

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики

Розробка веб-сайту для самвидаву художньої літератури
Текстова частина до курсової роботи за спеціальністю «Інженерія
програмного забезпечення» - 121

Керівник курсової роботи
Старший викладач Борозенний С.О.

(підпис)

“ ___ ” _____ 2023 року

Виконав студент ІПЗ-3

Калінбет В.В.

“ ___ ” _____ 2023 року

Київ 2023

Тема: Розробка веб-сайту для самвидаву художньої літератури

Календарний план виконання роботи:

№	Назва етапу роботи	Термін виконання етапу	Примітка
1.	Отримання теми курсової роботи.	30.01.2023	
2.	Дослідження предметної області.	лютий 2023 р.	
3.	Вивчення та дослідження інструментів розробки.	лютий – березень 2023 р.	
4.	Створення практичної частини роботи.	27.03.2023 – 02.04.2023	
5.	Написання пояснювальної частини роботи.	02.04.2023 – 09.04.2023	
6.	Надання роботи керівнику для перевірки.	квітень 2023 р.	
7.	Створення презентації.	16.05.2023 – 17.05.2023	

Студент Калінбет В. В.

Керівник Борозенний С.О.

“ ”

Зміст

Вступ	4
Розділ 1. Розгляд існуючих платформ	5
1.1 Платформа RoyalRoad	5
1.2 Платформа ScribbleHub	6
1.3 Аналіз існуючих рішень. Актуальність платформ.	7
Розділ 2. Розгляд застосованих технологій та інструментів розробки.....	8
2.1 IDE WebStorm.....	8
2.2 MySQL Workbench.....	8
2.3 Node.js	8
2.4 Express.js	9
2.5 Бібліотека node.bcrypt.js	9
2.7 EJS	10
2.8 Twitter Bootstrap	10
Розділ 3. Реалізація веб-сайту.....	11
3.1 Вимоги до проєкту.....	11
3.2 Опис реалізованого функціоналу сторінок веб-сайту.....	12
3.3 Опис структури даних проєкту	25
3.4 Розгляд програмних рішень в проєкті	27
3.4.1 Архітектура.....	27
3.4.2 Авторизація.....	28
3.4.3 Рівень відображення	29
Висновки	32
Список використаних джерел.....	33

Вступ

Для початку успішної кар'єри в письменницькій галузі сучасним письменникам-початківцям доводиться проходити через складний процес пошуку видавця, що готовий піти на ризик співпрацювати з новим, нікому не відомим автором. Цей процес за своєю суттю суб'єктивний, адже невелика редакторська колегія вирішує, чи має перспективи даний початківець. Через це, незважаючи на переважну досвідченість видавничих редакційних колегій, залишається немалий шанс, що потенціал молодого письменника не помітять і залишать перспективного автора без шансу на знаходження своєї аудиторії.

Веб-сайт для самвидаву художньої літератури допоможе прибрати посередника між зацікавленою аудиторією та автором, готовим запропонувати свої доробки широкому загалу. Після публікації своєї творчості на такій платформі автор отримує змогу знайти зацікавлених споживачів та використовувати це як для продовження самостійної кар'єри, так і в якості аргументу у переговорах із потенційними видавцями. Такий веб-сайт також може допомогти знаходити охочі до співпраці таланти й видавництвам, які можуть спиратися на сприйняття аудиторією веб-сайту книг автора, як на більш об'єктивне мірило потенційного сприйняття традиційно виданих книг автора.

Node.js – оточення для виконання JavaScript, для створення веб-застосунків, яке за даними Google Trends залишається одним з найпопулярніших інструментів в цій галузі протягом більш ніж 5 років. MySQL – одна з найпопулярніших СКБД серед новачків, за даними Stack Overflow 2022 Developer Survey. Через це було прийнято рішення вибрати саме ці інструменти розробки [1] [2].

Мета цієї роботи – удосконалення навичок роботи з обраними основними інструментами та супутніми ним веб-технологіями.

Розділ 1. Розгляд існуючих платформ

1.1 Платформа RoyalRoad

RoyalRoad – веб-сайт, спрямований на публікацію веб-новел та фан-фікшн. Користувачам пропонується широкий функціонал та зручний інтерфейс, окрім очевидної можливості переглянути та прочитати опубліковані книги, веб-сайт пропонує широкий спектр інших дій для користувача.

Існують декілька способів знайти цікаву вам книгу:

- згенеровані рекомендації;
- останні оновлення;
- найкраще оцінені книги;
- пошук за назвою книги;
- розширений пошук за декількома критеріями.

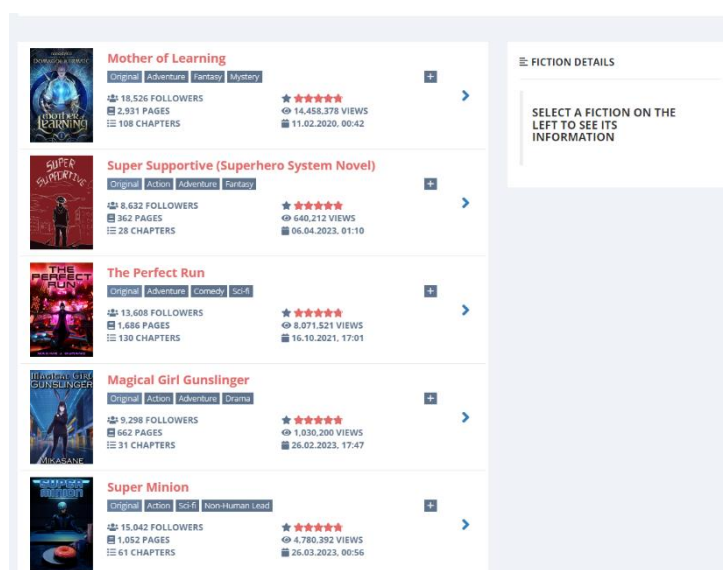


Рис. 1.1.1.1 Рекомендації на сайті RoyalRoad

Автори книг мають можливість публікувати книги та глави до них.

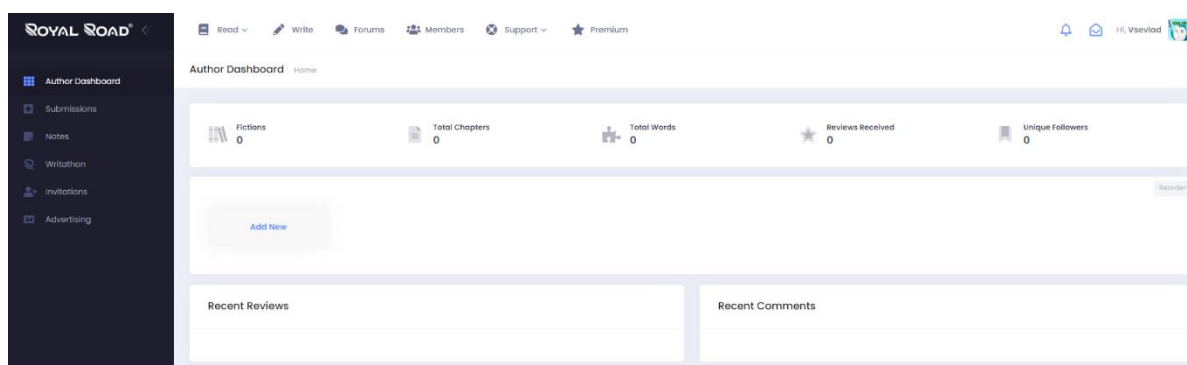


Рис. 1.1.1 Сторінка публікації нової книги на RoyalRoad

Усі користувачі також мають можливість:

- коментувати глави книг;
- писати рецензії до книг;
- оцінювати книгу від 1 до 5;
- У разі, коли користувач вбачає щось некоректне у опублікованому змісті коментаря, глави, тощо, він має змогу оформити скаргу на це модерації сайту.

1.2 Платформа ScribbleHub

Як і попередня платформа, ScribbleHub орієнтований на авторів веб-новел та фан-фікшн. Основний функціонал платформи майже повністю повторює свого конкурента. Як і на RoyalRoad існують декілька шляхів для знаходження цікавої вам книги, присутній пошук за назвою та за розширеними критеріями, користувачі так само мають можливість вільно публікувати свої книги, коментарі до глав, рецензії та оцінювати опубліковані книги. Так само присутній і механізм оформлення скарг. Основна різниця полягає у деталях реалізації певних систем веб-сайту. Наприклад, для публікації книги, ScribbleHub не вимагає одразу публікувати хоча б одну главу до неї. Певні розбіжності містяться і у

критеріях за якими користувачі можуть шукати книги, а автори, відповідно, покращувати шанси зацікавленої аудиторії знаходити їхні книги.

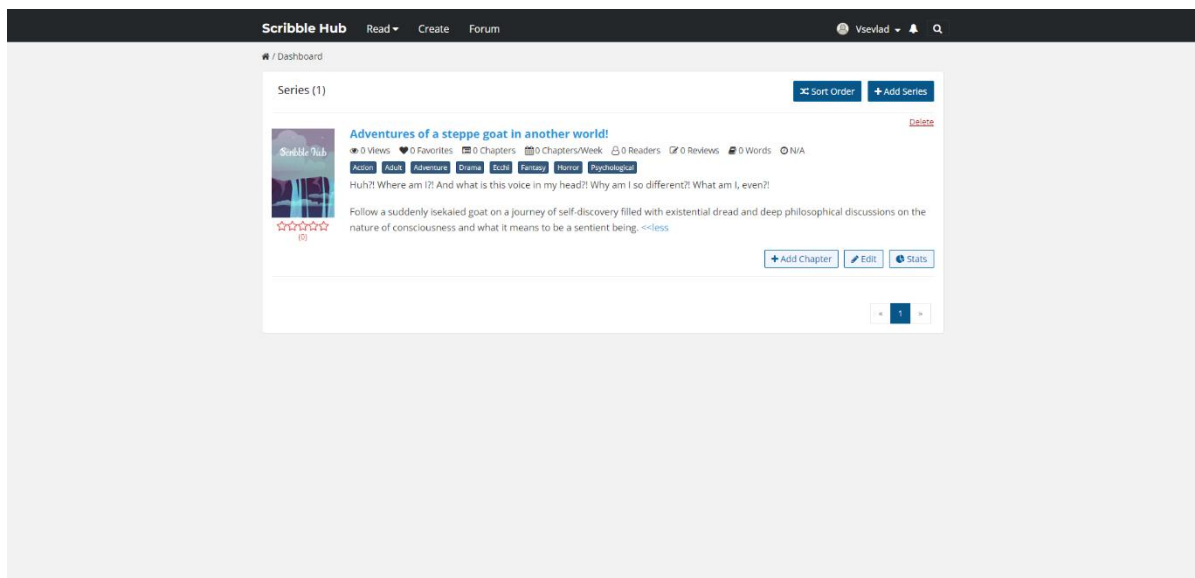


Рис. 1.1.2.1 Сторінка опублікованих користувачем книг на ScribbleHub

1.3 Аналіз існуючих рішень. Актуальність платформ.

Загальна актуальність подібних сайтів у наш час складно переоцінити, окрім згаданої раніше здатності позбавити читача та автора посередника у формі видавця, дані платформи також розширюють творчі можливості авторів, адже тут вони не обмежені ні в об'ємі, ні в темпах та формі публікації своєї творчості. Низький поріг входу також дозволяє ділитися найбільш нішевими доробками.

Розгляд наявних рішень показує, що великий об'єм функціоналу дублюється між платформами. Тим не менш, кожна з них має свою стабільну аудиторію і успішно співіснує із конкурентами. З цього можна зробити висновок, що для успішного існування подібних платформ, деталі, із якими виконаний цей функціонал, а також конкретні рішення модераторів та творців платформи щодо деталей інструментів доступних авторам для поширення своєї творчості мають велике значення. Прийнято

рішення реалізувати у програмному продукті найбільш поширений функціонал даних платформ.

Розділ 2. Розгляд застосованих технологій та інструментів розробки

2.1 IDE WebStorm

WebStorm – інтегроване середовище розробки для JavaScript та пов’язаних технологій, створене для полегшення роботи із поширеними рутинними задачами, з якими стикаються програмісти під час своєї роботи.

Використання WebStorm значно полегшило розробку в аспектах роботи зі збіркою та запуском проєкту, а також із системою керування версіями Git.

2.2 MySQL Workbench

MySQL Workbench – програма, що надає візуальний інтерфейс для роботи з базами даних та проєктування нових баз даних у СКБД MySQL.

Інструментарій з візуального проєктування реляційної схеми даних, а також автоматизований процес конвертації цієї схеми у MySQL DDL значно спростив розробку бази даних для ресурсу.

2.3 Node.js

Node.js – JavaScript-оточення, створене для побудови масштабованих веб-додатків, взаємодія із клієнтом заснована на колбек-функціях, що викликаються для кожного з’єднання, основна риса архітектури Node.js – цикл подій, що надає йому переваги над деякими іншими популярними моделями проєктування веб-додатків, як от потоковою моделлю – в Node.js рідше виникають умови для дедлоків. Також позитивною рисою подібної архітектури є її схожість на баузерну архітектуру роботи із JavaScript, що полегшує розробку. Важливим аспектом роботи із Node.js є те, що це оточення прямо орієнтоване на роботу із HTTP, і має багато вбудованих способів роботи із цим протоколом, а також розвинену

екосистему навколо нього, що робить Node.js однією із кращих платформ для розробки веб-сайтів [3].

2.4 Express.js

Express.js – швидкий та мінімалістичний фреймворк з відкритим кодом для оточення Node.js призначений для роботи із HTTP маршрутизацією. Express часто вважають стандартом роботи із Node.js, а багато інших фреймворків засновані на ньому. Механізм функцій middleware, що виконуються Express перед безпосередньою обробкою запита користувача бізнес-логікою, надає зручний та потужний функціонал для роботи із HTTP запитами та дозволяє ефективно перевикористувати великий об'єм коду та легко підключати модулі із екосистеми Node.js [4].

Одним із прикладів таких модулів, що використані у цьому проєкті є express-session – middleware для Express, що дозволяє легко створювати та підтримувати сесии користувача на веб-сайті. Express-session знімає з користувача необхідність прямо працювати з Cookies та автоматично створює session-store для зберігання даних сесии користувача, при цьому надаючи можливість глибоко налаштувати як саме будуть працювати сесии, та де будуть зберігатися дані сесій користувачів.

2.5 Бібліотека node.bcrypt.js

node.bcrypt.js – бібліотека, що надає реалізацію функції хешування bcrypt для застосування із JavaScript. Bcrypt необхідна для реалізації безпечного зберігання паролів користувачів. Bcrypt інкорпорує salting для захисту від rainbow-table атак, а також підтримує збільшення часу виконання функції у разі необхідності підвищення захисту від brute-force атак, що дозволяє їй залишатися актуальною функцією хешування для захисту паролів вже довгий час [6].

2.7 EJS

EJS (embedded JavaScript) – мова шаблонізації HTML, що використовує синтаксис та засоби JavaScript для генерації HTML файлів. Використана у проєкті для генерації деяких сторінок та їх спільних елементів на стороні сервера перед відправкою клієнту, можливості даної бібліотеки дозволяють зручно генерувати інтерфейс для різних категорій клієнтів[7].

2.8 Twitter Bootstrap

Twitter Bootstrap – бібліотека з відкритим кодом, створена для полегшення розробки користувацького інтерфейсу веб-застосунків, надає набір класів CSS для стилізації HTML елементів сторінок, а також JS код для інтерактивної взаємодії із деякими елементами, що надані бібліотекою.

В проєкті застосована для стилізації елементів навігації, таблиць для відображення даних, форм для взаємодії із сервером, основних контейнерів змісту сторінок а також кнопок, посилань тощо.

Розділ 3. Реалізація веб-сайту

3.1 Вимоги до проєкту

Передбачається робота з чотирма типами користувачів:

- Гість
- Зареєстрований користувач
- Модератор
- Заблокований користувач

Функціонал, що доступний всім користувачам системи:

- Перегляд рекомендацій
- Перегляд деталей про книгу
- Перегляд рецензій на книгу, глав книги, коментарів до глав, оцінки книги іншими користувачами.
- Пошук книг за назвою, пошук книг за тегами книги.
- Вхід, вихід із системи

Додатковий функціонал, що має бути доступний гостю:

- Реєстрація.

Функціонал, доступний всім зареєстрованим користувачам:

- Редагування особистої інформації
- Видалення акаунту і пов'язаної інформації
- Створення, редагування книг
- Створення, редагування та приховування глав до них
- Коментування глав будь-яких книг
- Оцінювання книг, опублікованих на сайті
- Рецензування опублікованих книг
- Видалення будь-якого опублікованого особисто змісту на сайті
- Створення закладок на главах книг, перегляд закладок, їх видалення
- Створення читацького списку книг, його перегляд та редагування.

- Перегляд списку власних опублікованих книг
- Створення скарг на коментарі та глави, опубліковані на сайті.

Функціонал доступний модераторам:

- Перегляд скарг, видалення скарг зі списку розгляду
- Блокування користувачів
- Приховування, видалення коментарів, глав із сайту
- Перегляд будь-якого прихованого змісту
- Перегляд детальної інформації користувача

Функціонал, до якого втрачає доступ заблокований користувач:

- Публікація книг
- Публікація та редагування коментарів, рецензій, глав

3.2 Опис реалізованого функціоналу сторінок веб-сайту

На кожній сторінці присутній елемент нав, наповнений різними опціями в залежності від ролі користувача.

Зображення варіації елемента навігації для гостя:



Рис. 3.2.1 Навігація вар. 1

Зображення варіації елемента навігації для авторизованого не-модератора:



Рис. 3.2.2 Навігація вар. 2

Зображення навігації модератора:

Рис. 3.2.3 Навігація вар. 3

«Home» сторінка передбачає показ рекомендованих книжок.

Recommendations

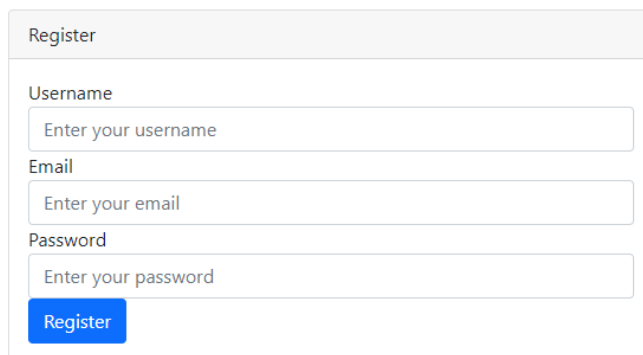
Title	Author	Rating
Development book	dev@dev.com#5	5
Really new book !	a#2	4.5
Book 3. has chapters and tags	a#2	4.5
New book	a#2	4.5
Book 1	a#2	3.5
Book 2	a#2	2.5
New book fresh out of ban!	dev@dev.com#5	0

Next

Рис. 3.2.4 Рекомендації

Таблиця відображення рекомендацій передбачає можливість переходу до наступної або ж попередньої сторінки рекомендацій. Рекомендації формуються на основі оцінок користувачами книг.

«Register» – форма для реєстрації у системі.

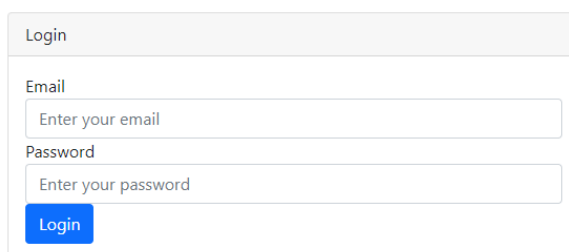


The image shows a web form titled "Register". It contains three input fields: "Username" with the placeholder text "Enter your username", "Email" with the placeholder text "Enter your email", and "Password" with the placeholder text "Enter your password". Below the password field is a blue button labeled "Register".

Рис. 3.2.5 Реєстрація

Передбачає введення імені користувача, унікальної адреси електронної пошти, паролю.

«Login» - форма для авторизації в системі.



The image shows a web form titled "Login". It contains two input fields: "Email" with the placeholder text "Enter your email" and "Password" with the placeholder text "Enter your password". Below the password field is a blue button labeled "Login".

Рис. 3.2.6 Логін

Потребує введення пошти та паролю користувача.

«Advanced search» - сторінка для пошуку книги за розширеними критеріями.

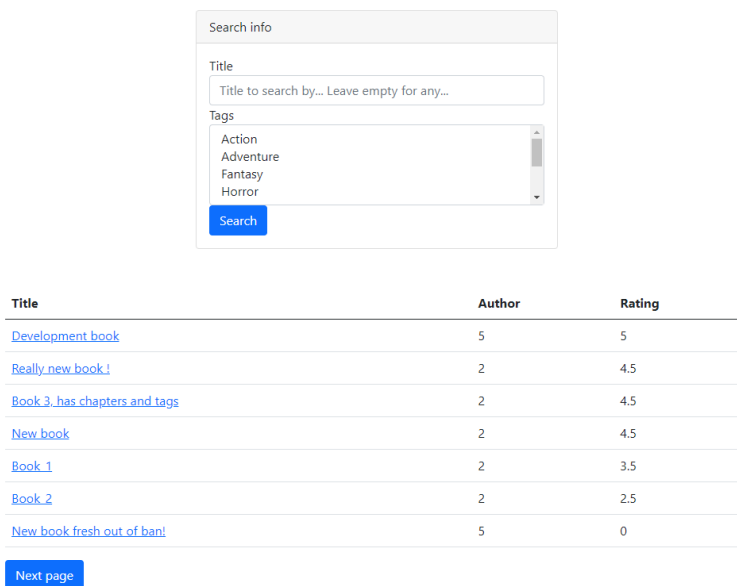


Рис. 3.2.7 Сторінка розширеного пошуку

Користувач може обирати теги, в яких він зацікавлений зі списку, теги, що є частиною запиту відобразатимуться зверху списку, для того, щоб прибрати їх із запиту, користувач має натиснути на відповідний тег, також, користувач має змогу ввести запит на назву книги. Пошук виконується за принципом знаходження книг, які відповідають хоча б всім перерахованим вимогам, тобто результати містять всі задані користувачем теги і можуть містити будь-які інші крім заданих. Пошук за назвою здійснюється за принципом знаходження заданої користувачем стрічки запиту в якості підстрічки назви. Як і на сторінці рекомендацій, передбачено перехід на наступну або ж попередню сторінку результатів пошуку.

«Publish» - сторінка публікації книги.

Publish a book!

Title

Enter the title of your book

Publish!

Рис. 3.2.8 Сторінка публікації книги

Від користувача потребується ввести назву нової книги, кнопка “Publish!” призведе до публікації нової книги з відповідною назвою.

«My Books» - сторінка відображення книг, автором яких є користувач, реалізована пагінація для перегляду даного списку, у таблиці списку містяться посилання на перехід до сторінок самих книг.

Your books

Title	Author	Rating
New book	2	4.5
Book 3, has chapters and tags	2	4.5
Really new book!	2	4.5
Book 2	2	2.5
Book 1	2	3.5

Next

Рис. 3.2.9 Сторінка перегляду книг користувача

«Reading list» - сторінка з відображенням читацького списку користувача, можливий перехід по сторінках читацького списку та до відповідної книги, можливе видалення книги із читацького списку натисканням відповідної кнопки “Remove”.

Your reading list

Title	Remove?
Development book	<input type="button" value="Remove"/>
Really new book !	<input type="button" value="Remove"/>

Рис. 3.2.10 Сторінка читацького списку

«Bookmarks» - сторінка закладок користувача у главах книг, надає такі можливості перегляду і керування списком як і сторінка читацького списку.

Your bookmarks

Title	Remove?
Updated Chap	<input type="button" value="Remove"/>
sdfsf	<input type="button" value="Remove"/>

Рис. 3.2.11 Сторінка списку закладок

«View reports» - сторінка, доступна модератору, що існує для перегляду і керування списками скарг на коментарі та глави книг. Модератор має змогу переглядати всі скарги, переходячи між сторінками відображення, а також видаляти скарги за допомогою відповідної кнопки. Для того, щоб перейти на перегляд вмісту, на який, власне, поскаржились, у представлення скарги вставлене посилання на цей вміст (сторінку перегляду глави, або ж редагування коментаря відповідно).

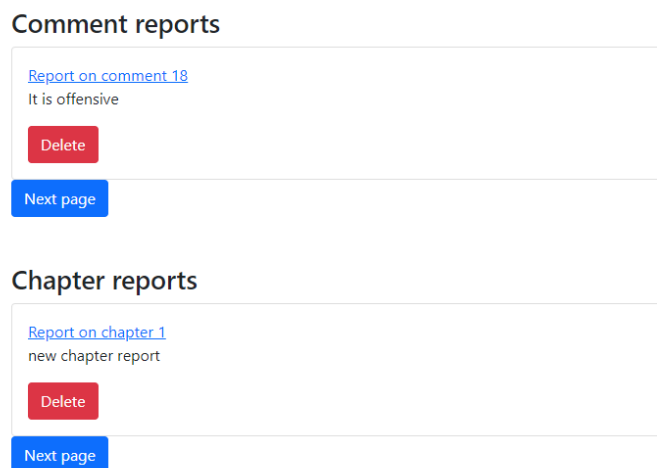


Рис. 3.2.12 Сторінка перегляду скарг

«Report page» - сторінка для надання користувачами скарг на деякий вміст сайту. Ідентична для скарг на коментарі та на глави книг, у формі на сторінці користувач має можливість описати суть своєї скарги, та, власне, надіслати скаргу модератору ресурсу.

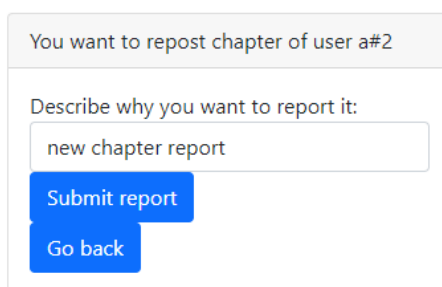


Рис. 3.2.13 Сторінка оформлення скарги

«Book details» - сторінка перегляду і взаємодії із книгою, на ній можна побачити назву, автора, перелік тегів, перелік глав у хронологічному порядку публікації (а саме їх частини, реалізований перехід по сторінкам

відображення цього списку), рецензії. Незаблокований користувач також має змогу опублікувати рецензію, якщо раніше на публікував її на цю книгу, перейти до редагування своєї рецензії в іншому випадку; оцінити книгу або ж змінити свою оцінку даної книги; додати її до свого списку читання. В разі якщо користувач є автором цієї книги, присутня кнопка переходу до її редагування. Автори та модератори також бачать всі опубліковані до книги глави, навіть якщо вони приховані, на відміну від звичайних користувачів. Приховані глави позначаються червоним посиланням, а рецензії збільшеною прозорістю. В разі якщо прихована рецензія, її бачить автор рецензії та модератори.

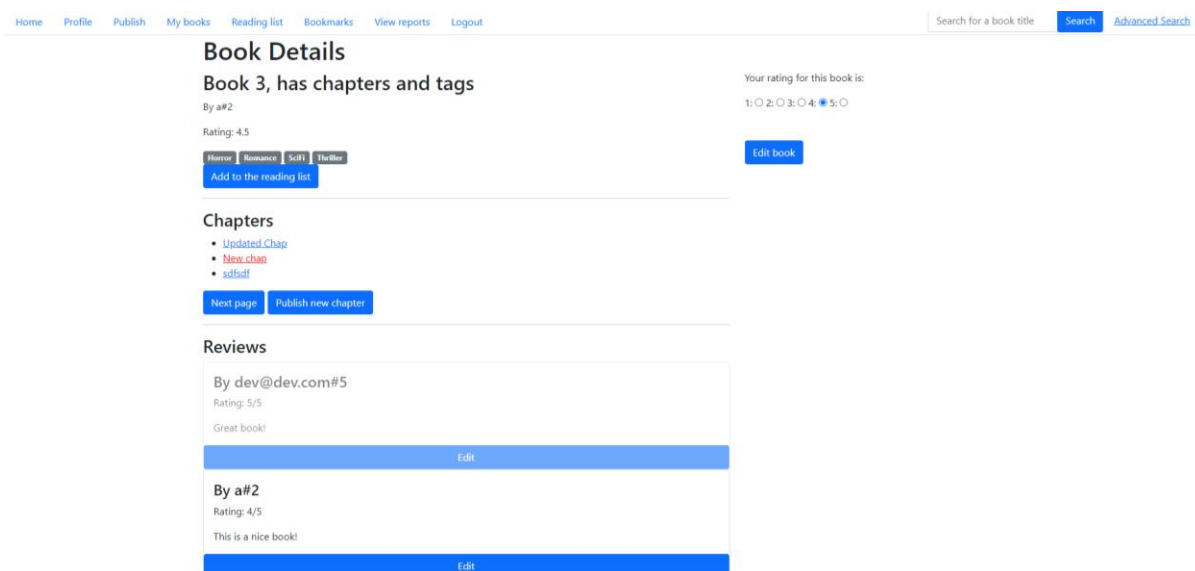


Рис. 3.2.14 Сторінка книги (з точки зору автора)

«Edit book» - сторінка редагування книги, яка доступна тільки автору. На ній автор має форму, в якій може змінити назву книги, додати нові теги, або ж видалити дану книгу.

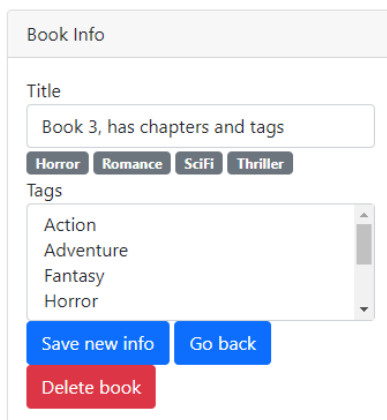


Рис. 3.2.15 Сторінка редагування книги

«Chapter publish» - сторінка створення нової глави для книги, доступна незаблокованим користувачам, до сторінки під'єднаний rich text editor Quill.js, за допомогою якого користувач має змогу створювати текст із HTML розміткою для кращого відображення кінцевого продукту. Після введення вмісту глави та її назви, користувач може опублікувати главу натиснувши відповідну кнопку.

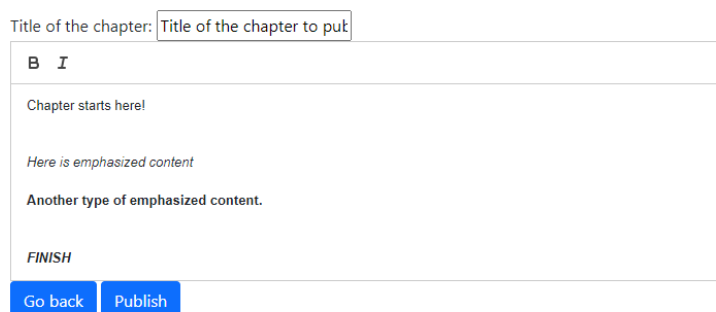


Рис. 3.2.16 Сторінка створення та публікування глави

«Chapter edit» - сторінка редагування глави, ідентична публікації, з єдиною різницею у тому, що редактор вже наповнений існуючим до редагування текстом. Також присутня кнопка для приховування/відкриття доступу для загалу до глави і кнопка для видалення глави. Ця сторінка доступна для автора глави та модератора, з різницею у тому, що модератор має змогу лише видалити або ж приховати главу, але не змінювати її вміст.

«Chapter view» - сторінка перегляду конкретної глави певної книги, доступна кнопка переходу від глави до книги, переходу до наступної/попередньої глави книги, форма для публікації коментарів, кнопка для встановлення закладки на цій главі. Внизу сторінки відображаються коментарі до глави, для їх перегляду реалізовано пагінацію, приховані коментарі видні для модераторів та авторів коментарів і виділені більшою прозорістю. Для кожного коментаря присутня кнопка, що ініціює процес складання скарги на коментар. Така ж кнопка («Report») присутня і для створення скарги на главу, після натискання даних кнопок, користувача спрямує на відповідну сторінку. Для автора і модератора існує кнопка переходу на сторінку редагування глави, а модератор має доступ до переходу на сторінку детальної інформації про автора глави.

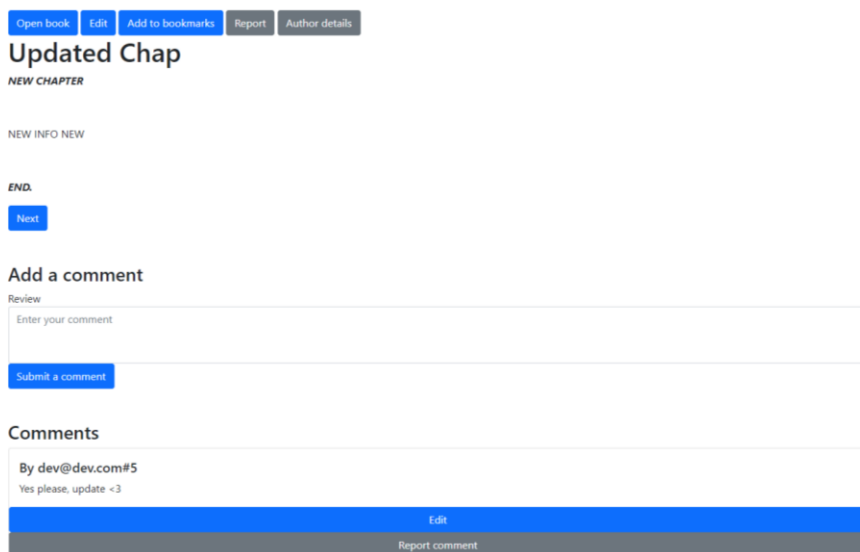


Рис. 3.2.17 Сторінка глави книги (з точки зору модератора)

«Comment edit» - сторінка редагування коментаря, доступна модератору та автору коментаря, з різницею у тому, що модератор не може редагувати вмісту коментаря, а лише здатен приховувати або ж видаляти його. Для модератора доступний перехід до перегляду детальної інформації автора коментаря (посилання “details”).

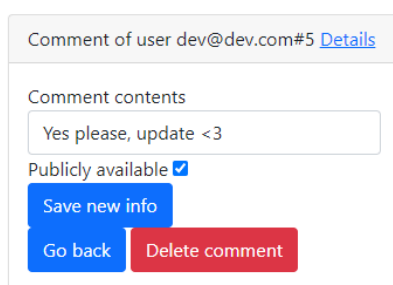
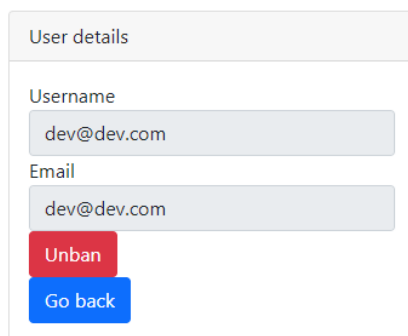


Рис. 3.2.18 Сторінка редагування коментаря (з точки зору модератора)

«User details» - сторінка, доступна модераторам, тут відображається детальна інформація про користувача, а також є функціонал блокування

або ж розблокування цього користувача, яким може скористатися модератор, якщо вважає за потрібне.



User details

Username
dev@dev.com

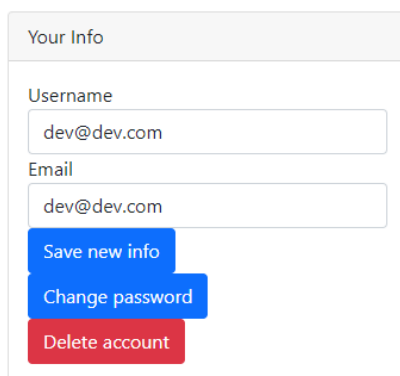
Email
dev@dev.com

Unban

Go back

Рис. 3.2.19 Сторінка деталей про користувача

«Profile» - сторінка керування профілем для користувача. Тут користувач може встановити нове ім'я, адресу електронної пошти, змінити пароль, видалити акаунт.



Your Info

Username
dev@dev.com

Email
dev@dev.com

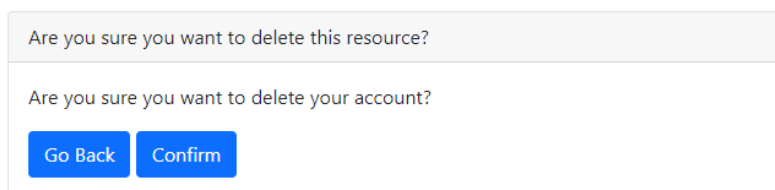
Save new info

Change password

Delete account

Рис. 3.2.20 Сторінка профілю користувача

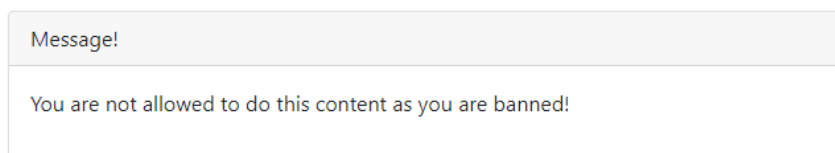
«Сторінка підтвердження наміру видалити ресурс» - в разі бажання видалити акаунт, книгу чи главу до неї ініціатор процесу видалення буде спрямований на сторінку підтвердження наміру видалити обраний ресурс.



The image shows a confirmation dialog box with a light gray header and a white body. The header contains the text "Are you sure you want to delete this resource?". The body contains the text "Are you sure you want to delete your account?" and two blue buttons: "Go Back" and "Confirm".

Рис. 3.2.21 Сторінка підтвердження видалення ресурсу

«Сторінка відображення повідомлення сервера» - у деяких випадках, в разі коли користувач не має відповідного рівня авторизації, або ж у випадку внутрішньої помилки сервера, користувач може бути спрямований на сторінку яка відображатиме повідомлення сервера. Наприклад, що він заблокований і не має права виконувати дію, що намагався.



The image shows a message notification box with a light gray header and a white body. The header contains the text "Message!". The body contains the text "You are not allowed to do this content as you are banned!".

Рис. 3.2.22 Сторінка відображення повідомлення сервера

Якщо користувач заблокований, або ж виконує яку-небудь заборонену і звичайним користувачам дію (наприклад, при створенні рецензії користувач залишив поле тексту рецензії пустим), на сторінці може з'явитися повідомлення в подібному форматі, із описом причини його появи:

× Review must be non-empty

Рис. 3.2.23 Елемент відображення повідомлення сервера

3.3 Опис структури даних проекту

Для збереження даних платформи використано СКБД MySQL.

Реляційна модель даних має наступний вигляд:

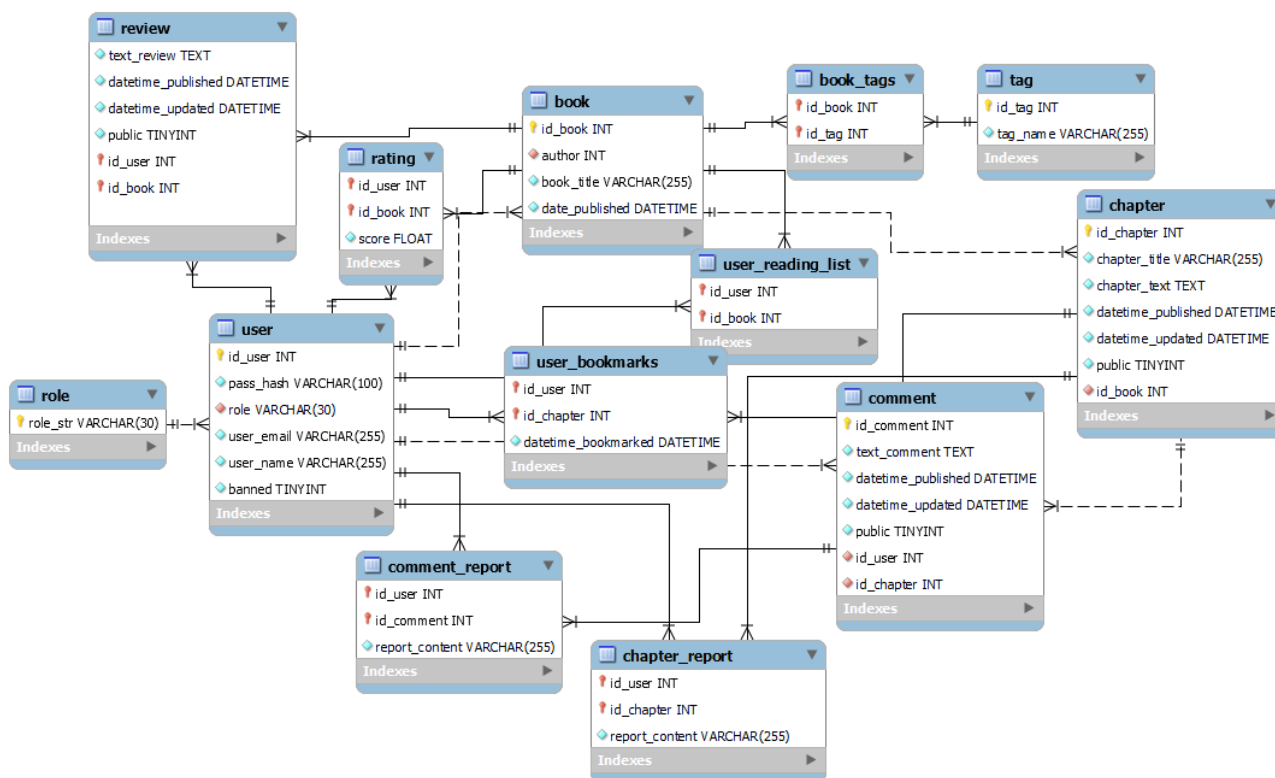


Рис. 3.3.1 Реляційна модель даних

Типи аутентифікованих користувачів системи реалізовано через додаткову реляцію `role`, посилення на кортежі котрої через `foreign key constraint` містить реляція `user`, що дозволяє визначати певний набір ролей користувачів платформи, така модель визначення ролей користувачів у системі гнучка, адже дозволяє у майбутньому визначити додаткові ролі для можливого розширення системи, але також дозволяє забезпечувати певну цілісність даних системи через обмеження на множину значень, які може приймати атрибут `role` реляції `user`. Чи заблокований користувач визначається атрибутом `banned` у `user`.

Книги, глави книг, коментарі представлені відповідними реляціями у моделі.

Список читання, список закладок користувача визначені через комбінований первинний ключ, і, відповідно, допускають лише одне входження відповідних сутностей до списків конкретного користувача. Так само, користувач може скласти лише одну рецензію певної книги і лише одну скаргу на певну главу чи коментар.

Допускається існування лише одного тегу з відповідною назвою, тож для забезпечення цієї умови а також для пришвидшення пошуку тегів за назвою, було створено `unique index` на `tag_name` в реляції `tag`, таке саме обмеження створено для атрибуту `user_email` реляції `user`.

Зв'язок тегів та книжок реалізовано через таблицю `book_tags`, пошук книжок за тегами при відповідному запиті користувача відбувається через наступний параметризований SQL запит з використанням механізму JOIN:

```

SELECT b.id_book, b.author, b.book_title, b.date_published, AVG(COALESCE(r.score, 0)) AS avg_score
FROM book b
LEFT JOIN rating r
ON b.id_book = r.id_book
LEFT OUTER JOIN book_tags ON b.id_book = book_tags.id_book
LEFT OUTER JOIN tag t ON book_tags.id_tag = t.id_tag
WHERE b.book_title LIKE ?
AND t.tag_name IN (?)
GROUP BY b.id_book, b.author, b.book_title, b.date_published
HAVING COUNT(DISTINCT t.tag_name) >= ?
ORDER BY avg_score DESC, b.date_published
LIMIT ?
OFFSET ?;`

```

Рис. 3.3.2 Запит на отримання книг за умови розширеного пошуку

Де знаходимо повну інформацію по книгам, враховуючи їх середню оцінку (яку за замовчуванням вважаємо рівною нулю, що прописуємо у функції COALESCE), і фільтруємо у результат лише ті книги, що мають зв'язок із тегами з певними визначеними користувачем назвами, і рівну або більшу кількість цих тегів у відповідній групі, що формується для цієї книги у GROUP BY після фільтрації. Таким чином досягаємо бажаного результату, знаходячи книги, що мають всі ті теги, що визначені користувачем.

3.4 Розгляд програмних рішень в проєкті

3.4.1 Архітектура

Для реалізації проєкту обрана архітектура MVC.

В ролі Model виступають модулі сервісів роботи з даними та пов'язані з ними класи DAO, що працюють безпосередньо з базою даних.

В якості View виступають ejs шаблони та пов'язані з ними клієнтські сценарії JavaScript.

Роль Controller виконують express Routers, що розділені по файлах відносно того, яку частину функціоналу вони контролюють, а також набір express middleware функцій, що використовуються для авторизації користувачів (аутентифікація виконується за вказаними у відповідному контролері шляхами).

3.4.2 Авторизація

Типові Express middleware функції отримують об'єкт запиту, об'єкт відповіді та наступу функцію обробки запиту. В Express існує два типи middleware не пов'язаних з обробкою помилок – middleware на рівні застосунку та middleware на рівні маршрутизатора [5]. Для виконання авторизації застосовуємо middleware на рівні маршрутизатора. В разі, коли користувач не авторизований належним чином для виконання певної дії, у певній middleware функції можна сформувавши належну відповідь і не робити виклик наступної функції обробки запиту. Наприклад, розглянемо функцію перевірки чи заблокований користувач:

```
requireNotBannedJsonResponse: (req, res, next) => {  
  if (!req.session.user) {  
    return res.status(403)  
      .send(JSON.stringify( value: {msg: "Only logged in users may view this content!"}));  
  } else if (req.session.user.banned) {  
    return res.status(403)  
      .send(JSON.stringify( value: {msg: 'You are not allowed to do this content as you are banned!'}));  
  }  
  next();  
},
```

Рис. 3.4.2.1 Функція перевірки, чи заблокований користувач

Перше, що відбувається при виклику даної функції – перевірка, що за допомогою express-session middleware був сформований об'єкт з даними сеансу користувача під назвою user, в разі, коли він був не сформований, функція надсилає відповідь для клієнта із належним повідомленням про недостатній рівень прав і необхідність авторизації. В іншому випадку, перевіряється поле banned об'єкта user, яке відповідає однойменному атрибуту відповідної реляції і є індикатором того, чи заблокований користувач. В разі, якщо користувач заблокований, належна відповідь формується і робота функції завершується, В іншому випадку викликаємо наступну функцію обробки запиту.

Інші функції авторизації користувача мають подібну структуру та логіку.

Розглянемо приклад застосування цих функцій у маршрутизаторах:

```

Vsevolod Kalinbet
app.delete( path: '/comment/:id', auth.requireNotBannedJsonResponse,
  auth.requireCommentAuthorshipOrModerationJsonResponse, async (req :..., res :...) => {
    await commentService.deleteComment(req, res);
  });

```

Рис. 3.4.2.2 Застосування middleware до маршрутизаторів

На даному рисунку бачимо застосування middleware функцій авторизації до шляху маршрутизатора роботи з коментарями. За допомогою них, обмежуємо доступ до видалення коментаря до незаблокованого автора та медераторів.

3.4.3 Рівень відображення

Для побудови користувацького інтерфейсу використано поєднання шаблонізатора EJS з клієнтськими сценаріями JavaScript для надання інтерактивності сторінкам веб-сайту. Сторінки стилізовано через використання Bootstrap.

Певні спільні елементи веб-сторінок винесені в окремі файли для перевикористання за допомогою механізму include з EJS[7], наприклад елемент повідомлення:

```

<div id="message" class="alert alert-primary" role="alert">
  
  <span id="msg_msg"></span>
</div>

```

Рис. 3.4.3.1 Елемент повідомлення, HTML

Та клієнтський сценарій до нього:

```

<script>
  const message = document.getElementById("message");
  const messageSpan = document.getElementById("msg_msg");
  message.addEventListener('click', () => {
    message.style.display = 'none';
  });
  message.style.display = 'none';
</script>

```

Рис. 3.4.3.2 Елемент повідомлення, JS

Для генерації інтерактивних частин сторінки використані звичайні можливості JavaScript без використання клієнтських фреймворків. Наприклад, відображення отриманих через fetch рецензій для певної книги відбувається у наступній функції, що є частиною клієнтського сценарію сторінки книги:

```

function drawReviews(res) {
  document.getElementById( elementId: "reviewAnchor").innerHTML = "";
  console.log(res);
  res.forEach(function (review) {
    const divCard = document.createElement( tagName: 'div');
    divCard.classList.add('card');
    const divCardBody = document.createElement( tagName: 'div');
    divCardBody.classList.add('card-body');
    const h4 = document.createElement( tagName: 'h4');
    h4.classList.add("card-title");
    const p1 = document.createElement( tagName: 'p');
    p1.classList.add("card-text");
    const p2 = document.createElement( tagName: 'p');
    p2.classList.add("card-text");
    if (!review.public) {
      divCard.style.opacity = "0.6";
    }
    h4.textContent = `By ${review.author}`;
    p1.textContent = `Rating: ${review.score === 'null' ? 'not given' : review.score}/5`;
    p2.textContent = review.text_review;
    divCardBody.appendChild(h4);
    divCardBody.appendChild(p1);
    divCardBody.appendChild(p2);
    divCard.appendChild(divCardBody);
    if (review.editable) {
      const editButton = document.createElement( tagName: 'a');
      divCard.classList.add('card');
      editButton.classList.add('btn');
      editButton.classList.add('btn-primary');
      editButton.textContent = "Edit";
      editButton.href = `/book/${review.id_book}/reviews/${review.id_user}`;
      divCard.appendChild(editButton);
    }
    document.getElementById( elementId: "reviewAnchor").appendChild(divCard);
  });
}

```

Рис. 3.4.3.3 Функція відображення списку рецензій

Як бачимо на рисунку, створення карток відображення рецензій відбувається через вбудовані в JavaScript функції маніпуляції DOM.

Висновки

В результаті виконання курсової роботи проаналізовано існуючі веб-сайти для самвидаву художньої літератури, оцінено актуальність подібних платформ у сучасному світі, виокремлено та узагальнено основні спільні риси даних платформ, та те, у чому вони різняться. Під час виконання практичної частини курсової роботи успішно реалізовано у власному програмному продукті основні з цих загальних для подібних платформ рис, у процесі чого були отримані нові знання та навички роботи із оточенням JavaScript Node.js та супутньою екосистемою програмних продуктів, удосконалені навички роботи із СКБД MySQL та SQL загалом. Фінальний програмний продукт може бути розвинений далі, перспективним напрямком розробки є покращення графічного інтерфейсу користувача, поліпшення примітивної системи рекомендацій книг, поширення функціоналу пошуку книг та додавання роботи з мультимедіа як-от обгортки книг та вбудовані зображення у глави.

Список використаних джерел

1. Stack Overflow Developer Survey 2022. [Електронний ресурс]. – Режим доступу: <https://survey.stackoverflow.co/2022/>
2. Google Trends. [Електронний ресурс]. – Режим доступу: <https://trends.google.co.in/home>
3. About | Node.js. [Електронний ресурс]. – Режим доступу: <https://nodejs.org/en/about>
4. Express – Node.js web application framework. [Електронний ресурс]. – Режим доступу: <https://expressjs.com/>
5. Using Express middleware. [Електронний ресурс]. – Режим доступу: <https://expressjs.com/en/guide/using-middleware.html>
6. Contributors to Wikimedia projects. bcrypt. [Електронний ресурс]. – Режим доступу: <https://en.wikipedia.org/wiki/Bcrypt>
7. EJS -- Embedded JavaScript templates. [Електронний ресурс]. – Режим доступу: <https://ejs.co/>