash] = stored.split(":")
 if (!salt || !hash) return false
 const hashBuffer = Buffer.from(hash, "hex")
 const candidate = scryptSync(password, salt, 64)
```

```
  return hashBuffer.length === candidate.length && timingSafeEqual(hashBuffer,
candidate)
}

// ---------------------------------------------------------------------------
// Seed data – default admin + sample books
// ---------------------------------------------------------------------------

function seed() {
  const row = db.prepare("SELECT COUNT(*) AS c FROM users").get() as { c:
number }
  if (row.c === 0) {
    db.prepare("INSERT INTO users (name, email, password, role, bio) VALUES (?,
?, ?, ?, ?)").run(
      "Адміністратор",
      "admin@club.ua",
      hashPassword("admin123"),
      "admin",
      "Засновник клубу читачів.",
    )
    db.prepare("INSERT INTO users (name, email, password, role, bio) VALUES (?,
?, ?, ?, ?)").run(
      "Олена Книголюб",
      "olena@club.ua",
      hashPassword("user123"),
      "user",
      "Люблю класику та сучасну прозу.",
    )
  }

  const booksCount = db.prepare("SELECT COUNT(*) AS c FROM books").get() as
{ c: number }
  if (booksCount.c === 0) {
    const insert = db.prepare(
      "INSERT INTO books (title, author, description, genre, cover_url, year)
VALUES (?, ?, ?, ?, ?, ?)",
    )
    const sample: Array<[string, string, string, string, string, number]> = [
      [
        "Тіні забутих предків",
        "Михайло Коцюбинський",
        "Поетична повість про кохання Івана та Марічки на тлі гуцульських
звичаїв і вірувань Карпат.",
        "Класика",
        "/book-shadows-of-forgotten-ancestors.png",
```

```
      1911,
    ],
    [
      "Місто",
      "Валер’ян Підмогильний",
      "Урбаністичний роман про молоду людину, яка приїжджає підкорювати Київ 1920-х років.",
      "Класика",
      "/book-the-city-modernist.png",
      1928,
    ],
    [
      "Інтернат",
      "Сергій Жадан",
      "Історія вчителя, який пробирається крізь охоплений війною Донбас, щоб забрати племінника.",
      "Сучасна проза",
      "/book-the-orphanage-war-novel.png",
      2017,
    ],
    [
      "Польові дослідження з українського сексу",
      "Оксана Забужко",
      "Сповідальний роман про жіночу ідентичність, любов і національну пам’ять.",
      "Сучасна проза",
      "/book-field-research-introspective.png",
      1996,
    ],
    [
      "Солодка Даруся",
      "Марія Матіос",
      "Драма буковинського села крізь долю жінки, скаліченої тоталітарною добою.",
      "Драма",
      "/book-sweet-darusya-village-drama.png",
      2004,
    ],
    [
      "Лісова пісня",
      "Леся Українка",
      "Драма-феєрія про кохання Мавки й Лукаша, про красу природи та зраду мрії.",
      "Драма",
      "/book-forest-song-fairy-drama.png",
```

```
      1911,
    ],
  ]
  for (const b of sample) insert.run(...b)
 }
}
```

Лістинг А.2 – Дані бази

```
import { db, initDb, plain, plainAll } from "./db"

export type Book = {
 id: number
 title: string
 author: string
 description: string
 genre: string
 cover_url: string
 year: number | null
 created_at: string
}

export type BookWithStats = Book & {
 avg_rating: number
 review_count: number
}

export type Review = {
 id: number
 book_id: number
 user_id: number
 rating: number
 body: string
 created_at: string
 user_name: string
}

export type Comment = {
 id: number
 book_id: number
 user_id: number
```

```
  body: string
  created_at: string
  user_name: string
  user_role: string
}

export function listBooks(search = "", genre = ""): BookWithStats[] {
  initDb()
  let sql = `
    SELECT b.*,
      COALESCE(ROUND(AVG(r.rating), 1), 0) AS avg_rating,
      COUNT(r.id) AS review_count
    FROM books b
    LEFT JOIN reviews r ON r.book_id = b.id
  `
  const where: string[] = []
  const params: (string | number)[] = []
  if (search) {
    where.push("(b.title LIKE ? OR b.author LIKE ?)")
    params.push(`%${search}%`, `%${search}%`)
  }
  if (genre) {
    where.push("b.genre = ?")
    params.push(genre)
  }
  if (where.length) sql += ` WHERE ${where.join(" AND ")}`
  sql += " GROUP BY b.id ORDER BY b.created_at DESC"
  return plainAll(db.prepare(sql).all(...params) as BookWithStats[])
}

export function listGenres(): string[] {
  initDb()
  const rows = db
    .prepare("SELECT DISTINCT genre FROM books WHERE genre != '' ORDER
BY genre")
    .all() as { genre: string }[]
  return rows.map((r) => r.genre)
}

export function getBook(id: number): BookWithStats | null {
  initDb()
  const row = db
    .prepare(
      `SELECT b.*,
        COALESCE(ROUND(AVG(r.rating), 1), 0) AS avg_rating,
```

```
      COUNT(r.id) AS review_count
     FROM books b LEFT JOIN reviews r ON r.book_id = b.id
     WHERE b.id = ? GROUP BY b.id`,
   )
   .get(id) as BookWithStats | undefined
 return row ? plain(row) : null
}

export function getReviews(bookId: number): Review[] {
 initDb()
 const rows = db
   .prepare(
    `SELECT r.*, u.name AS user_name
     FROM reviews r JOIN users u ON u.id = r.user_id
     WHERE r.book_id = ? ORDER BY r.created_at DESC`,
   )
   .all(bookId) as Review[]
 return plainAll(rows)
}

export function getUserReview(bookId: number, userId: number): Review | null {
 initDb()
 const row = db
   .prepare(
    `SELECT r.*, u.name AS user_name FROM reviews r JOIN users u ON u.id =
r.user_id
     WHERE r.book_id = ? AND r.user_id = ?`,
   )
   .get(bookId, userId) as Review | undefined
 return row ? plain(row) : null
}

export function upsertReview(bookId: number, userId: number, rating: number, body:
string) {
 initDb()
 db.prepare(
   `INSERT INTO reviews (book_id, user_id, rating, body) VALUES (?, ?, ?, ?)
    ON CONFLICT (book_id, user_id) DO UPDATE SET rating = excluded.rating,
body = excluded.body, created_at = datetime('now')`,
 ).run(bookId, userId, rating, body)
}

export function deleteReview(id: number) {
 initDb()
 db.prepare("DELETE FROM reviews WHERE id = ?").run(id)
```

```
}

export function getComments(bookId: number): Comment[] {
  initDb()
  const rows = db
    .prepare(
      `SELECT c.*, u.name AS user_name, u.role AS user_role
       FROM comments c JOIN users u ON u.id = c.user_id
       WHERE c.book_id = ? ORDER BY c.created_at ASC`,
    )
    .all(bookId) as Comment[]
  return plainAll(rows)
}

export function addComment(bookId: number, userId: number, body: string) {
  initDb()
  db.prepare("INSERT INTO comments (book_id, user_id, body) VALUES (?, ?,
?)").run(bookId, userId, body)
}

export function deleteComment(id: number) {
  initDb()
  db.prepare("DELETE FROM comments WHERE id = ?").run(id)
}

// ---- Admin helpers ----

export function createBook(data: {
  title: string
  author: string
  description: string
  genre: string
  cover_url: string
  year: number | null
}) {
  initDb()
  db.prepare(
    "INSERT INTO books (title, author, description, genre, cover_url, year) VALUES
(?, ?, ?, ?, ?, ?)",
  ).run(data.title, data.author, data.description, data.genre, data.cover_url, data.year)
}

export function updateBook(
  id: number,
```

```
  data: { title: string; author: string; description: string; genre: string; cover_url:
string; year: number | null },
) {
  initDb()
  db.prepare(
    "UPDATE books SET title = ?, author = ?, description = ?, genre = ?, cover_url =
?, year = ? WHERE id = ?",
  ).run(data.title, data.author, data.description, data.genre, data.cover_url, data.year,
id)
}

export function deleteBook(id: number) {
  initDb()
  db.prepare("DELETE FROM books WHERE id = ?").run(id)
}

export function adminStats() {
  initDb()
  const users = (db.prepare("SELECT COUNT(*) AS c FROM users").get() as { c:
number }).c
  const books = (db.prepare("SELECT COUNT(*) AS c FROM books").get() as { c:
number }).c
  const reviews = (db.prepare("SELECT COUNT(*) AS c FROM reviews").get() as {
c: number }).c
  const comments = (db.prepare("SELECT COUNT(*) AS c FROM
comments").get() as { c: number }).c
  return { users, books, reviews, comments }
}

export type AdminUser = {
  id: number
  name: string
  email: string
  role: string
  created_at: string
  review_count: number
  comment_count: number
}

export function listUsers(): AdminUser[] {
  initDb()
  const rows = db
    .prepare(
      `SELECT u.id, u.name, u.email, u.role, u.created_at,
```

```
      (SELECT COUNT(*) FROM reviews r WHERE r.user_id = u.id) AS
review_count,
      (SELECT COUNT(*) FROM comments c WHERE c.user_id = u.id) AS
comment_count
     FROM users u ORDER BY u.created_at DESC`,
   )
   .all() as AdminUser[]
 return plainAll(rows)
}

export function setUserRole(id: number, role: "admin" | "user") {
 initDb()
 db.prepare("UPDATE users SET role = ? WHERE id = ?").run(role, id)
}
export function deleteUser(id: number) {
 initDb()
 db.prepare("DELETE FROM users WHERE id = ?").run(id)
}
export type AdminReview = Review & { book_title: string }
export function listAllReviews(): AdminReview[] {
 initDb()
 const rows = db
   .prepare(
     `SELECT r.*, u.name AS user_name, b.title AS book_title
     FROM reviews r JOIN users u ON u.id = r.user_id JOIN books b ON b.id =
r.book_id
     ORDER BY r.created_at DESC`,
   )
   .all() as AdminReview[]
 return plainAll(rows)
}
```

Лістинг А.3 – Реалізація каталогу

```
import Link from "next/link"
import { Search } from "lucide-react"
import { listBooks, listGenres } from "@/lib/data"
import { SiteHeader } from "@/components/site-header"
import { BookCard } from "@/components/book-card"
import { Input } from "@/components/ui/input"
import { Badge } from "@/components/ui/badge"
export default async function BooksPage({
```

```
  searchParams,
}: {
  searchParams: Promise<{ q?: string; genre?: string }>
}) {
  const { q = "", genre = "" } = await searchParams
  const books = listBooks(q, genre)
  const genres = listGenres()

  return (
    <div className="min-h-svh bg-background">
      <SiteHeader />
      <main className="mx-auto w-full max-w-6xl px-4 py-8">
        <div className="mb-6">
          <h1 className="font-serif text-3xl font-bold text-foreground">Каталог
книг</h1>
          <p className="mt-1 text-muted-foreground">
            Обирайте книгу, читайте рецензії спільноти та долучайтеся до
обговорень.
          </p>
        </div>

        <form className="mb-6 flex flex-col gap-4">
          <div className="relative max-w-md">
            <Search className="absolute left-3 top-1/2 size-4 -translate-y-1/2 text-
muted-foreground" />
            <Input
              name="q"
              defaultValue={q}
              placeholder="Пошук за назвою або автором..."
              className="pl-9"
            />
            {genre && <input type="hidden" name="genre" value={genre} />}
          </div>
          <div className="flex flex-wrap items-center gap-2">
            <Link href="/books">
              <Badge variant={genre ? "secondary" : "default"} className="cursor-
pointer">
                Усі жанри
              </Badge>
            </Link>
            {genres.map((g) => (
              <Link key={g} href={`/books?genre=${encodeURIComponent(g)}${q ?
`&q=${encodeURIComponent(q)}` : ""}`}>
                <Badge variant={genre === g ? "default" : "secondary"}
className="cursor-pointer">
```

```
                {g}
              </Badge>
            </Link>
          ))}
        </div>
      </form>
      {books.length === 0 ? (
        <p className="rounded-xl border border-dashed border-border py-16 text-
center text-muted-foreground">
          Книг не знайдено. Спробуйте змінити запит.
        </p>
      ) : (
        <div className="grid grid-cols-2 gap-4 sm:grid-cols-3 lg:grid-cols-4">
          {books.map((book) => (
            <BookCard key={book.id} book={book} />
          ))}
        </div>
      )}
    </main>
  </div>
 )
}
```

ДОДАТОК Б

ІНТЕРФЕЙС ВЕБЗАСТОСУНКУ

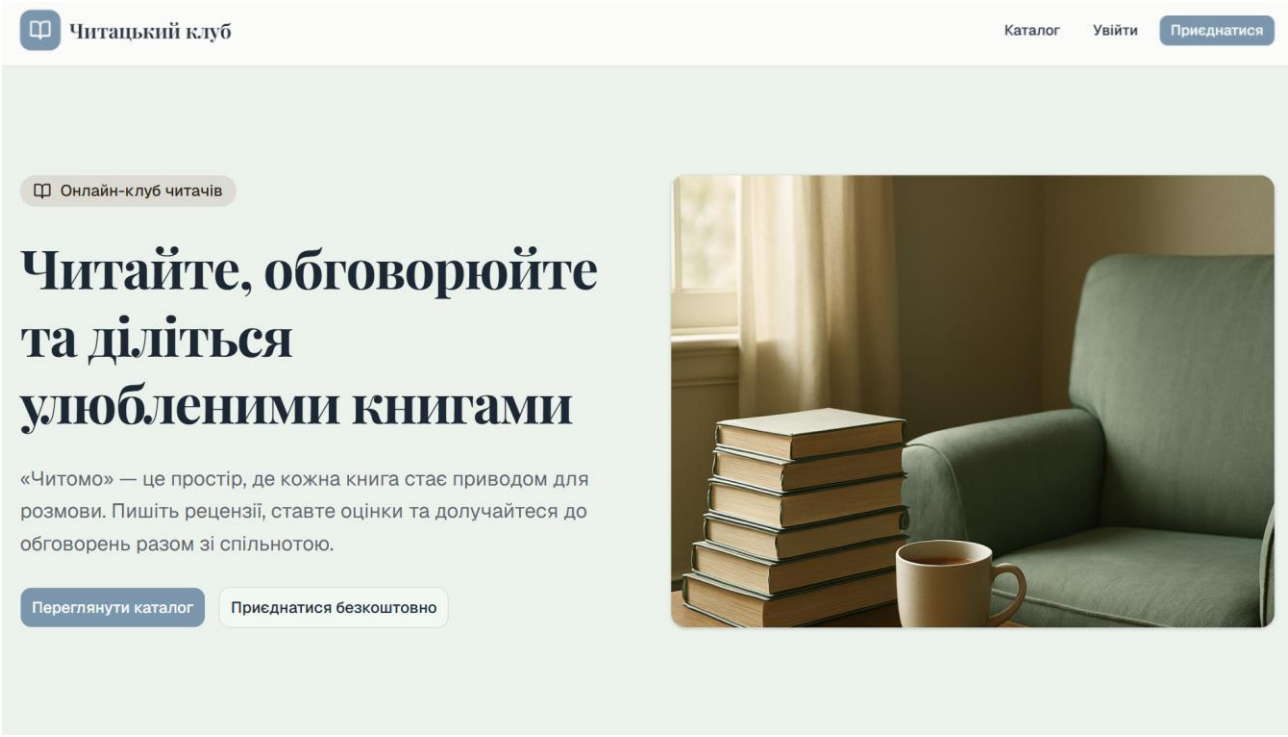


Рисунок Б.1 – Головна сторінка

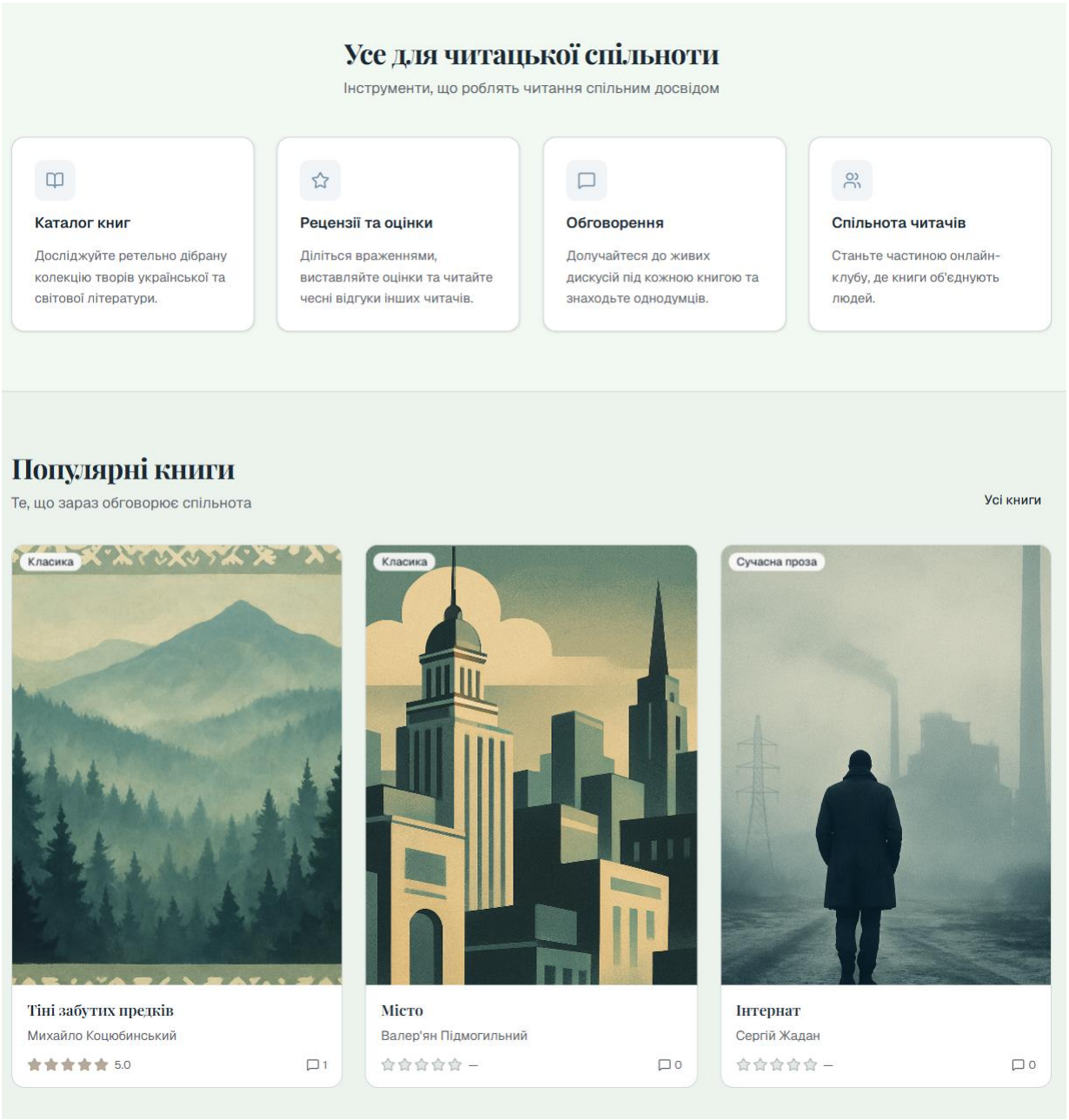


Рисунок Б.2 – Головна сторінка (2)

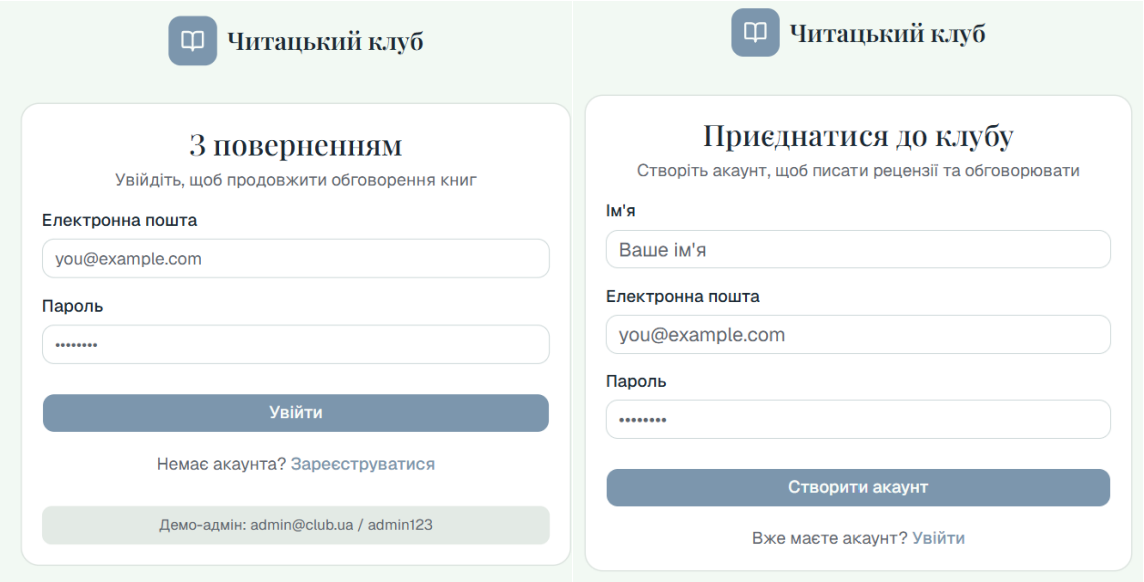


Рисунок Б.3 – Сторінка входу/реєстрації

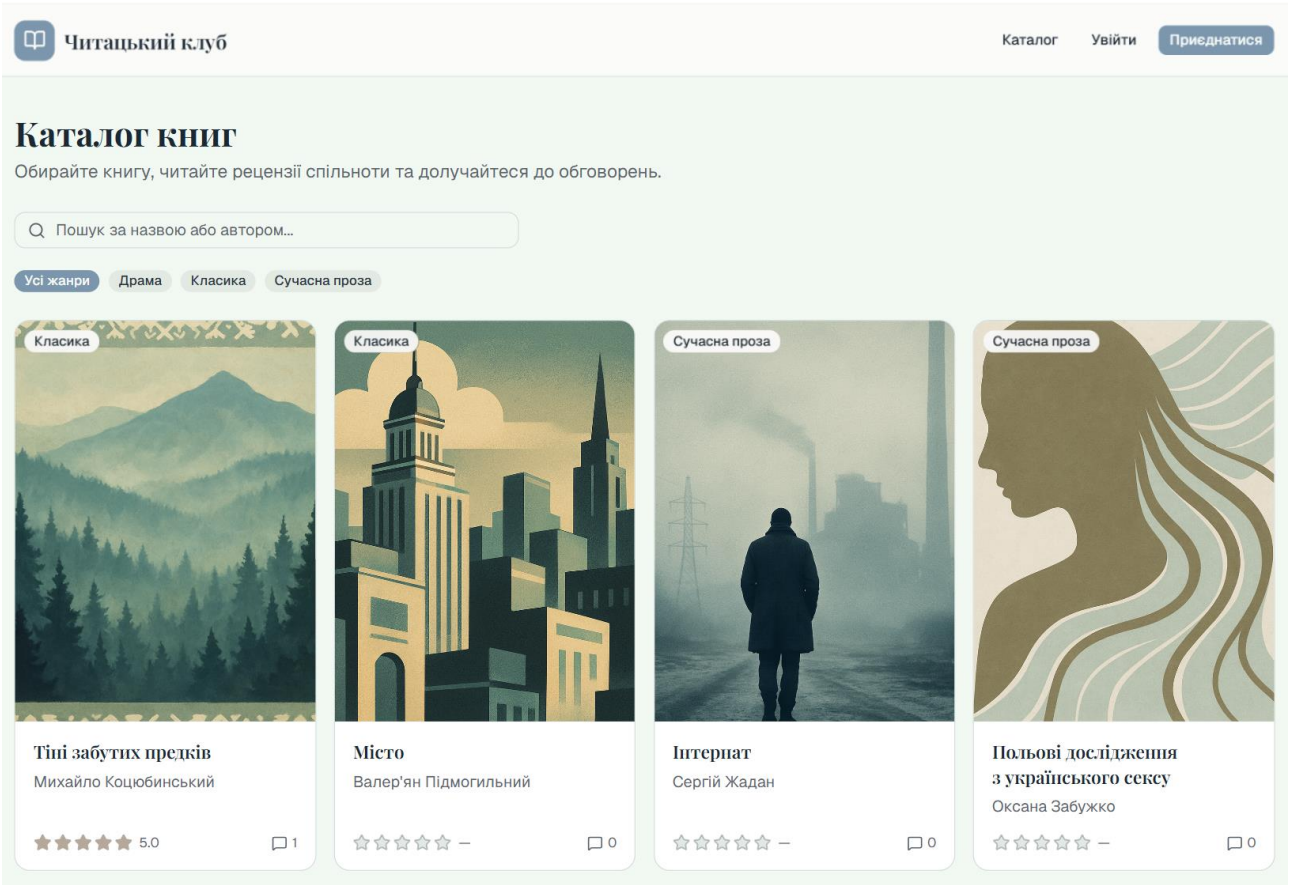


Рисунок Б.4 – Каталог

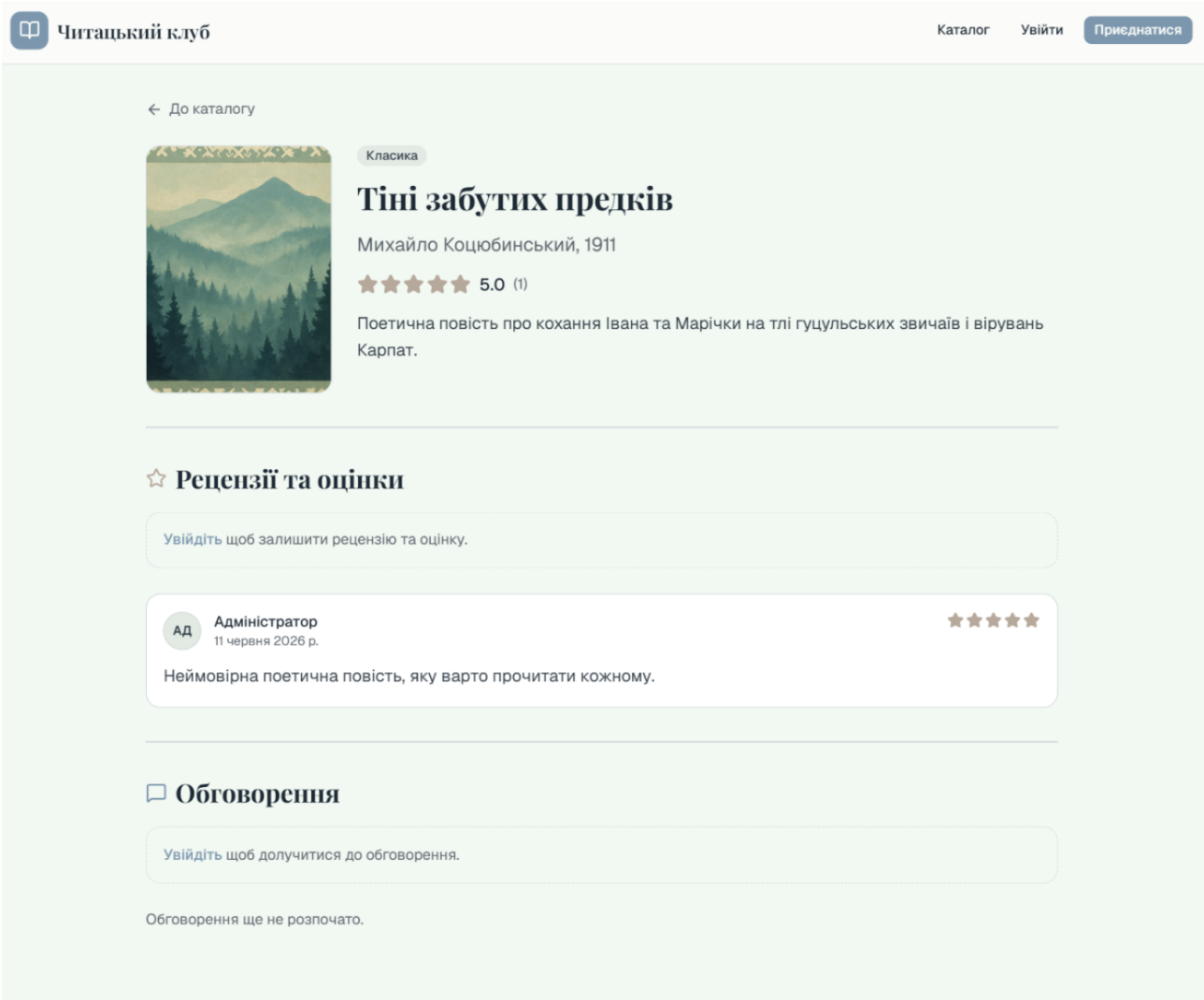


Рисунок Б.5 – Дані про книгу

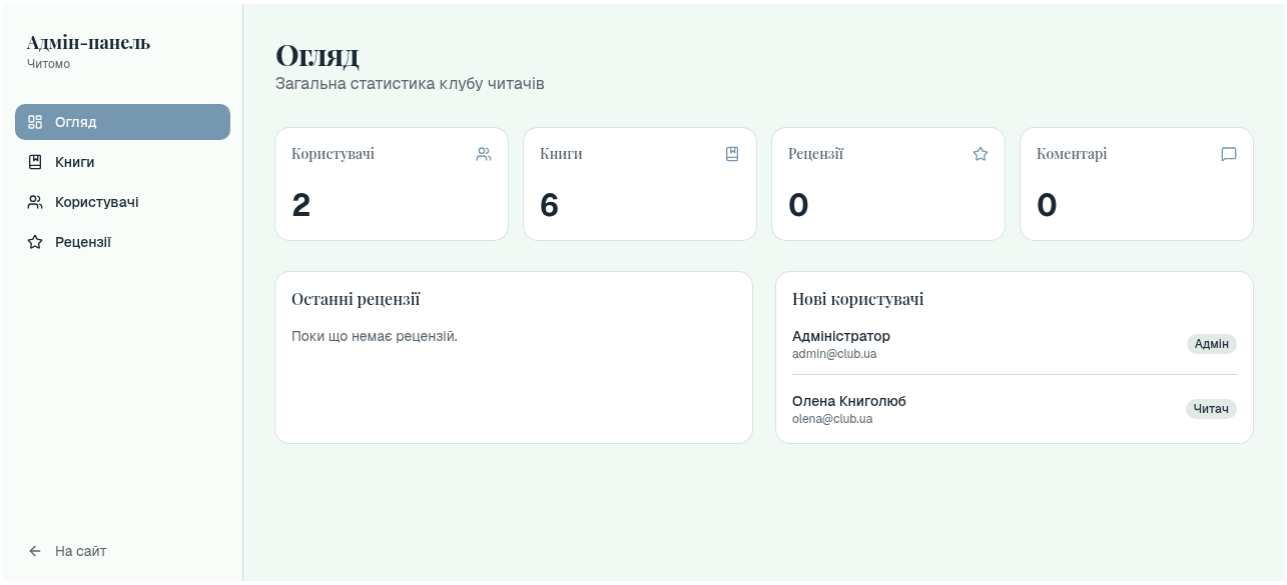


Рисунок Б.6 – Адмін-панель

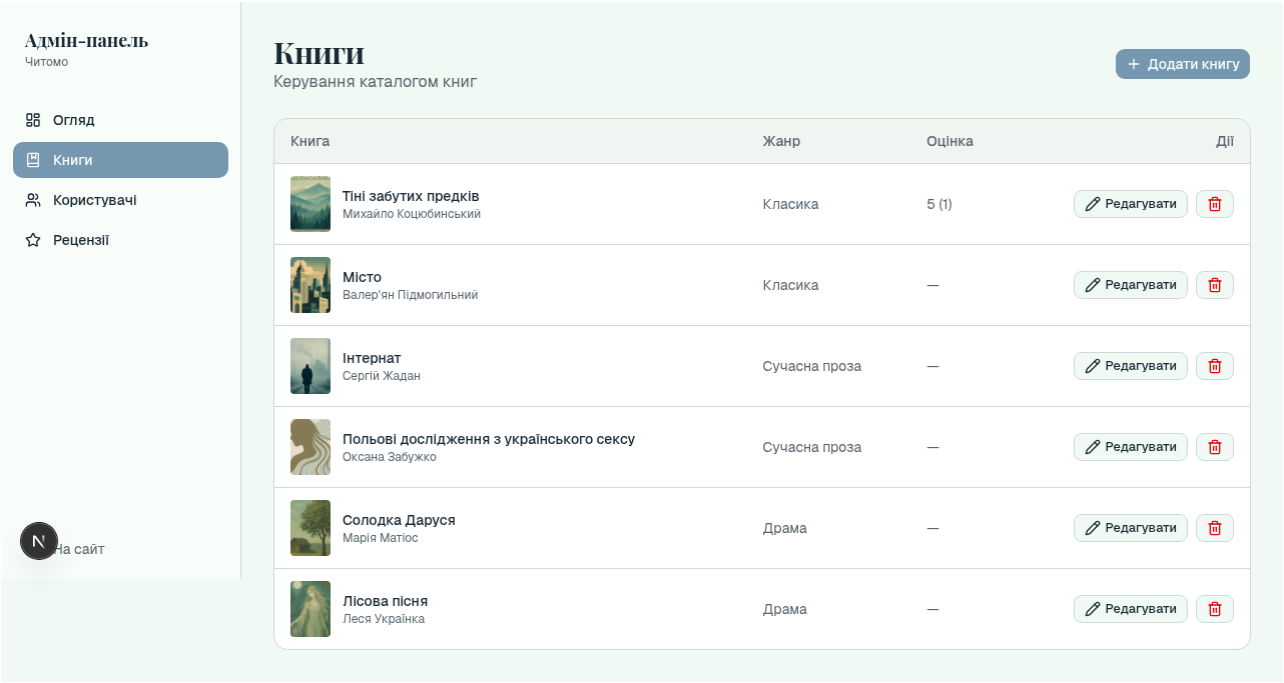


Рисунок Б.6 – Сторінка книг в адміністратора

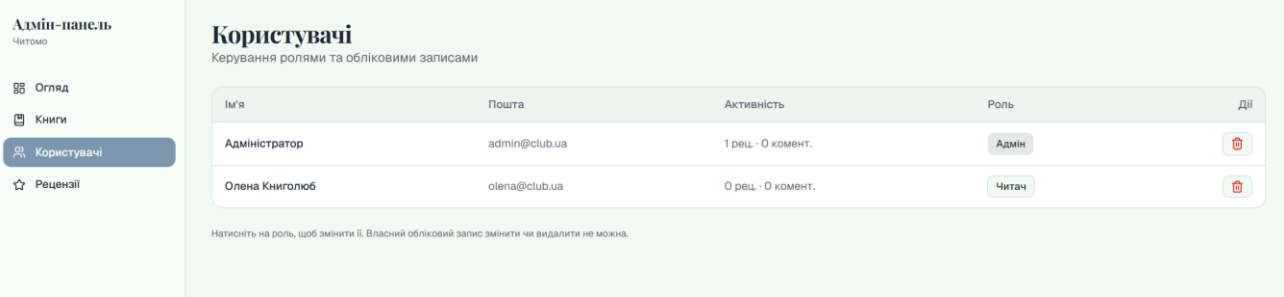


Рисунок Б.7 –Сторінка користувачів в адміністратора

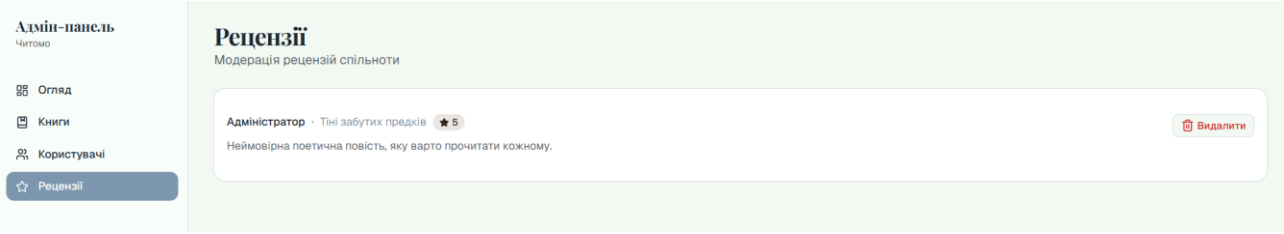


Рисунок Б.8 – Сторінка рецензії в адміністратора