

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему: «Інформаційна система підтримки буккросингу»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студента групи ІТ.м-12 Любицький Єгор Іванович

Кваліфікаційну роботу
захищено на засіданні ЕК
з оцінкою _____

«___» грудня 2022 р.

Науковий керівник _____

(підпис)

к.т.н., доц., Федотова Н.А.

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент _____
(підпис)

Суми-2022

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

В.о. зав. кафедри ІТ

_____ С. М. Ващенко
«__» _____ 2022 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентіві

Любицький Єгор Іванович
(прізвище, ім'я, по батькові)

1 Тема проекту Інформаційна система підтримки буккросингу

затверджена наказом по університету від «04» листопада 2022 р. № 1015-VI

2 Термін здачі студентом закінченого проекту «10» грудня 2022 р.

3 Вхідні дані до проекту _____ технічне завдання _____

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їй належить розробити) АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ, ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ, МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ, ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

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

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Аналіз предметної області	Федотова Н. А.		
Постановка задачі та вибір методів розробки	Федотова Н. А.		
Практична реалізація інформаційної системи	Федотова Н. А.		

Дата видачі завдання _____.

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів випускної проекту	Термін виконання етапів проекту	Примітка
1	Дослідження предметної області	01.09.22-05.09.22	
2	Підготовка матеріалу	05.09.22-07.09.22	
3	Оцінка вимог та ризиків проекту	07.09.22-10.09.22	
4	Розробка календарного плану	10.09.22-12.09.22	
5	Розробка прототипу	12.09.22-17.09.22	
6	Підготовка та верстка сайту	17.09.22-05.10.22	
7	Розробка програмної частини	05.10.22-20.10.22	
8	Наповнення контентом	20.10.22-22.10.22	
9	Тестування	22.10.22-27.10.22	
10	Підготовка та перенесення на сервер	27.10.22-29.10.22	

Магістрант _____

Любицький Є.І.

Керівник роботи _____

к.т.н., доц. Федотова Н.А.

РЕФЕРАТ

Тема кваліфікаційної роботи магістра «Інформаційна система підтримки буккросингу».

Практична значення лежить у створенні інформаційної системи по ініціації обміну книгами серед користувачів та їх обговорення. Користувачі можуть не лише створювати обмін, а й вести активно ділитись думками на форумі.

В першому розділі – проаналізовано предметну область інформаційної системи, проведено розбір останніх досліджень, та публікацій, аналіз останніх продуктів.

В другому розділі – поставлено задачу та обрано програмне забезпечення.

В третьому розділі – змодельовано структуру інформаційної системи, структурно-функціональне моделювання процесу, моделювання діаграми варіантів використання та проектування бази даних.

В четвертому розділі – продемонстровано практичну реалізацію інформаційної системи з боку користувача та адміністратора.

Результатом проведеної роботи є створена інформаційна система підтримки буккросингу.

Кваліфікаційна робота містить 103 сторінок, 8 таблиць, 69 рисунків, список літератури 30 найменувань, 2 додатки.

Ключові слова: ІНФОРМАЦІЙНА СИСТЕМА, БУККРОСИНГ, ПРОЕКТ, ФОРУМ, КОРИСТУВАЧ, МОДЕЛЮВАННЯ, ПЛАНУВАННЯ, ВІЗУАЛІЗАЦІЯ, ВІДГУК, АДМІНІСТРАТОР, ОБГОВОРЕННЯ, КОМУНІКАЦІЯ.

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Огляд останніх досліджень і публікацій.....	8
1.2 Аналіз програмних продуктів.....	10
2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ.....	22
2.1. Мета та задачі дослідження.....	22
2.2. Розробка візуальної частини інформаційної системи.....	23
2.3. Розробка функціональної частини інформаційної системи.....	24
3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ІС.....	26
3.1 Структура інформаційної системи.....	26
3.2 Структурно-функціональне моделювання процесу.....	27
3.3 Моделювання діаграми варіантів використання.....	28
3.4. Проектування бази даних.....	31
4 ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	38
4.1 Програмна реалізація.....	38
4.2 Використання інформаційної системи зі сторони користувача.....	39
4.3 Адміністрування інформаційної системи.....	54
ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60
ДОДАТОК А.....	63
ДОДАТОК Б. КОД РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	75

ВСТУП

Деградація навколишнього середовища була гарячою темою з минулого століття. У перші два десятиліття двадцятого століття відбулося велике порушення та надмірне використання природних ресурсів, що призвело до виснаження навколишнього середовища на землі з подальшим вирубуванням лісів для забезпечення зростання промисловості та населення.

Виробництво будь-якої книги чи друкованого продукту для видавництва чи маркетингу включає численні етапи процесу та види матеріалів. Як видавцям, покупцям друкованих видань або виробникам книг, нам потрібно розглядати досить детально, щоб зробити значущий вибір і постійно вдосконалювати спосіб виробництва готових виробів.

Щороку друкується приголомшлива кількість книжок: через традиційних видавців і дистриб'юторів ми розглядаємо цифру, яка, за припущеннями експертів, становить від 500 000 до 1 мільйона найменувань, що виходять із друку кожні дванадцять місяців. І це теж галузеві видавці – якщо ми розширимо наше визначення, включивши також самовидавничі назви, цифра фактично сягає близько 4 мільйонів назв на рік, величезна цифра, навіть якщо взяти до уваги значне зростання продажів книжок, яке відбулося через Covid-19 пандемія викликана ще в перший рік соціальних обмежень і локдаунів. Загалом ми друкуємо більше книжок, ніж будь-коли раніше, і в міру зростання видавничих секторів ми, ймовірно, побачимо, що це посилюватиметься.

Використання паперу та обробка вуглецю є основними речами, про які слід турбуватися, коли мова заходить про загальні недоліки паперових книг перед електронними книгами, але в галузі докладаються зусилля, щоб бути екологічнішими та використовувати перероблений папір, сертифікований FSC. зменшити вимоги до вирубки дерев і переконатися, що дерева не виснажуються до непоправних рівнів. Більш екологічні методи виробництва з меншим вмістом вуглецю на даний момент є гарячою темою в багатьох галузях виробництва,

включаючи паперові та електронні книги, а також пристрої для читання електронних книг, і багато авторів вимагають від своїх видавців більш екологічних методів друку.

Отже, було прийнято рішення розробити інформаційну систему підтримки буккросингу для популяризації екологічного використання книг в Україні.

Тема: «Інформаційна система підтримки буккросингу».

Мета: Розробка інформаційної системи підтримки буккросингу із реалізацією усіх можливостей для взаємодії користувачів між собою. Користувачі інформаційної системи повинні мати можливість провести обмін книгами через створену систему. Обмін повинен виконуватися зі згодою обох користувачів.

Об'єкт дослідження: процес взаємодії користувачів системи для обміну книгами.

Предмет дослідження: функціональні та методологічні розв'язання шляхів поставлених задач інформаційної системи підтримки буккросингу.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень і публікацій

У США, Японії та Європі в середньому людина витрачає від 200 до 250 кілограмів паперу щороку. В Індії цей показник становить 5 кг, а в деяких країнах менше 1 кг. Для виробництва 1 кілограма паперу потрібно в 2-3 рази більше його ваги в деревах [1]. Якби кожен використовував 200 кілограмів паперу на рік, то не залишилося б жодного дерева.

Виробництво паперу вимагає багато води. Щоб виготовити один аркуш паперу формату А4, залежно від заводу, потрібно від 2 до 13 літрів води [2]. Целюлозно-паперова промисловість є найбільшим промисловим споживачем води в західних країнах.

Кожна тонна переробленого паперу може уникнути використання 17 дерев; 1440 літрів олії; 2,3 куба м площі полігону; 4000 кіловатів енергії та 26500 літрів води. Папір досить просто переробити, але 55 відсотків світових запасів паперу надходить із щойно зрізаних дерев [3-4].

Рідко люди навіть усвідомлюють, що «звичайний» папір містить лише 3% целюлозних волокон (тобто деревної целюлози), решта матеріалів — це речовини, які використовуються для палітурки, фарбування, покриття тощо.

Для виробництва тонни офсетного паперу потрібно приблизно 10 м³ води. Целюлозна суміш 0,2-1,5% волокна переробляється в папероробній машині; решта просто вода. Імовірно, для виготовлення тонни «звичайного» паперу для копіювальних апаратів або повсякденного використання потрібно близько 15 м³ води, у порівнянні з вищезазначеними 3-4 м³, продемонстрованими Манкеном.

У специфікації будь-якої книги є багато інших деталей, деякі маленькі, деякі великі. Якщо ми виокремимо їх, то зможемо визначити варіанти, які мають менший вплив на навколишнє середовище. Контролювати вплив на навколишнє середовище протягом усього життєвого циклу продукту важко [6-8]. Це особливо актуально, якщо ви працюєте в невеликому масштабі як фрилансер, приватний

підприємець або невелика компанія. Оскільки все це є платформою для висвітлення соціальних, політичних, культурних, глобальних проблем разом із голосами звичайних людей, щоб привернути увагу керівного органу до відповідних питань, засоби масової інформації також відіграють роль агента поінформованості.

Термін «буккросинг» використовується для позначення практики кидання книги в громадському місці з наміром, щоб хтось інший забрав її та прочитав. Його надихнув веб-сайт bookcrossing.com, який реєструє користувачів, які бажають взяти участь у буккросингу, і відстежує понад три мільйони книг у всьому світі, коли вони переходять із рук в руки через біржі буккросингу. Буккросинг також може означати обмін книгами з друзями або дарування книг.

Концепція BookCrossing поширилася в популярну культуру, і сайт має понад півмільйона учасників. Буккросери проводять конференції, торгують книгами поштою та підтримують членів спільноти, які цього потребують. Вони також можуть брати участь у регіональних зустрічах, щоб поговорити про книги та просто зустрітися один з одним [9]. Хоча деякі видавці критикували цю практику, стверджуючи, що вона шкодить продажам, буккросери заперечують, що вільний обмін книгами насправді заохочує людей брати інші книги того ж автора або на ту ж тему, таким чином збільшуючи продажі, оскільки буккросинг викликав чийсь інтерес.

Незалежно від того, яка з причин є важливою для вас, найважливішим висновком із цих даних є те, що паперові книги виглядають так, ніби вони тут, щоб залишитися, і багато людей, схоже, віддають перевагу їм над їхніми цифровими аналогами [10]. І будь-яке читання добре – читання будь-яким способом, який вам підходить, є важливою частиною розвитку звички читати для задоволення, а також можливості читати зручно та інформативно під час навчання та на роботі [11-15]. Але наша перевага паперу свідчить про те, що книги протягом століть відіграють важливу роль у тому, як ми будуємо свою ідентичність, знаходимо нову інформацію, формуємо соціальні зв'язки та ділимося своїми знаннями, і наша прихильність до їх фізичної форми, можливо, показує, наскільки важливі книги

призначені для нашого самопочуття, а також для всього того, що знання та історії в книгах можуть нам допомогти.

1.2 Аналіз програмних продуктів

Розвиток буккросингу в Україні йде шляхом розширення як числа людей, залучених у процес книгообігу, так й числа місць, де створюються місця обміну книгами. Це може бути як кафе чи клуби, так й бібліотеки. Іноді місця для книгообігу створюються і в приємальних пунктах вторинної сировини, де книги замість здачі на переробку знайдуть своїх нових власників.

Не дивлячись на це, інформаційних систем для буккросингу досить мало та вони не розповсюдженні серед користувачів інтернету. Для аналізу продуктів аналогів було обрано:

- «Bookcrossing-ua.at.ua» [16];
- «Bookcrossing.com» [17];
- Facebook група «Bookcrossing Lviv» [18].

1.2.1 Сайт «Bookcrossing-ua.at.ua»

Першим прикладом є веб-сайт «Bookcrossing-ua.at.ua». Даний сайт був розроблений у 2011 році та з того часу так й не оновлювався в дизайні та функціоналі, що є досить значним недоліком.

На рисунку 1.1 представлена головна сторінка «Bookcrossing-ua.at.ua», на якій є новини, посилання на соціальні мережі, опитування про сайт та статистика.

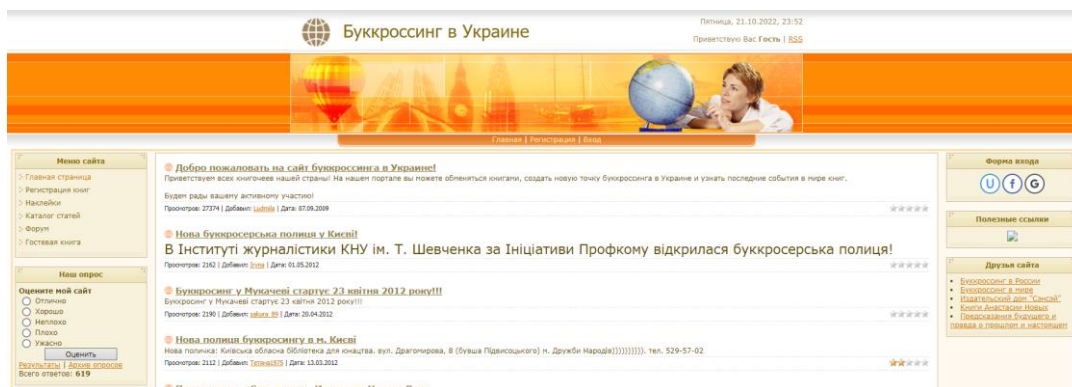


Рисунок 1.1 – «Bookcrossing-ua.at.ua» головна сторінка

Перейдемо на будь-яку новину для перегляду прикладу (рис. 1.2). На даній сторінці можна побачити, що відсутня можливість написання коментаря.

Також користувачу представлені незрозумілі дані у вигляді рейтингу. Рейтинг не змінюється від обраної новини. Крім того, додано кількість переглядів новини та хто автор новини.

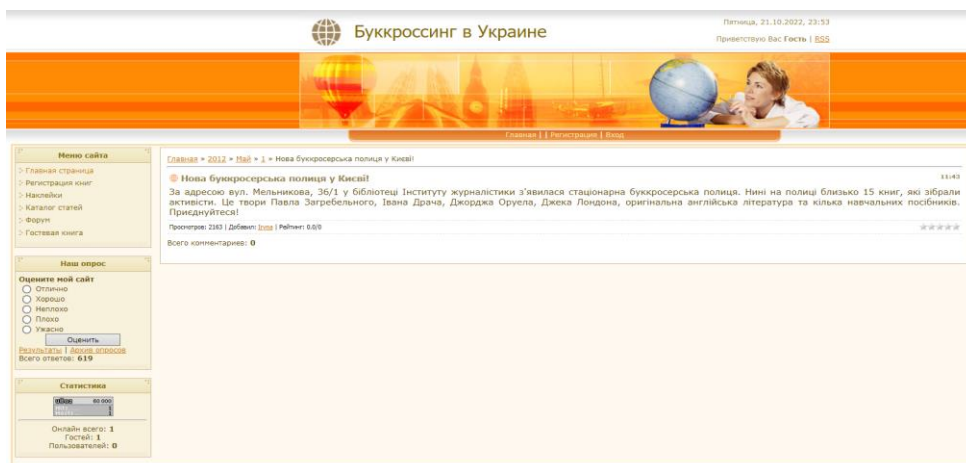


Рисунок 1.2 – «Bookcrossing-ua.at.ua» сторінка із новинами

Перейшовши на сторінку для реєстрації книг користувач отримує на сторінці два посилання – для реєстрації книги та перегляду журналу з книгами (рис. 1.2). Рішення також є досить не практичним, оскільки при переході на сторінку «Реєстрація книг», то користувач очікує саме це, а не додаткові функції для перегляду.

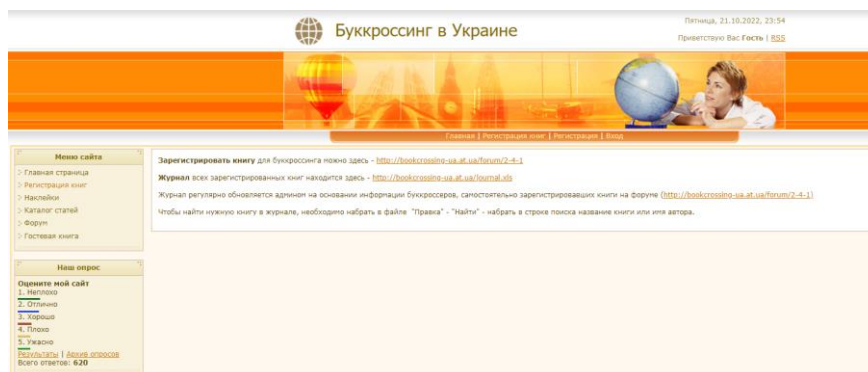


Рисунок 1.3 – «Bookcrossing-ua.at.ua» сторінка із інформацією про реєстрацію

Перейдемо на сторінку для реєстрації книг (рис. 1.4). На даній сторінці немає форми для заповнення форми й реєстрації книги. Для додавання книги потрібно написати дані в коментарі.

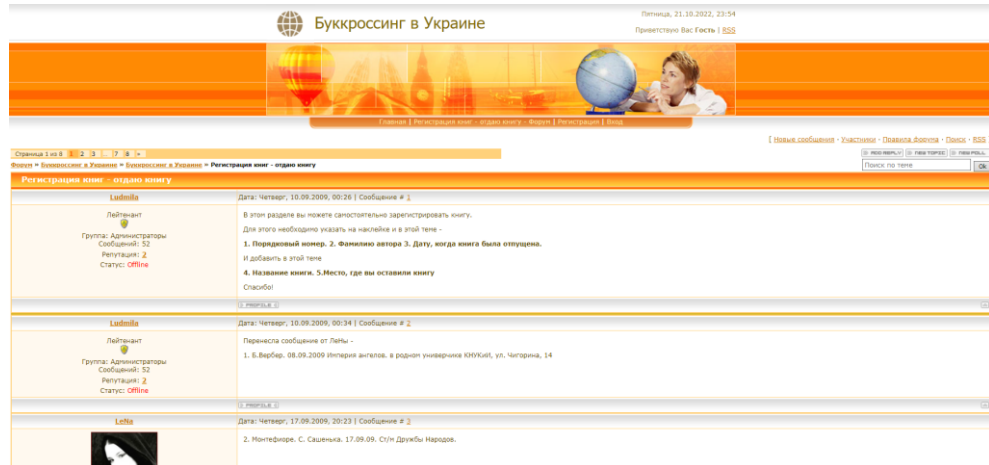


Рисунок 1.4 – «Bookcrossing-ua.at.ua» сторінка для реєстрації книги

Переходячи на посилання для перегляду книг, автоматично завантажується файл із даними (рис. 1.5). Дане рішення є не зручним, адже дані й файли статичні та повинні оновлюватися мануально, що є зручним у використанні для інших користувачів. Так не зрозуміло, що потрібно зробити для оновлення даних про книгу, де саме вносити дані.

Номер	Автор	Дата освобождия	Название	Место освобождения	Владелец	Буккроссер1	Дата	Место	Буккроссер2	Дата	Мест
1	Вербер	9/8/2009	Империя ангелов	в родном универчике	КН LeNa						
2	Монтефиоре	9/17/2009	Сашенька	Ст/м Дружбы Народов	LeNa						
3	Девид Дж. Шварц	9/19/2009	искусство мыслить масштабе	киев метро	vikont						
4	Эл Секунда	9/19/2009	принцип 15-ти секунд	киев метро	vikont						
5	Александр Свияш	9/22/2009	как быть, когда все не так,	киев метро	vikont						
6	Брайан Трейси	9/22/2009	Как нанять и удержать хорс	киев метро	vikont						
7	Рихард Дюбель	10/9/2009	Кодекс Люцифера	книгарня С (Лисенка 3)	Aston						
8	Анатолий Рыбаков	10/13/2009	Тяжелый песок	киев метро	LeNa						
9	Юрий Покальчук	11/3/2009	Анатомия греха	киев 454 маршрутка	rayel						
10	Анджей Сапковский	2/6/2010	"Башня Штуов" (золотая се	відпущу книгу на платф	Світланка						
11	Хитоми Канехара	3/7/2010	Змеи и Серги	оставлю книгу в Метро	SodeNoShirayuki						
12	Александра Маринина	4/5/2010	"Убийца поневоле"	универ	Поплавского, ска	Acscinya					
13	Луис Ривера	4/13/2010	Есть только те, кто сражае	г. Славутич, скамейка на	Satako						
14	Майн Рид	4/13/2010	Всадник без головы	Киевская обл, Бориспол	Ананаска						
15	Шарлотта Бронте	4/18/2010	Джейн Эйр	Сучасна книгарня «С»	Nalia						

Рисунок 1.5 – «Bookcrossing-ua.at.ua» файл з інформацією про книги

Отже, проаналізувавши сайт «Bookcrossing-ua.at.ua» можна зробити висновок, що якість виконання – середня. Сайт має багато недоліків та не повний функціонал. Головним недоліком є застарілий дизайн та відсутність актуальної інформації.

1.2.2 Сайт «Bookcrossing.com»

Наступним прикладом реалізації буккросинг системи є «Bookcrossing.com». Даний сайт був розроблений командою з Сандпойнти, штат Айдахо. Даний сайт функціонує завдяки волонтерським зборам з усього світу.

Також «Bookcrossing.com» є всесвітньою платформою (рис. 1.6). Для його використання та обміном книг з іншими, зовсім неважливо в якій країні знаходиться користувач. За для цього була виконана велика кількість функціонала. Це значно розширює можливості для будь-якого користувача.



Рисунок 1.6 – «Bookcrossing.com» головна сторінка

Головною перевагою сайту «Bookcrossing.com» є зручність пошуку книг. Одним із функціоналом є світова карта для перегляду книг, що додаються на сайт (рис. 1.7).

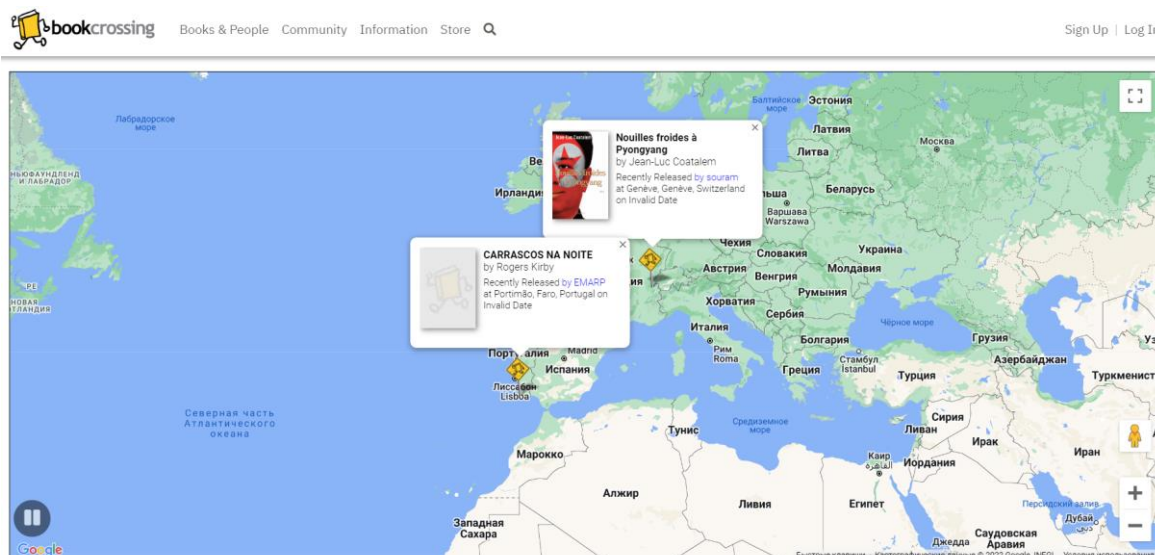


Рисунок 1.7 – «Bookcrossing.com» пошук книг на карті

На рисунку 1.8 представлений інший пошук книги за назвою, автором, типом та іншим. Також є можливість сортувати дані.

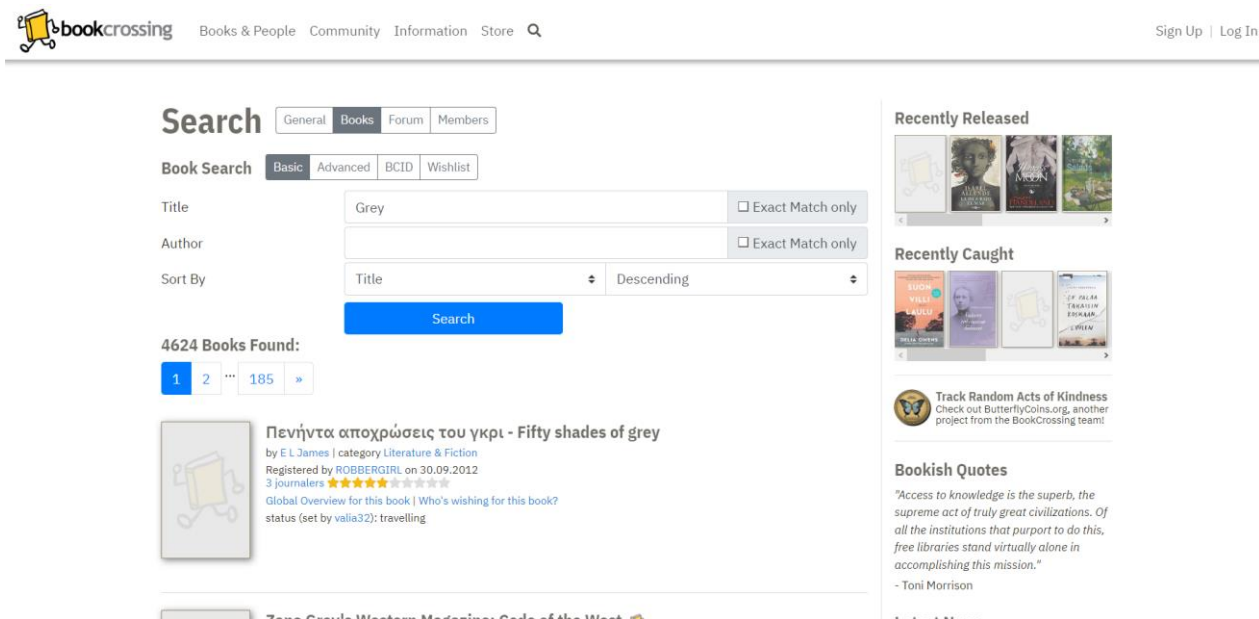


Рисунок 1.8 – «Bookcrossing.com» пошук книг

При обранні книги при пошуку, користувач перенаправляється на сторінку із детальною інформацією (рис. 1.9). На цій сторінці є також відгуки тих, хто вже обмінювався цією книгою.



Zane Grey's Lassiter: Brother Gun
by Jack Slade | Westerns | This book has not been rated.
ISBN: 9780843962383 [Global Overview for this book](#)
Registered by  KarenBC of Prince George, British Columbia Canada on 04.10.2019
Buy from one of these Booksellers:
[Amazon.com](#) | [Amazon UK](#) | [Amazon CA](#) | [Amazon DE](#) | [Amazon FR](#) | [Amazon IT](#) | [Bol.com](#)
  This book is in a Controlled Release!

1 journaler for this copy...

 Journal Entry 1 by  KarenBC from Prince George, British Columbia Canada on 4 октября 2019 r.
Lassiter, the solitary hero from Zane Grey's Riders of the Purple Sage, became one of the greatest Western legends of all time. Now America's favorite roughrider is back in further adventures filled with gun-slinging action and rawhide-tough characters.

 Journal Entry 2 by  KarenBC at Lac La Hache, British Columbia Canada on 4 октября 2019 r.
Released 3 yrs ago (04.10.2019 UTC) at Lac La Hache, British Columbia Canada

CONTROLLED RELEASE NOTES:

This book will be traveling to my family in the Cariboo region of British Columbia - when they are finished with it they will pass it along for new readers to enjoy.

Recently Released



Recently Caught





 **Track Random Acts of Kindness**
Check out ButterflyCoins.org, another project from the BookCrossing team!

Bookish Quotes

"From your parents you learn love and laughter and how to put one foot before the other. But when books are opened you discover that you have wings."
- Helen Hayes


Рисунок 1.9 – «Bookcrossing.com» сторінка обраної книги

На сайті «Bookcrossing.com» є також багато популярних форумів, на яких користувачі обговорюють різні тематики, книги та взаємодіють один з одним (рис. 1.10).


 Books & People Community Information Store 

Sign Up | Log In

The Forums

[BookCrossing Site Announcements](#) 6996  Search

Official site announcements from management, Support and Newsletter teams and from our Supply Store. Check here frequently to keep up-to-date on such things as new feature notifications, site updates, store items and general BC "official" news. Members may comment in threads, but not post announcements here. Please use the Community Bulletin Board for member generated news.

[Community Bulletin Board](#) 1815


News to share, or something to shout about? Put it on the Community Bulletin Board. A place for any special announcements to the community, including info on upcoming special events (also see our [Conventions and Meetings forum](#)), community activities, new media exposure, local BookCrossing gatherings, etc.
If you're uncertain if your post fits here, try this test: would you hang a notice about this on an actual bulletin board in your community?

If you have a question, the place to go is [here](#).


[Conventions and Meetings](#) 5383


Meeting other BookCrossers is great! Planning a special BookCrossing get-together in your area? Use this forum for planning and announcements.
If you're using a device and browser that support emojis, please mark your thread with the emoji for the country where the event is taking place. If not, then don't worry because BC Support will do it for you.

最新放生



最新捕獲



 **Track Random Acts of Kindness**
Check out ButterflyCoins.org, another project from the BookCrossing team!

Bookish Quotes

"Why do writers write? Because it isn't there."
- Thomas Berger

Latest News

Рисунок 1.10 – «Bookcrossing.com» форум

До переваг сайту «Bookcrossing.com» відноситься й велика кількість інформації. Як, наприклад, на рисунку 1.11 представлена інформація про обмін книгами та систему буккросингу. Виділення окремих сторінок для цього є гарним рішенням.

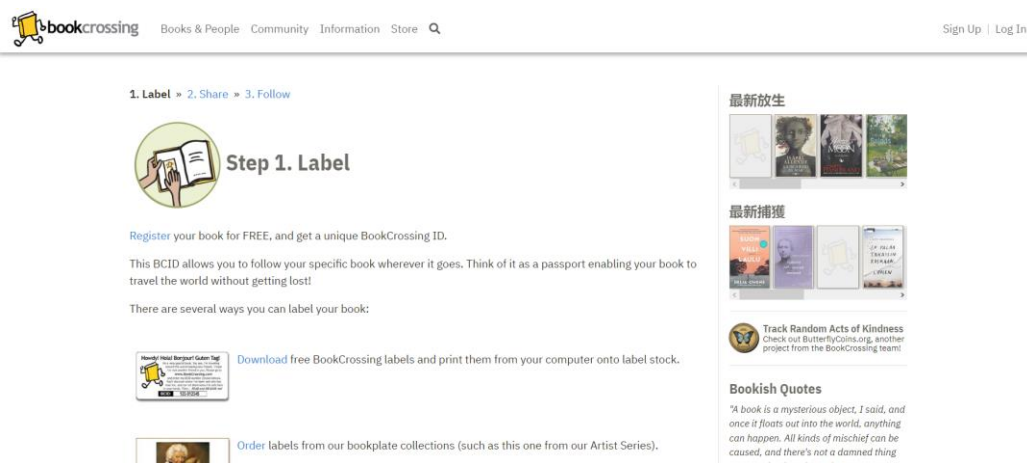


Рисунок 1.11 – «Bookcrossing.com» інформація про обмін книгам

У розділі з інформацією, на окрему сторінку було винесено інформацію що до розробників (рис. 1.12). Коротка історія про команду та зародження ідеї створення всесвітньої мережі для буккросингу може привабити користувачів сайту «Bookcrossing.com».

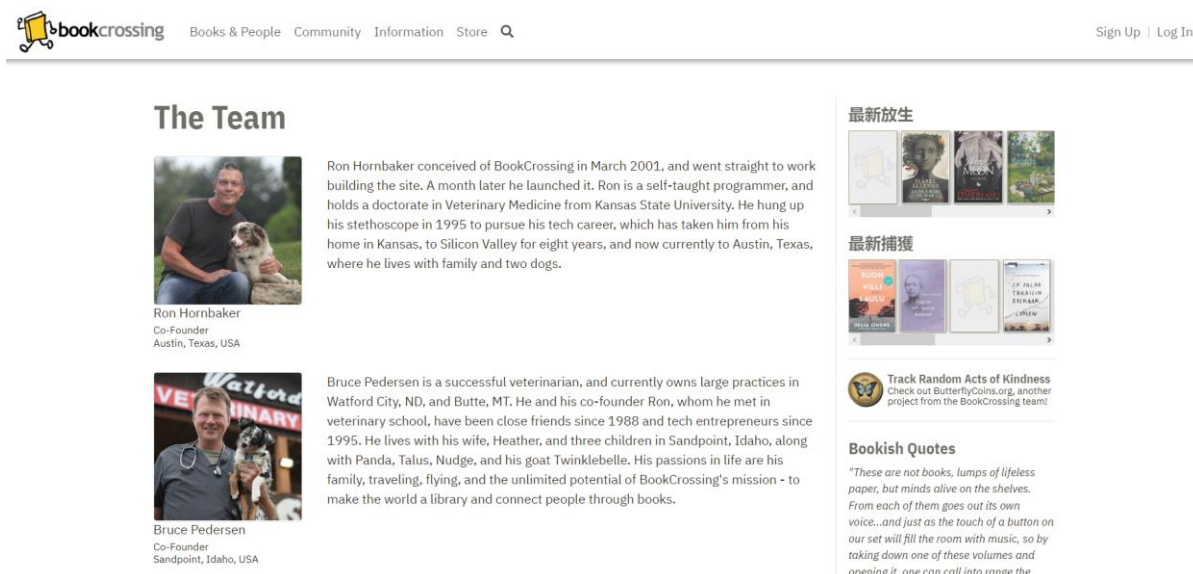


Рисунок 1.12 – «Bookcrossing.com» інформація про засновників

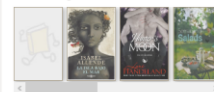
Ще одні із переваг «Bookcrossing.com» є перегляд топ користувачів сайту, що мають найбільшу кількість обмінами книг та коментарів (рис. 1.13). На сторінці можна регулювати в якому саме напрямі виводити топ користувачів.

Leader Board - Books Registered

[leaders - books](#) | [leaders - releases](#) | [leaders - referrals](#) | [newest members](#)
Show Top

Books Registered All-Time			Books Registered Last Week		
#	Member	Amount	#	Member	Amount
1	countofmonte	87 120	1	KarenBC	217
2	KarenBC	57 559	2	sophianibris	210
3	solittletime	55 713	3	sognalibro	144
4	pjlareau	50 350	4	Andrea-Berlin	134
5	Doris52	46 747	5	PNProLoco	129
6	ekorren	46 509	6	Doris52	120
7	bookasale	44 202	7	UNIVALI	113
8	Biblioamericana	36 978	8	SISBIUFU	111
9	Ghanescha	35 563	9	FTranvia	103
10	DameEdna	34 426	10	imeretoo	101
11	RoseOfDarkness	33 564	11	Zwergengewelt	97
12	bookstogive	33 318	12	Maranase	93
13	YYCanuck	30 275	13	Quintanalara	92

Recently Released



Recently Caught



Track Random Acts of Kindness
 Check out ButterflyCoins.org, another project from the BookCrossing team!

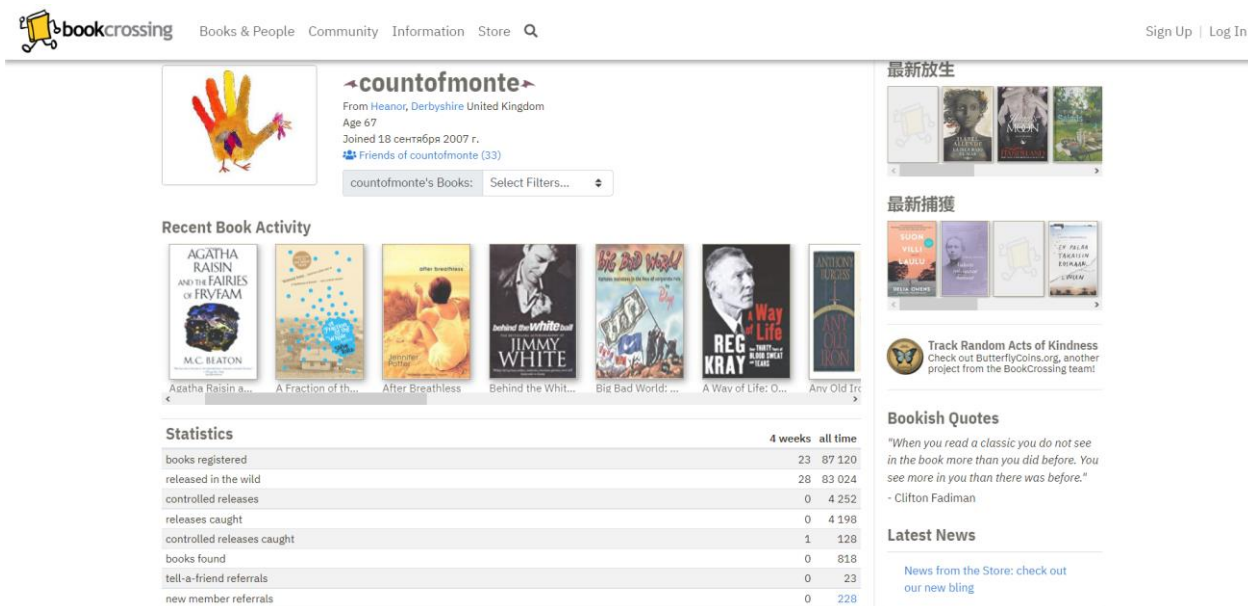
Bookish Quotes

"Reading is thinking with someone else's head instead of one's own"
 - Arthur Schopenhauer

Latest News

Рисунок 1.13 – «Bookcrossing.com» перегляд топ користувачів

Перейшовши на сторінку користувача сайту «Bookcrossing.com», можна побачити детальну інформацію про вік, дату реєстрації, статистику та інше (рис. 1.14).



countofmonte
 From Heanor, Derbyshire United Kingdom
 Age 67
 Joined 18 сентября 2007 г.
 Friends of countofmonte (33)

countofmonte's Books:

Recent Book Activity

Agatha Raisin a... A Fraction of th... After Breathless Behind the Whit... Big Bad World: ... A Way of Life: O... Any Old Tr...

Statistics

	4 weeks	all time
books registered	23	87 120
released in the wild	28	83 024
controlled releases	0	4 252
releases caught	0	4 198
controlled releases caught	1	128
books found	0	818
tell-a-friend referrals	0	23
new member referrals	0	228

Bookish Quotes

"When you read a classic you do not see in the book more than you did before. You see more in you than there was before."
 - Clifton Fadiman

Latest News

News from the Store: check out our new bling

Рисунок 1.14 – «Bookcrossing.com» сторінка користувача

Крім того, користувачі сайту можуть писати додаткову інформацію про цікаві книжні знахідки (рис. 1.15). До кожного допису можна додавати текст та посилання.

Some of my catches:

www.bookcrossing.com/---/9779112/

Going to India...

www.bookcrossing.com/---/8113400

Another one going to India, journalled by someone from Split-Dalmatia, Croatia...

www.bookcrossing.com/---/9765912

Someone loves this book...

www.bookcrossing.com/---/8275228

This one caught at Stratford-on-Avon. The finder hopes BC isn't cranky!

www.bookcrossing.com/---/8045204

This book inspired the finder who took it to the South Africa World Cup...

www.bookcrossing.com/---/7854349

The finder got some other bargains when she found this book...

www.bookcrossing.com/---/8548211

Рисунок 1.15 – «Bookcrossing.com» дописи користувачів

До переваг сайту «Bookcrossing.com» можна віднести якісно виконану сторінку із контактною інформацією (рис. 1.16). Кожен користувач може поставити питання чи занести побажання по якомусь функціоналу.

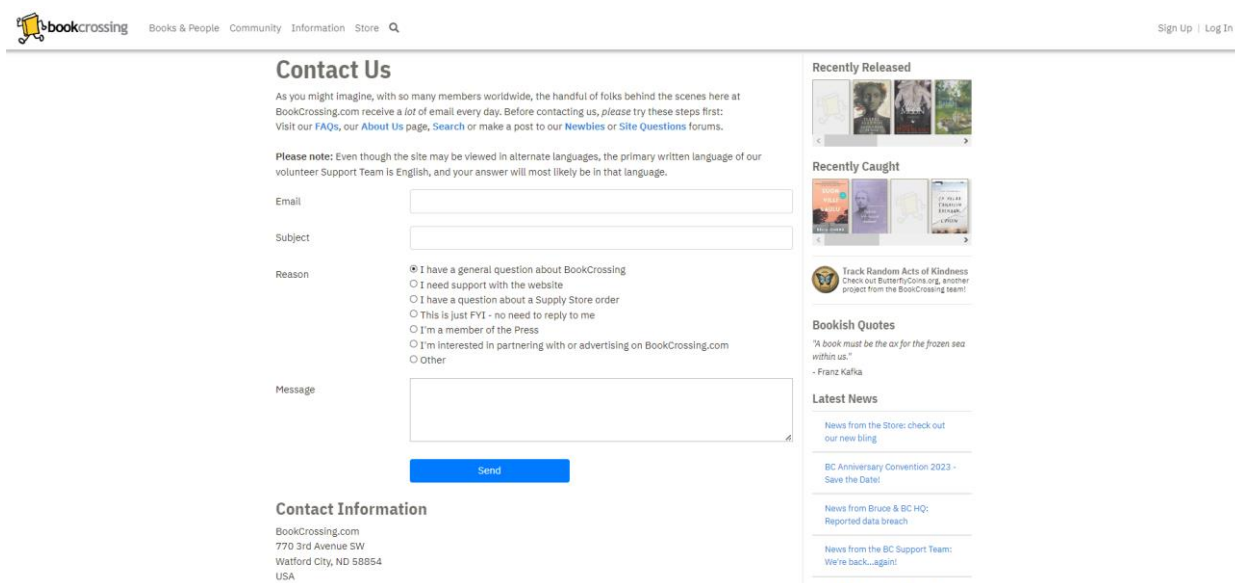


Рисунок 1.16 – «Bookcrossing.com» сторінка із контактною інформацією

Отже, сайт «Bookcrossing.com» є гарним приклад реалізації системи підтримки буккросингу. Сайт має багато переваг, але головним недоліком є орієнтованість на жителів Америки, Англії та Австралії. Для країн Європи є складнощі та значні затримки в обміні книгами.

1.2.3 Facebook група «Bookcrossing Lviv»

Останнім прикладом буккросингу є facebook група з назвою «Bookcrossing Lviv». Через нестачу ресурсів та якісних сайтів для буккросингу, користувачі соціальних мереж самостійно створюють групи людей за інтересами та місцем проживання для обміну книгами (рис. 1.17).

Facebook група з назвою «Bookcrossing Lviv» має більше ніж сто учасників, але переважні їх частина не активні. Крім того, переглядаючи дописи групи, можна побачити, що на цей час її використовують рідко (рис. 1.18) [19].



Рисунок 1.17 – Facebook група «Bookcrossing Lviv» головна сторінка

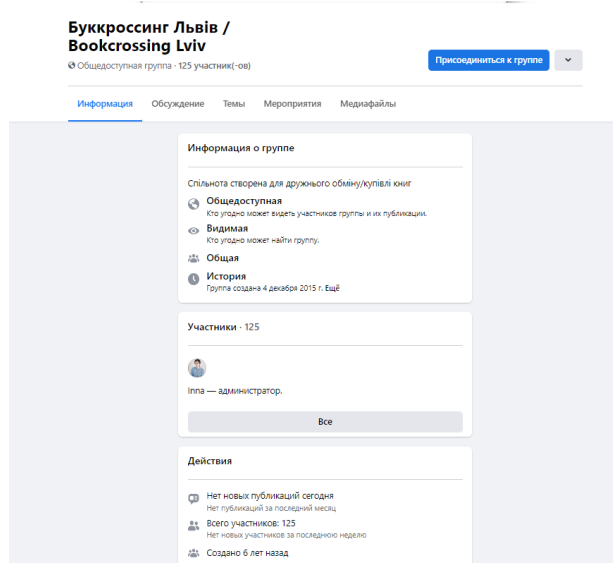


Рисунок 1.18 – Facebook група «Bookcrossing Lviv» детальна інформація

Також, можна побачити тенденцію, що такі групи використовують не для обміну книгами, що представляє саме з себе буккросинг, а саме продаж (рис. 1.19). Дані публікації та пропозиції не відповідають заданій тематиці.

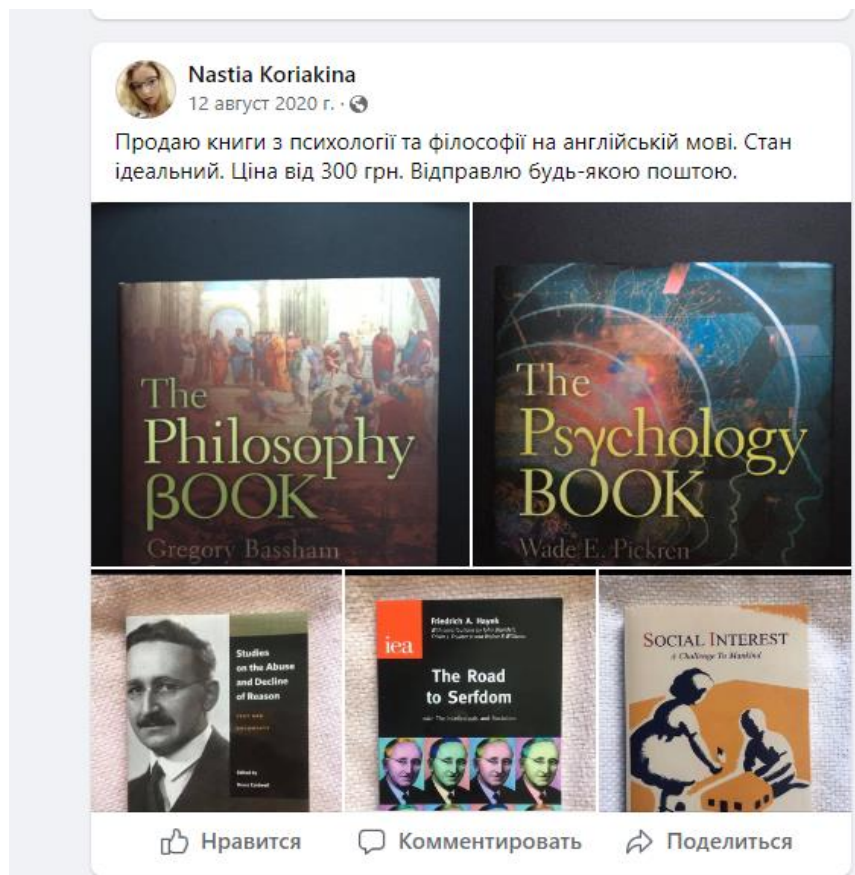


Рисунок 1.19 – Facebook група «Bookcrossing Lviv» приклад посту

Отже, створені групи у соціальних мережах підтримки буккросингу не є ефективним інструментом для проведення популяризації даного напрямку. Соціальні мережі не дають можливості реалізувати весь необхідний функціонал.

1.2.4 Висновки та порівняльна характеристика аналогів

По тематиці буккросинг було обрано такі сайти аналоги інформацій системі:

- «Bookcrossing-ua.at.ua»;
- «Bookcrossing.com»;
- Facebook група «Bookcrossing Lviv».

Отже, було проведено аналіз кожного сайту, що представлено в таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика сайтів підтримки буккросингу

Критерії порівняння	«Bookcrossingua.at.ua»	«Bookcrossing.com»	Facebook група «Bookcrossing Lviv»	Інформаційна система підтримки буккросингу
Можливість зареєструватися	-	+	+	+
Написання коментаря під постами	-	-	+	+
Перегляд інформації про інші книги	+	+	+	+
Додавання інформації про власні книги	+	+	-	+
Обмін книгами із іншими користувачами	-	+	-	+
Залишення коментаря про обмін чи книгу	-	+	-	+
Комунікація із іншими користувачами	+	+	+	+
Створення власних дописів на сторінці	-	-	-	+
Встановлення власних уподобань у книгах для обміну із іншими користувачами	+	-	+	+
Можливість задати запитання адміністратору сайту	-	+	+	+
Зручна мобільна адаптація	-	+	+	+

Отже, представлено порівняльну характеристику «Bookcrossing-ua.at.ua», «Bookcrossing.com», facebook група «Bookcrossing Lviv» з інформаційною системою підтримки буккросингу. Весь функціонал буде виконаний у подальшій розробці.

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1. Мета та задачі дослідження

Метою інформаційної системи підтримки буккросингу є розробка інформаційної системи підтримки буккросингу із реалізацією усіх можливостей для взаємодії користувачів між собою. Користувачі інформаційної системи повинні мати можливість провести обмін книгами через створену систему. Обмін повинен виконуватися зі згодою обох користувачів.

Для досягнення поставленої мети, а саме розробка інформаційної системи підтримки буккросингу, потрібно реалізувати увесь перелік функціонала.

Розглянемо функціональні вимоги:

- Авторизація та реєстрація в інформаційній системі підтримки буккросингу;
- Можливість задати запитання адміністратору інформаційної системи;
- Додавання користувачем дописів про книги для обміну;
- Редагування створених користувачем дописів про книги для обміну;
- Встановлення власних уподобань в книгах для підбору книг на обмін (жанр, автор, видавництво та інше);
- Можливість написати іншому зареєстрованому користувачеві інформаційної системи підтримки буккросингу;
- Можливість поскаржитися на іншого користувача інформаційної системи
- Подання запиту для обміну книгами;
- При отриманні запиту є можливість підтвердити чи відхилити запит від іншого користувача;
- При оформленні замовлення кожен із учасників додаватиме посилання;
- Залишення відгуку про обмін книгами з користувачем в п'ятибальній шкалі;
- Редагування основної інформації в системі підтримки буккросингу;

- Блокування користувачів інформаційної системи підтримки буккросингу;
- Оформлення розсилки на пошту користувачам про події інформаційної системи підтримки буккросингу.

Розглянемо не функціональні вимоги:

- Перегляд інформації, що вже додана в інформаційну систему;
- Фільтрація при пошуку книги для обміну;
- Можливість відповісти користувачу на запитання.

2.2. Розробка візуальної частини інформаційної системи

Розробка інформаційної системи для буккросингу розділювалася на певні етапи для структурованої поетапної праці над кожним пунктом. Першим кроком розробки була візуальна частина інформаційної системи, де створення front-end лягло на плечі HTML5 та CSS. Елементарні інструменти створення візуальної частини сайту були використані для оптимізації верстки сайту [20].

Для створення додаткової візуальної привабливості є JavaScript. Інструментарій цього програмного забезпечення дає можливість не лише покращити візуальну привабливість, шляхом додавання цікавих анімацій, а й тонко налаштувати взаємозв'язки між користувачами. Універсальність цієї мови програмування дозволяє розширити функціонал, що своєю чергою позитивно відображається на написанні коду, шляхом прискорення цього процесу [21].

Для знаходження підходящого фреймворку, потрібно провести аналіз, серед яких будуть Vue.js, Angular.js, React.js.

Першим в порівнянні – Vue.js, який має як свої переваги, так і недоліки. Не дивлячись на деяку складність у написанні коду, даний фреймворк має все більшу аудиторію. Бібліотеки не вражають своєю кількістю, але фреймворк підтримує більшість IDE. Супутня документація реалізована у вигляді бібліотеки з чіткою структурою та власною системою навігації, що полегшує доступ до потрібної інформації [22].

Angular.js – досить складний фреймворк з високим рівнем для входу в процес розробки, але це обумовлено прив'язкою даних, яке має подвійний характер. Автоматизація процесів зі сторони сервера позитивно відображається на його продуктивності, але з іншої сторони це негативно відображається на часі завантаження сторінки та на повному відображенні програмного забезпечення в цілому [23].

Одним із найбільш стабільніших фреймворків є React.js. Перевагами цього фреймворку також є продуктивність системи та тонке налаштування. Ці переваги мають велику ціну – складнощі при розробці [24].

Аналіз фреймворків показав головні переваги та недоліки серед яких Vue.js виділяється своєю простотою, що є головним критерієм при виборі для створення інформаційної системи.

Фреймворк не потребує високого рівня знань, а потрібна документація дає можливість легко в ній орієнтуватися. Недоліки цей фреймворк теж має, але в межах цієї інформаційної системи, вони незначні та складнощі при розробці не виникне [25].

2.3. Розробка функціональної частини інформаційної системи

Створення back-end – важкий етап розробки, адже постійна взаємодія з веб-службами, бібліотеками та різноманітними API потребує постійного контролю та приділення уваги кожному аспекту зі сторони розробника. Ретельний підбір потрібного фреймворка пов'язаний із полегшенням створення back-end [26].

Серед великого набору фреймворків, достатньо виділити три найбільш підходящих.

Laravel оперує великим набором переваг, серед яких виділяється документація з чіткою структурою, різноманітний набір шаблонів, маршрутизація та власний CLI. Негативною стороною є ресурсозатратність для побудови, що підвищує час розробки [27].

Оскільки інформаційна система не має великого масштабу, то доречно розглянути Spring Boot, який має власне програмне забезпечення з прив'язкою до хмарних технологій. Потрібно виділити високий поріг входу, що негативно відображається на часі розробки.

Сучасна система Node.js приваблює розробників своєю швидкістю обробки процесів та роботою з аудиторією. Серед недоліків виділяється відсутність фахівців та високий поріг входу [28].

Аналіз програмного забезпечення для створення back-end показав непомірно високий поріг входу для більшості фреймворків, тому було взято Laravel. Фреймворк має ряд переваг та недоліків, але краще підходить під поставлені критерії для створення даної інформаційної системи [29].

Для створення бази даних буде використаний MySQL. MySQL – універсальний інструмент з підтримкою більшості таблиць. Серед них можна виділити таблиці з підтримкою пошуку повного тексту (MyISAM), таблиці з підтримкою транзакцій цільовим пошуком (InnoDB). MySQL має свої спеціалізовані таблиці в яких легко орієнтуватися та створювати власні [30].

3 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ ІС

3.1 Структура інформаційної системи

Проектування та інформаційної системи починається із структуризації всієї інформаційної системи. Структура ділиться на етапи розробки, де створюються задачі та підзадачі, які вирішуються за допомогою програмного забезпечення та їх інструментів.

Аналіз предметної області та розробка – це окремі етапи створення інформаційної системи. Та в реальності ці етапи взаємопов'язані, бо створення інформаційної системи бере початок на етапі створення цілей та задач, які повинен виконувати проект, та продовжує удосконалюватися до етапу тестування для вдосконалення системи. Все це робить шлях створення проекту розгалуженим та складним.

Перше враження від переходу на головну веб-сторінку повинне бути позитивним, та зацікавити випадкового користувача скористатися інформаційною системою.

Головна сторінка, яка зустрічає користувача надає доступ до таких модулів:

- Головна;
- Реєстрація;
- Авторизація.

Функціонал цих модулів має елементарний характер та надає можливість переглядати інформацію головної сторінки, зареєструвати нового користувача, або авторизувати вже зареєстрованого користувача.

Основні можливості інформаційна система надає лише зареєстрованим користувачам. Зареєстровані користувачі не мають якоїсь ієрархії та не відрізняються між собою за різновидом функціонала.

Інформаційна система для буккросингу має елементарний характер із мінімалістичним стилем оформлення візуальної частини. Зареєстрований користувач має можливість обмінюватися літературними витворами мистецтва,

переглядати наявність потрібних книг в інших користувачів та коментувати шляхом користування форуму.

3.2 Структурно-функціональне моделювання процесу

Розглянемо IDEF0 діаграму інформаційної системи підтримки буккросингу (рис. 3.1).

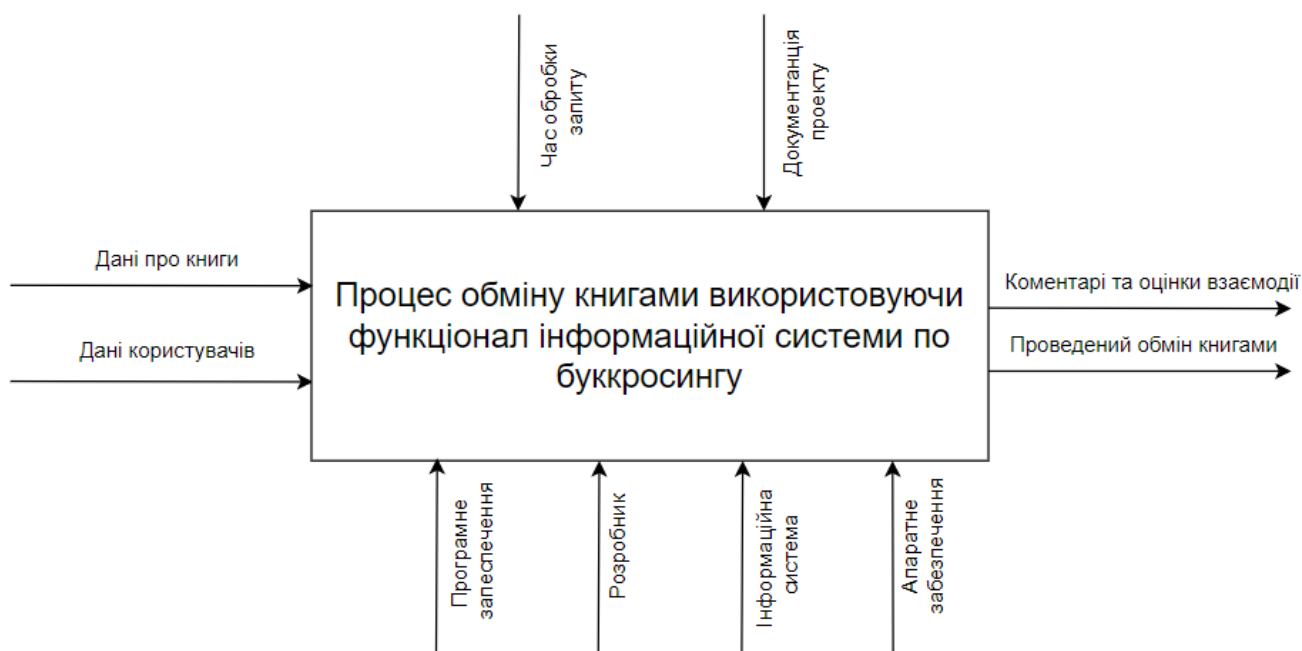


Рисунок 3.1 – IDEF0 інформаційної системи підтримки буккросингу

Наступним етапом є проведення детальної декомпозиції контекстної діаграми на функціональні блоки:

- Реєстрація користувачів в системі підтримки буккросингу
- Перегляд інформації про книги.
- Подання запиту на обмін книгами з іншим користувачем системи підтримки буккросингу.
- Обговорення усієї інформації та надання даних для відправки.
- Звітність про проведений обмін книгами по системі буккросингу.

Розглянемо IDEF1 інформаційної системи підтримки буккросингу (рис. 3.2).

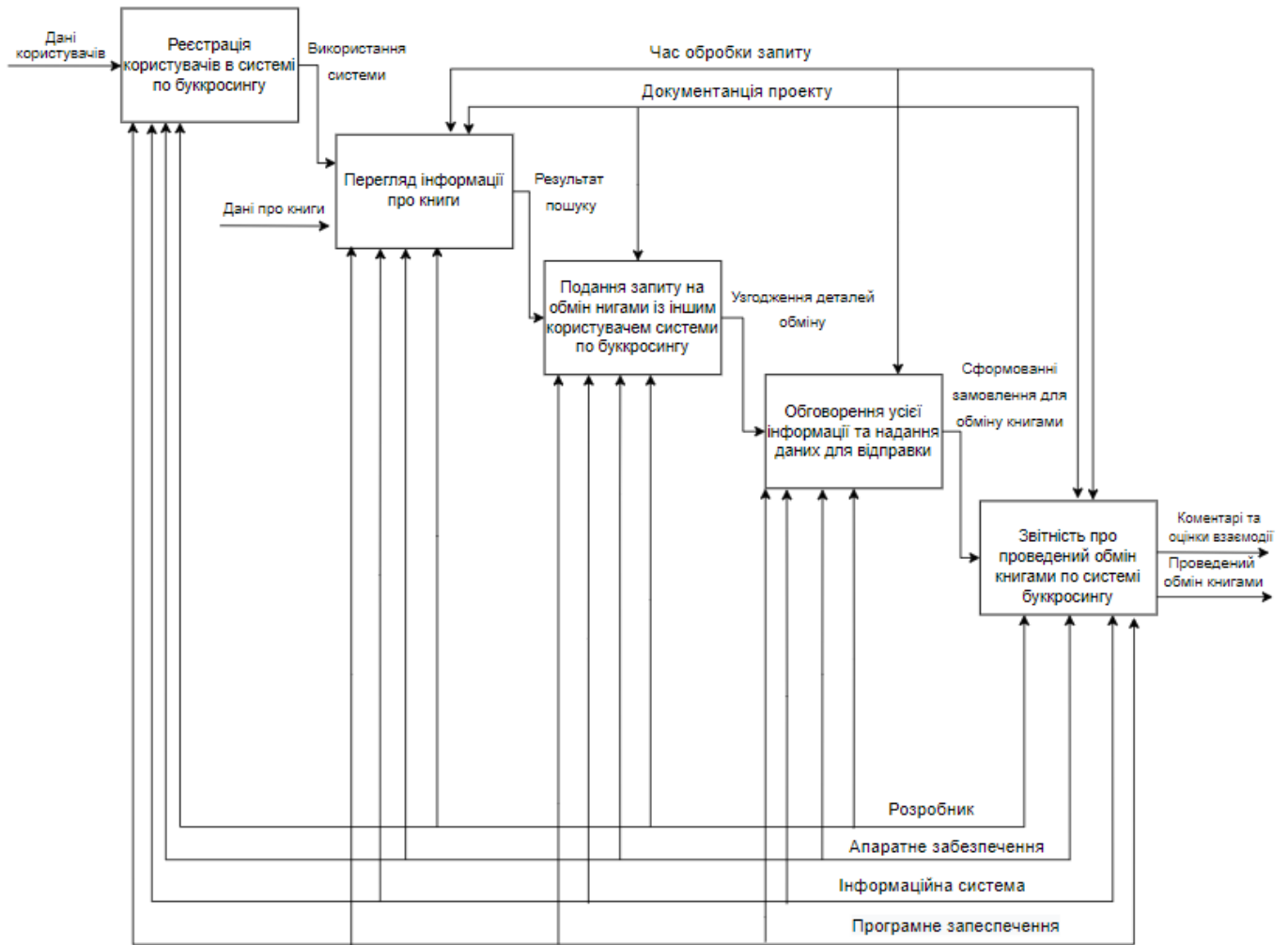


Рисунок 3.2 – IDEF1 інформаційної системи підтримки буккросингу

3.3 Моделювання діаграми варіантів використання

Для представлення функціоналу інформаційної системи підтримки буккросингу було розроблено діаграму варіантів використання (табл.3.1-2).

Таблиця 3.1. – Опис акторів

№	Іменування акторів	Детальний опис
1	Адміністратор	Користувач, що може видозмінювати та редагувати головну інформацію в інформаційній системі підтримки буккросингу.
2	Зареєстрований користувач	Користувач має можливість використовувати розширені можливості інформаційної системи
3	Не зареєстрований користувач	Користувач має обмежені можливості інформаційної системи

Таблиця 3.2. – Опис варіантів використання

№	Варіанти використання	Детальний опис
1	Обмін книгами	Функціонал для обміну книгами між користувачами системи.
2	Подання запиту	Функціонал для подання запиту на обмін.
3	Фільтрація при пошуку	Функціонал для фільтрації та пошуку книг для обміну.
4	Редагування даних для обміну	Функціонал для редагування даних при обміні книгами.
5	Редагування даних	Функціонал для редагування власних даних та їх редагування.
6	Авторизація	Функціонал для авторизації.
7	Реєстрація	Функціонал для реєстрації в системі.
8	Перегляд інформації в інформаційній системі	Функціонал для перегляду інформації в інформаційній системі зареєстрованими та не зареєстрованими користувачами.

№	Варіанти використання	Детальний опис
9	Блокування користувачів	Функціонал адміністраторів для блокування користувачів.
10	Редагування даних в системі	Функціонал для редагування даних в системі.
11	Додавання інформації про книг на власну сторінку	Функціонал для додавання

Сформувавши представлення інформаційної системи підтримки буккросингу було виконано діаграму варіантів використання на рисунку 3.3-4.



Рисунок 3.3 – Діаграма варіантів використання для користувачів системи



Рисунок 3.4 – Діаграма варіантів використання для адміністратора

3.4. Проектування бази даних

Користувачі інформаційної системи поділяються на зареєстрованих та незареєстрованих. Функціонал незареєстрованого користувача дуже обмежений, тому вся увага прикута до зареєстрованого користувача. Зареєстрований користувач може ініціювати обмін книгами та відмовлятися від пропозицій обміну.

База даних створена на основі поставлених задач та враховуючи всі вимоги до функціонала інформаційної системи. Важливим аспектом є створення обговорення в форматі форуму, що розширює загальний функціонал проекту.

ER-діаграма – це візуалізація зв'язків між користувачем та інформаційною системою показаних на рисунку 3.5. Інформаційна система має визначений функціонал для розв'язання конкретних задач поставлених користувачем. У таблиці 3.3 вказані імена таблиць, поля, їх зміст та тип, вид ключа та обмеження.

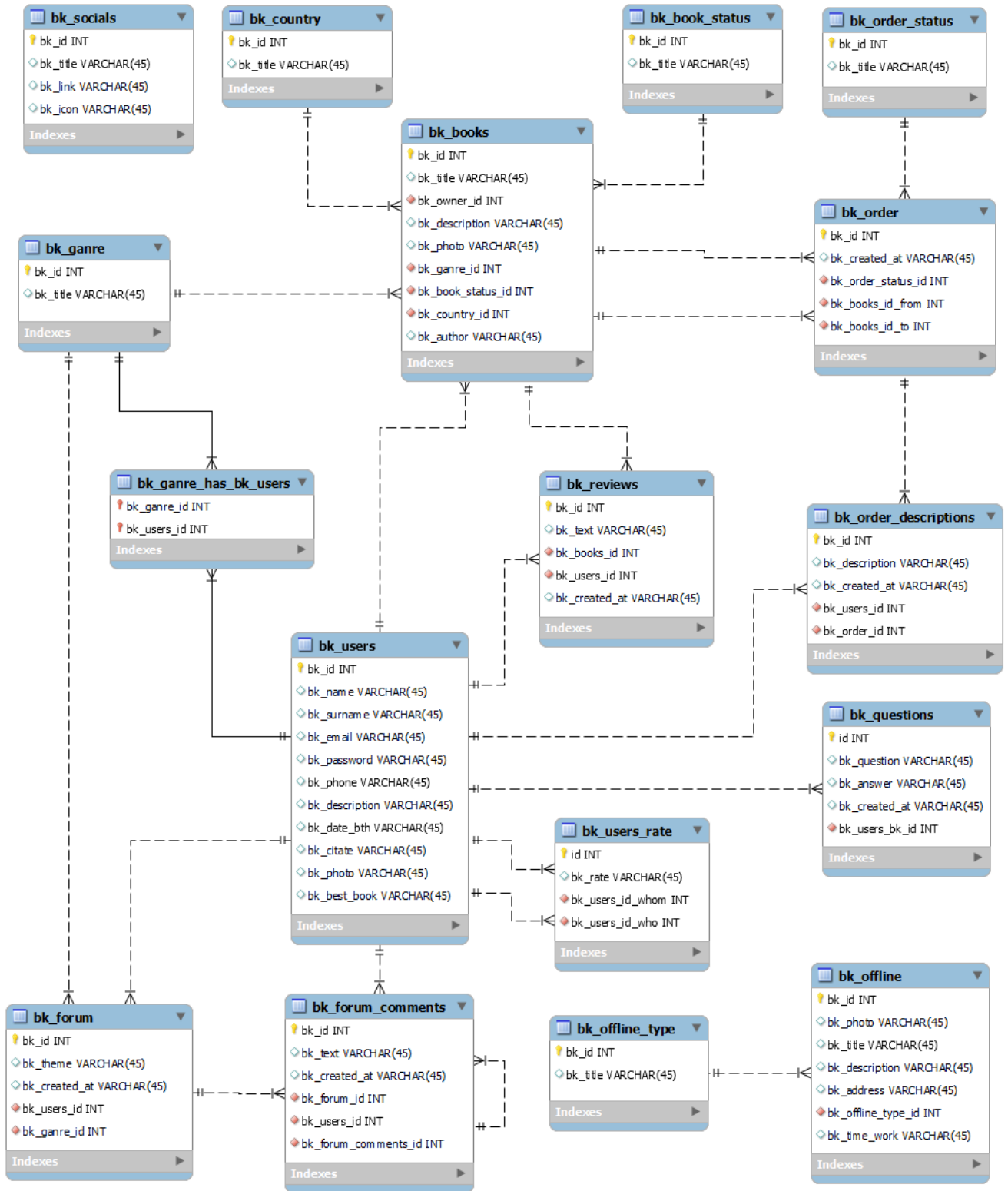


Рисунок 3.5 – ER-діаграма

Таблиця 3.3 – Структура за ER-діаграми інформаційної системи для буккросингу

№	Таблиця	Поле	Зміст	Тип	Ключі	Обмеження
1	bk_users	bk_id	Ідентифікатор користувача	INTEGER	PK	Не порожній
		bk_name	Ім'я користувача	VARCHAR(45)		Не порожній
		bk_surname	Прізвище користувача	VARCHAR(45)		Не порожній
		bk_email	Електронна пошта	VARCHAR(45)		Не порожній
		bk_password	Пароль користувача	VARCHAR(45)		Не порожній
		bk_phone	Номер телефону	VARCHAR(45)		
		bk_description	Опис користувача	VARCHAR(45)		
		bk_date_bth	Дата народження	VARCHAR(45)		
		bk_citate	Цитата користувача	VARCHAR(45)		
		bk_photo	Фото користувача	VARCHAR(45)		
		bk_best_book	Улюблена книга користувача	VARCHAR(45)		
2	bk_forum_comments	bk_id	Ідентифікатор коментаря	INTEGER	PK	Не порожній
		bk_text	Текст коментаря	VARCHAR(45)		Не порожній
		bk_created_at	Дата створення	VARCHAR(45)		Не порожній
		bk_forum_id	Ідентифікатор теми	INTEGER	FK	Не порожній
		bk_users_id	Ідентифікатор користувача	INTEGER	FK	Не порожній
		bk_forum_comments_id	Ідентифікатор відповіді	INTEGER	FK	
3	bk_forum	bk_id	Ідентифікатор теми	INTEGER	PK	Не порожній
		bk_theme	Опис теми	VARCHAR(45)		Не порожній

№	Таблиця	Поле	Зміст	Тип	Ключі	Обмеження
		bk_created_at	Дата створення	VARCHAR(45)		Не порожній
		bk_users_id	Ідентифікатор користувача	INTEGER	FK	Не порожній
		bk_ganre_id	Ідентифікатор теми дискусії	INTEGER	FK	Не порожній
4	bk_ganre	bk_id	Ідентифікатор теми дискусії	INTEGER	PK	Не порожній
		bk_title	Опис дискусії	VARCHAR(45)		Не порожній
5	bk_ganre_has_bk_users	bk_ganre_id	Ідентифікатор теми дискусії	INTEGER	FK	Не порожній
		bk_users_id	Ідентифікатор користувача	INTEGER	FK	Не порожній
6	bk_books	bk_id	Ідентифікатор книги	INTEGER	PK	Не порожній
		bk_title	Назва книги	VARCHAR(45)		Не порожній
		bk_owner_id	Ідентифікатор власника книги	INTEGER	FK	Не порожній
		bk_description	Опис книги	VARCHAR(45)		Не порожній
		bk_photo	Фото книги	VARCHAR(45)		Не порожній
		bk_ganre_id	Жанр книги	INTEGER	FK	Не порожній
		bk_book_status_id	Статус книги	INTEGER	FK	Не порожній
		bk_country_id	Ідентифікатор країни	INTEGER	FK	Не порожній
		bk_author	Автор книги	VARCHAR(45)		Не порожній
7	bk_country	bk_id	Ідентифікатор країни	INTEGER	PK	Не порожній
		bk_title	Назва країни	VARCHAR(45)		Не порожній
8	bk_book_status	bk_id	Ідентифікатор статусу книги	INTEGER	PK	Не порожній
		bk_title	Назва статусу книги	VARCHAR(45)		Не порожній

№	Таблиця	Поле	Зміст	Тип	Ключі	Обмеження
9	bk_order	bk_id	Ідентифікатор обміну	INTEGER	PK	Не порожній
		bk_created_at	Дата створення	VARCHAR(45)		Не порожній
		bk_order_status_id	Статус замовлення	INTEGER	FK	Не порожній
		bk_books_id_from	Ідентифікатор книги до обміну	INTEGER	FK	Не порожній
		bk_books_id_to	Ідентифікатор книги після обміну	INTEGER	FK	Не порожній
10	bk_order_status	bk_id	Ідентифікатор ордера статусу	INTEGER	PK	Не порожній
		bk_title	Назва ордеру статусу	VARCHAR(45)		Не порожній
11	bk_order_descriptions	bk_id	Ідентифікатор контакту	INTEGER	PK	Не порожній
		bk_description	Опис контакту	VARCHAR(45)		Не порожній
		bk_created_at	Дата публікації контакту	VARCHAR(45)		Не порожній
		bk_users_id	Ідентифікатор користувача	INTEGER	FK	Не порожній
		bk_order_id	Ідентифікатор ордера	INTEGER	FK	Не порожній
13	bk_reviews	bk_id	Ідентифікатор відгуку	INTEGER	PK	Не порожній
		bk_text	Текст відгуку	VARCHAR(45)		Не порожній
		bk_books_id	Ідентифікатор книги	INTEGER	FK	Не порожній
		bk_users_id	Ідентифікатор користувача	INTEGER	FK	Не порожній
		bk_created_at	Дата публікації	VARCHAR(45)		Не порожній
14	bk_questions	bk_id	Ідентифікатор питання	INTEGER	PK	Не порожній

№	Таблиця	Поле	Зміст	Тип	Ключі	Обмеження
		bk_question	Текст запитання	VARCHAR(45)		Не порожній
		bk_answer	Текст відповіді	VARCHAR(45)		
		bk_created_at	Дата створення відповіді	VARCHAR(45)		Не порожній
		bk_users_bk_id	Ідентифікатор користувача зі створеним запитанням	INTEGER	FK	Не порожній
15	bk_users_rate	bk_id	Ідентифікатор оцінки	INTEGER	PK	Не порожній
		bk_rate	Оцінка	VARCHAR(45)		Не порожній
		bk_users_id_whom	Ідентифікатор користувача якому поставили оцінку	INTEGER	FK	Не порожній
		bk_users_id_who	Ідентифікатор користувача який поставив оцінку	INTEGER	FK	Не порожній
16	bk_offline	bk_id	Ідентифікатор закладу	INTEGER	PK	Не порожній
		bk_photo	Фото закладу	VARCHAR(45)		
		bk_title	Назва закладу	VARCHAR(45)		Не порожній
		bk_description	Опис закладу	VARCHAR(45)		Не порожній
		bk_address	Адреса закладу	VARCHAR(45)		Не порожній
		bk_offline_type_id	Ідентифікатор виду закладу	INTEGER	FK	Не порожній
		bk_time_work	Час роботи	VARCHAR(45)		Не порожній
17	bk_offline_type	bk_id	Ідентифікатор типу закладу	INTEGER	PK	Не порожній

№	Таблиця	Поле	Зміст	Тип	Ключі	Обмеження
		bk_title	Назва типу закладу	VARCHAR(45)		Не порожній
18	bk_social	bk_id	Ідентифікатор соціальних мереж	INTEGER	PK	Не порожній
		bk_title	Назва соціальних мереж	VARCHAR(45)		Не порожній
		bk_link	Посилання соціальних мереж	VARCHAR(45)		Не порожній
		bk_icon	Логотип соціальних мереж	VARCHAR(45)		Не порожній

4 ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

4.1 Програмна реалізація

Для досягнення мети в розробці інформаційної системи для буккросингу, було використано такі фреймворки:

- Vue.js для front-end;
- Laravel для back-end.

На рисунку 4.1 показано набір файлів, використаних при розробці інформаційної системи.

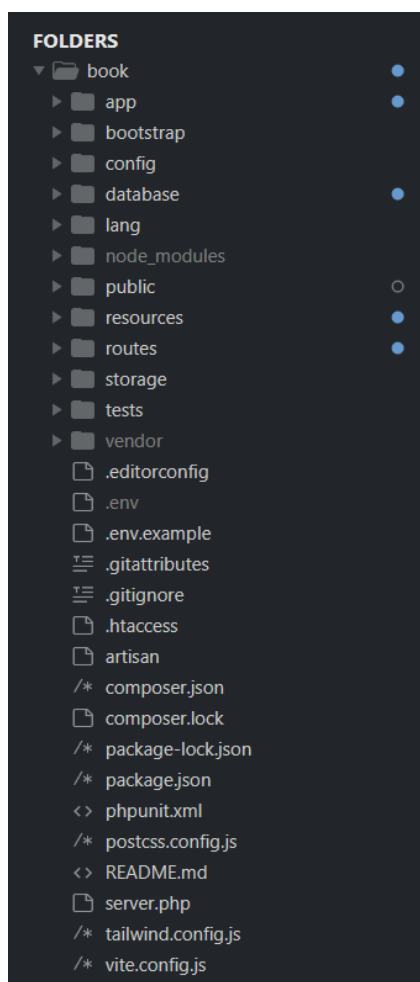


Рисунок 4.1 – Файли проекту

У системі розробки були використані такі файли: app, bootstrap, config, database, lang, public, resources, routes, storage, tests.

Після ретельного аналізу, отримана структура проекту інформаційної системи з шаблонами, файлами, ресурсами, даними та налаштуваннями.

4.2 Використання інформаційної системи зі сторони користувача

Використання інформаційної системи починається з відвідування презентованої сторінки за посиланням <http://bookcrossing-diplom.zzz.com.ua>. На рисунку 4.2 показана презентована сторінка, яка зустрічає не авторизованого користувача, тобто звичайного відвідувача.

Сторінка зустрічає мінімалістичним стилем з пропозицією зареєструватися, або ж авторизуватися, якщо особиста сторінка вже створена.

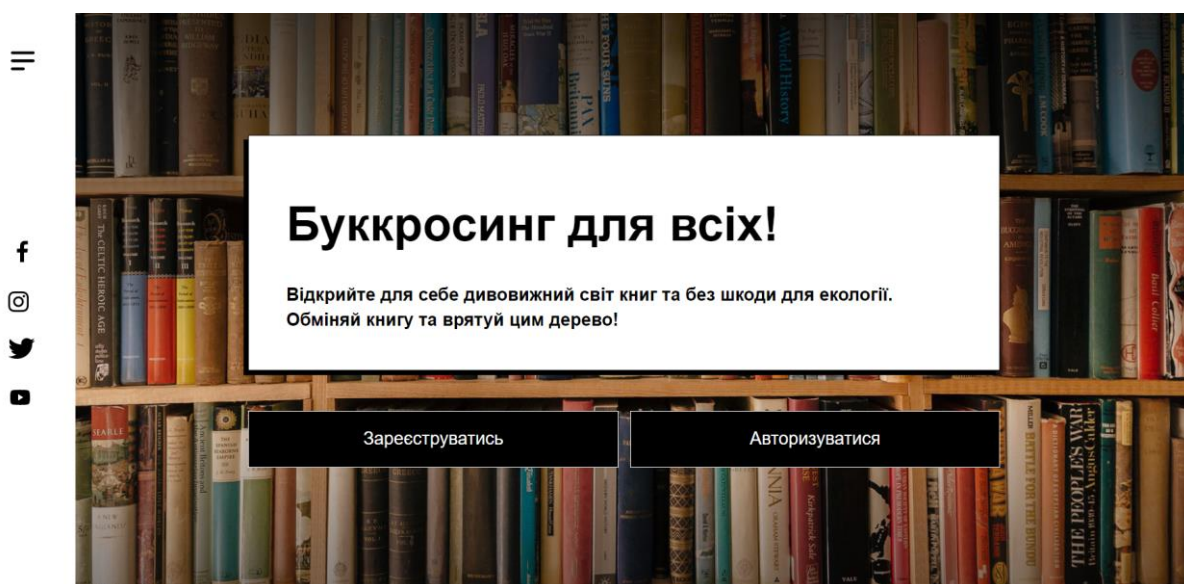


Рисунок 4.2 – Головна презентована сторінка

При реєстрації, мінімалістичний стиль супроводжує користувача та не відвертає увагу від заповнення потрібних полів.

Реєстрація нового користувача буде успішною після заповнення всіх полів, які обов'язково заповнити. Кнопка «Зареєструватися» буде натискатися після заповнення полів коректними даними.

Після успішної реєстрації, користувачеві буде запропоновано авторизуватися на рисунку 4.4. Автоматичний перехід на сторінку авторизації звільняє користувача

від додаткових переходів та економить час. Успішна авторизація буде після заповнення полів «Електронна пошта» та «Пароль», які вказував користувач при реєстрації акаунту.

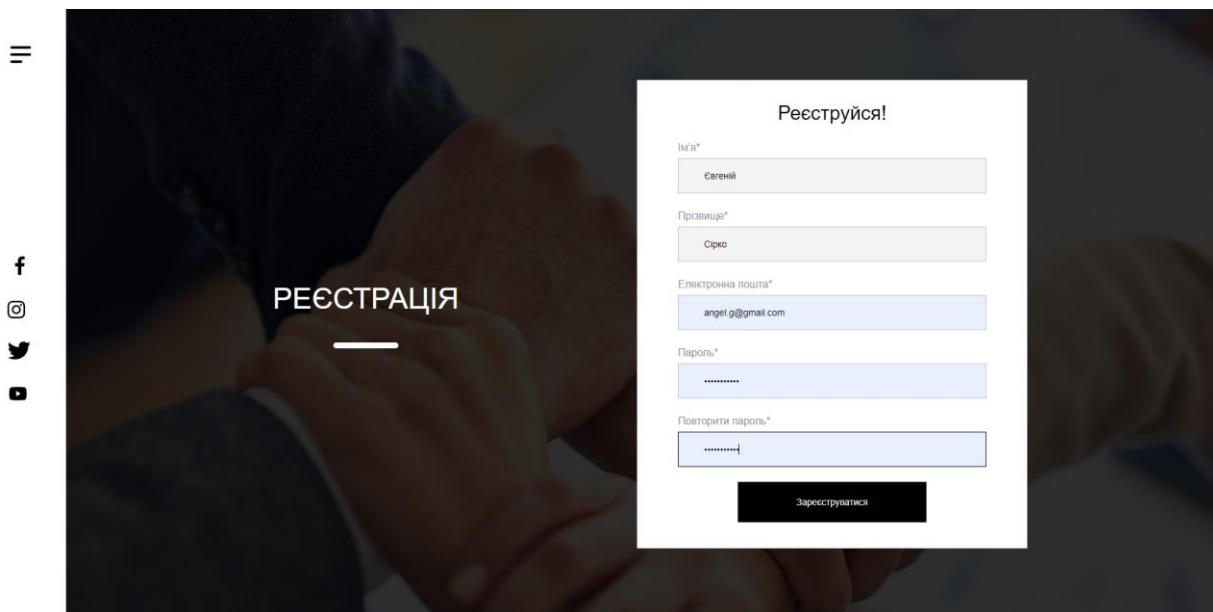


Рисунок 4.3 – Сторінка реєстрації нового користувача

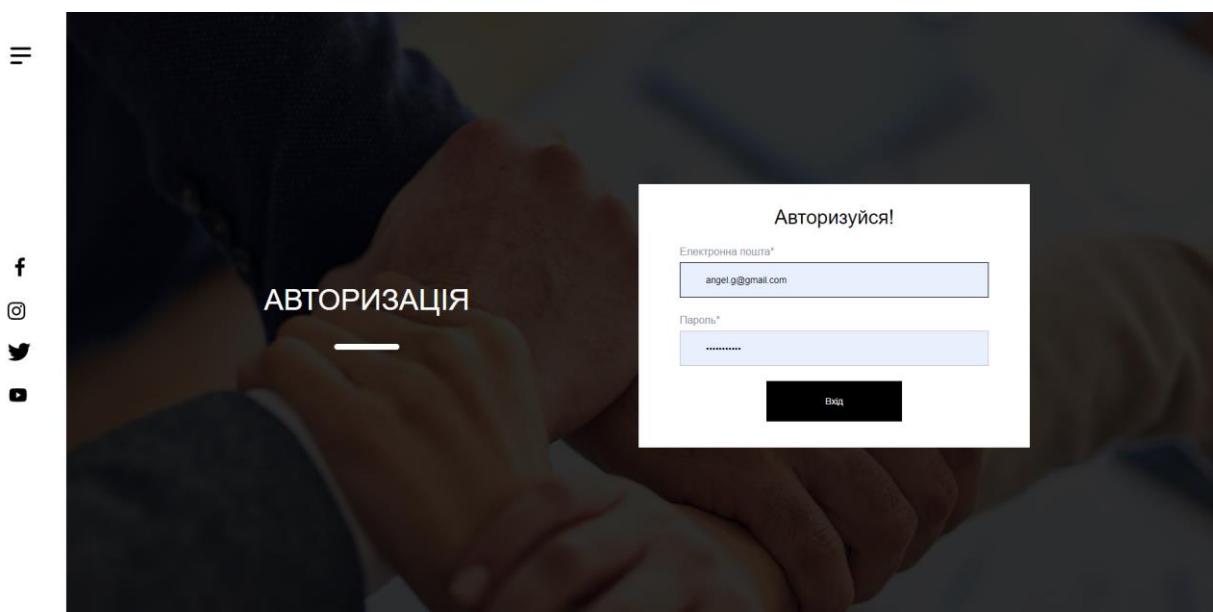


Рисунок 4.4 – Сторінка авторизації

Після успішної авторизації буде автоматичний перехід на головну сторінку на рисунку 4.5.

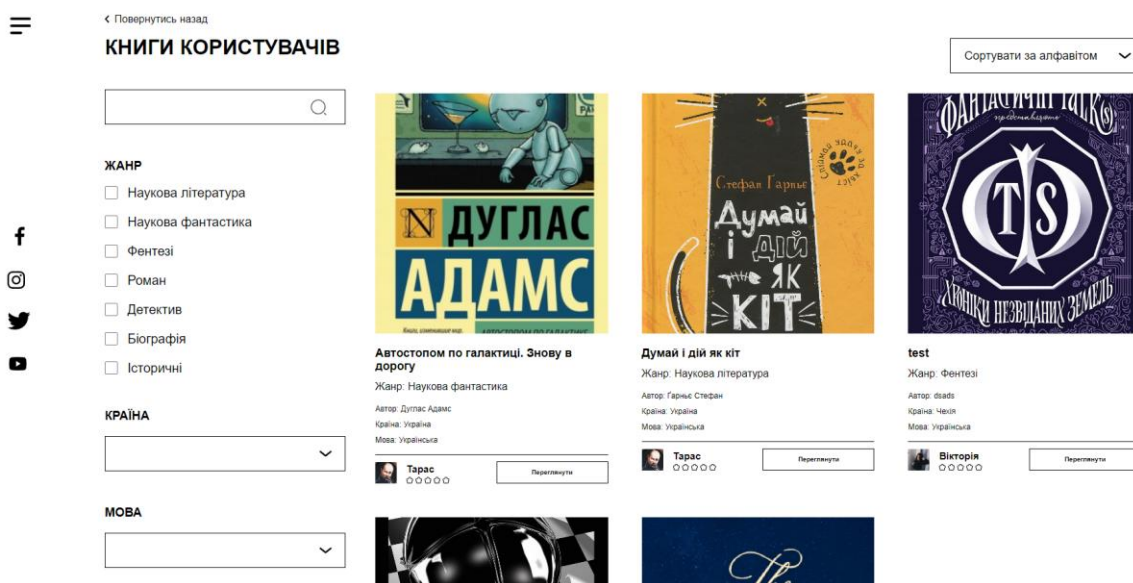


Рисунок 4.5 – Головна сторінка кабінету користувача

В першу чергу потрібно налаштувати особисту інформацію про користувача в особистій сторінці. Користувач може редагувати такі поля:

- Власне фото;
- Прізвище та ім'я;
- Електронну пошту;
- Мобільний номер телефону;
- Країну походження;
- Коротку інформацію про себе;
- Улюблена цитата;
- Улюблена книга;
- Змінити пароль з підтвердженням.

Мінімалістичний та комфортний стиль прослідковується і на цій сторінці. Всю інформацію можна переглянути без перегортання сторінки.

На рисунку 4.6 зображено сторінку редагування особистої інформації.

← Повернутись назад

Ім'я * Сurname *
 Email * Телефон
 Країна
 Інформація про вас
 Улюблена цитата
 Улюблена книга
 Новий пароль Повторіть пароль
 Зберегти

Рисунок 4.6 – Сторінка налаштування особистої інформації

Для легкої навігації використано вид бокової панелі, яка відкривається та надає не лише швидкий доступ до функціонала, а й короткий опис категорій на рисунку 4.7.

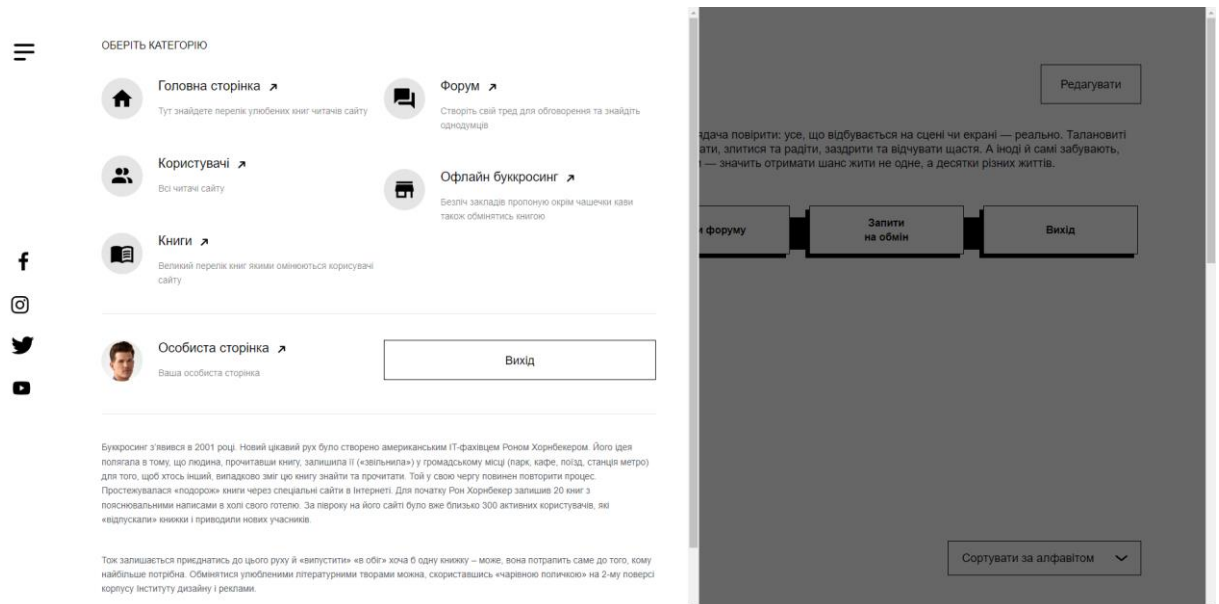


Рисунок 4.7 – Вид бокової панелі

Після внесених змін, в особисті дані, особиста сторінка почне відображати актуальну інформацію.

На рисунку 4.8 показана особиста сторінка користувача зі зміненою інформацією.

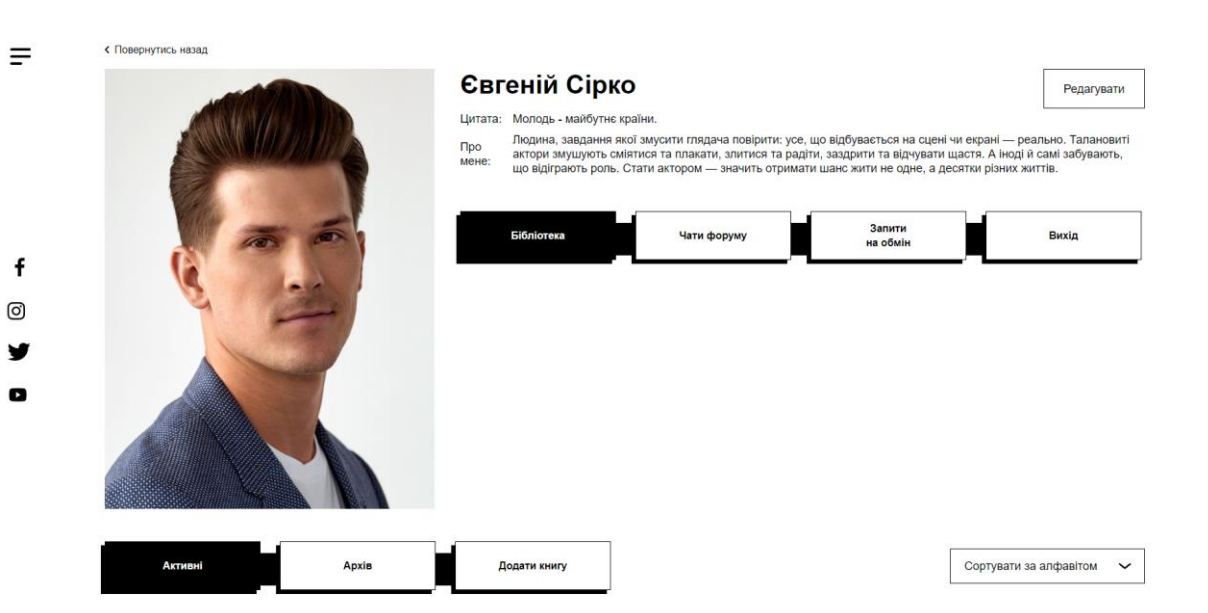


Рисунок 4.8 – Особиста сторінка користувача

Для створення сторінки своєї книги потрібно перейти на «Додати книгу» на рисунку 4.9.

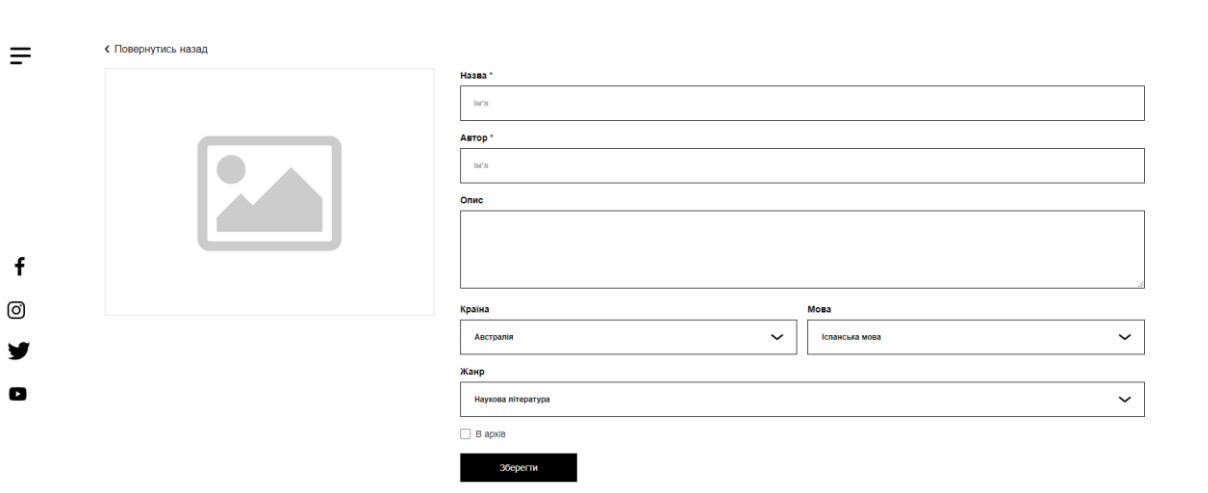
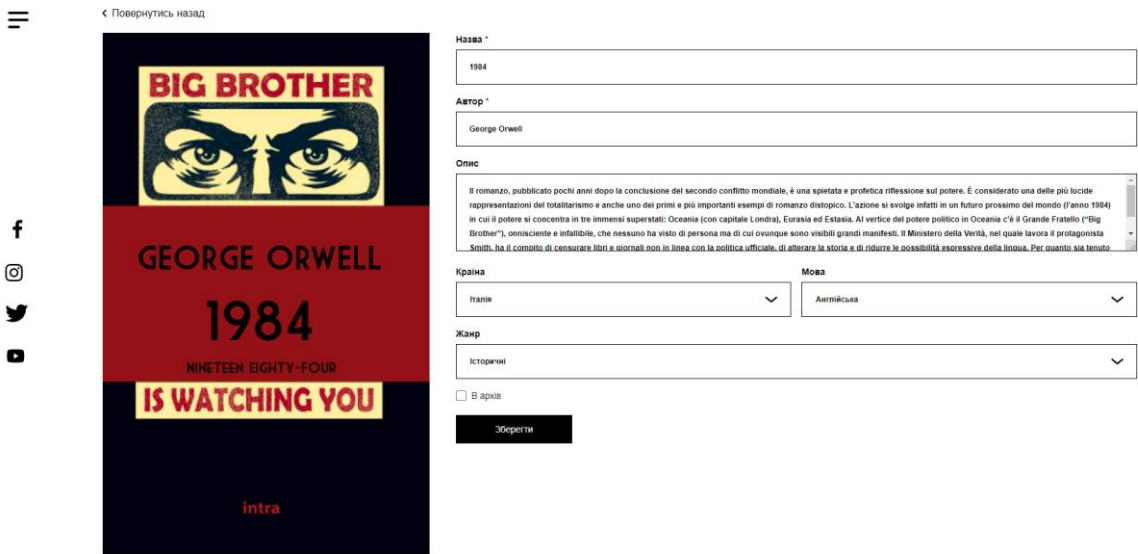


Рисунок 4.9 – Сторінка додавання книги

Для успішного додавання книги, потрібно заповнити всі поля та додати фото як на рисунку 4.10.



← Повернутись назад

BIG BROTHER

GEORGE ORWELL

1984

NINETEEN EIGHTY-FOUR

IS WATCHING YOU

intra

Назва *

1984

Автор *

George Orwell

Опис

Il romanzo, pubblicato pochi anni dopo la conclusione del secondo conflitto mondiale, è una spietata e profetica riflessione sul potere. È considerato una delle più lucide rappresentazioni del totalitarismo e anche uno dei primi e più importanti esempi di romanzo distopico. L'azione si svolge infatti in un futuro prossimo del mondo (l'anno 1984) in cui il potere si concentra in tre immensi superstati: Oceania (con capitale Londra), Eurasia ed Estasia. Al vertice del potere politico in Oceania c'è il Grande Fratello ("Big Brother"), onnisciente e infallibile, che nessuno ha visto di persona ma di cui ovunque sono visibili grandi manifesti. Il Ministero della Verità, nel quale lavora il protagonista Smith, ha il compito di censurare libri e giornali non in linea con la politica ufficiale, di alterare la storia e di ridurre le possibilità espressive della lingua. Per quanto sia tenuto

Країна

Italia

Мова

Англійська

Жанр

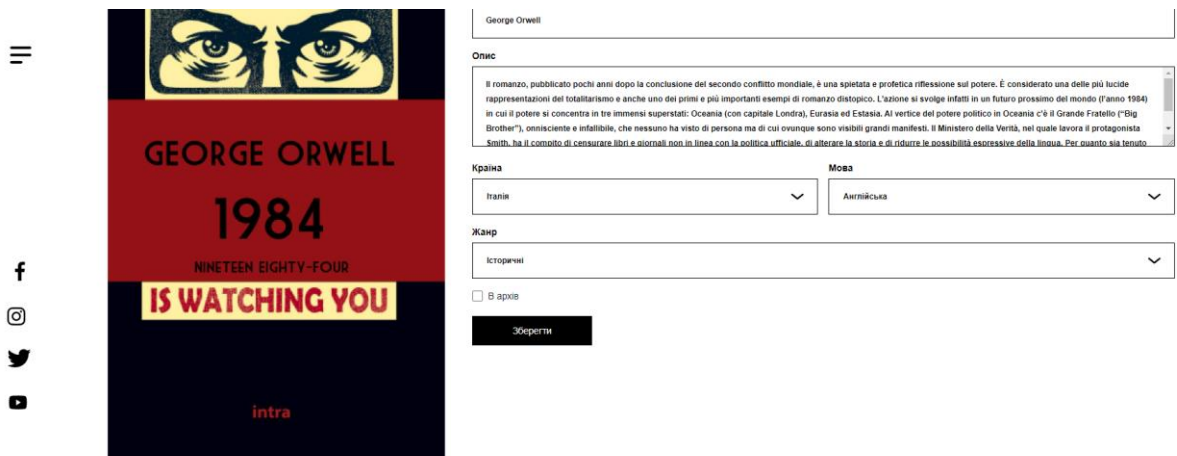
Історичні

В архів

Зберегти

Рисунок 4.10 – Приклад вибору напрямку при створенні

Якщо при створенні сторінки книги виникли питання, то їх можна задати адміністрації сайту. Потрібна кнопка знаходиться в нижній частині сторінки, як це вказано на рисунках 4.11-4.12.



George Orwell

Опис

Il romanzo, pubblicato pochi anni dopo la conclusione del secondo conflitto mondiale, è una spietata e profetica riflessione sul potere. È considerato una delle più lucide rappresentazioni del totalitarismo e anche uno dei primi e più importanti esempi di romanzo distopico. L'azione si svolge infatti in un futuro prossimo del mondo (l'anno 1984) in cui il potere si concentra in tre immensi superstati: Oceania (con capitale Londra), Eurasia ed Estasia. Al vertice del potere politico in Oceania c'è il Grande Fratello ("Big Brother"), onnisciente e infallibile, che nessuno ha visto di persona ma di cui ovunque sono visibili grandi manifesti. Il Ministero della Verità, nel quale lavora il protagonista Smith, ha il compito di censurare libri e giornali non in linea con la politica ufficiale, di alterare la storia e di ridurre le possibilità espressive della lingua. Per quanto sia tenuto

Країна

Italia

Мова

Англійська

Жанр

Історичні

В архів

Зберегти

LET'S GET IN TOUCH

Задати запитання

For business inquiry please send email to nanangprasetya.2000@gmail.com

Рисунок 4.11 – Можливість поставити питання адміністрації

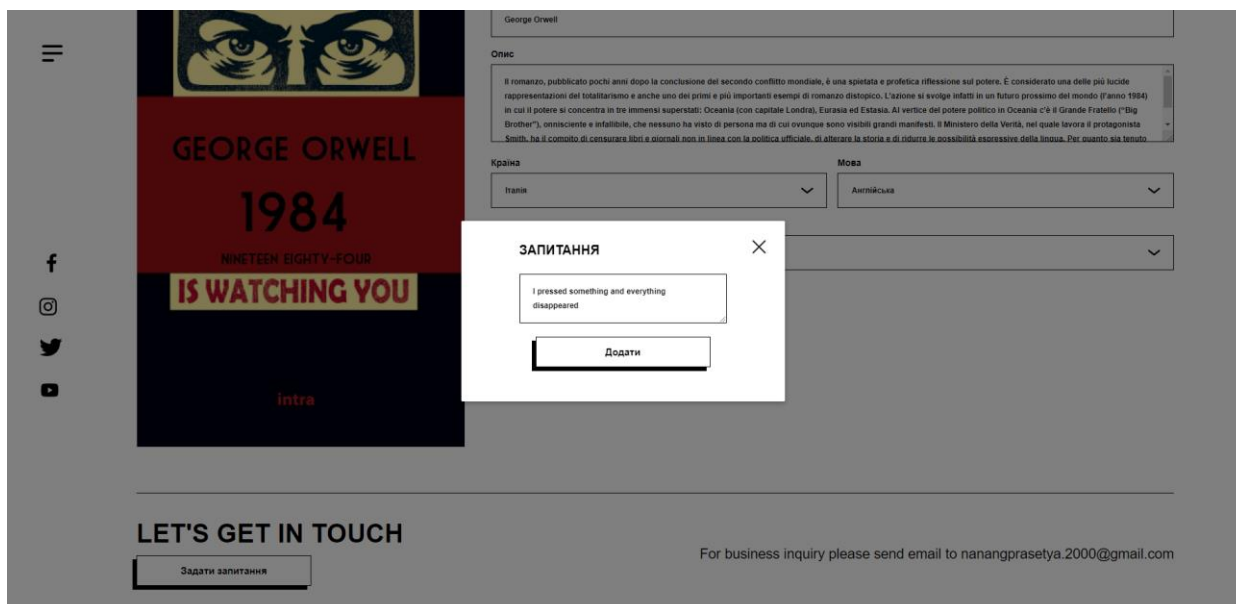


Рисунок 4.12 – Функція «Задати питання»

Після створення сторінки книги, для користувача надається можливість редагувати інформацію про книгу або переглянути коментарі від інших користувачів, як це показано на рисунках 4.13-4.14.

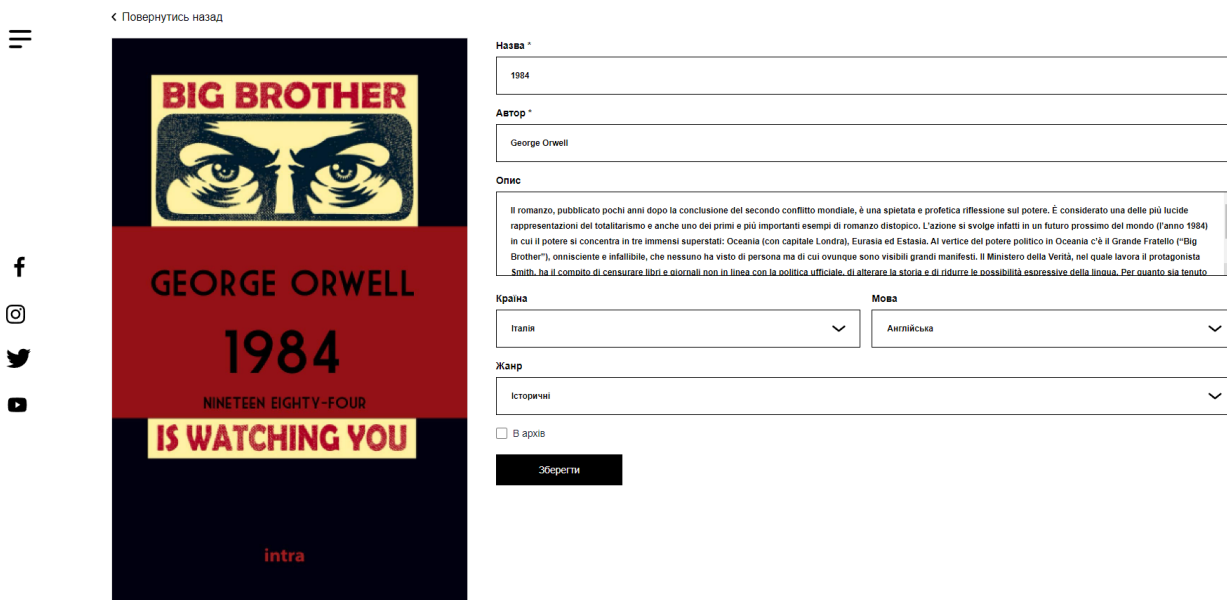


Рисунок 4.13 – Редагування інформації про книгу

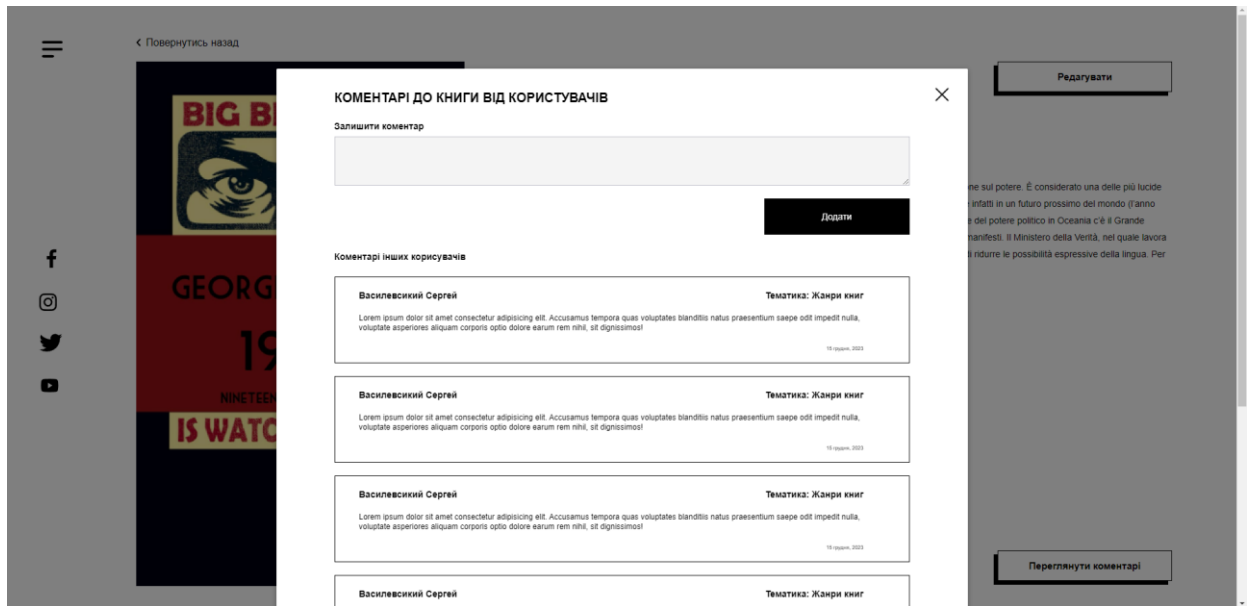


Рисунок 4.14 – Перегляд коментарів від інших користувачів

Серед функціоналу інформаційної системи реалізовано повноцінний форум, який відкривається лише для зареєстрованих користувачів на рисунку 4.15. Створення нової теми проходить на головній сторінці форуму як на рисунку 4.16. Відображення тем на форумі можна налаштувати шляхом сортування на рисунку 4.17.

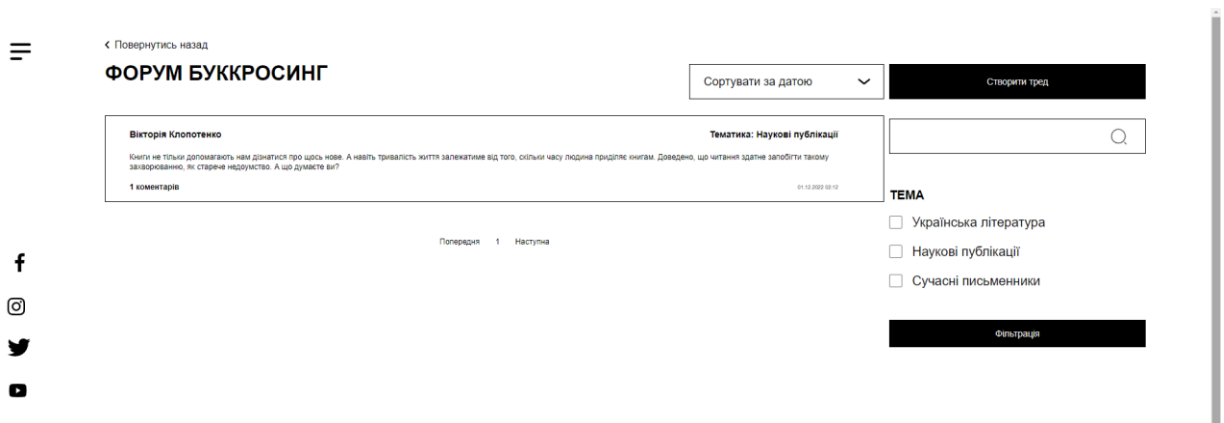


Рисунок 4.15 – Головна сторінка форуму

← Повернутись назад

ФОРУМ БУККРОСИНГ

Сортувати за датою

Створити тред

Заголовок: The Trial

Тема: Наукові публікації

Опис: A terrifying psychological trip into the life of one Joseph K, an ordinary man who wakes up one day to find himself accused of a crime he did not commit, a crime whose nature is never revealed to him. Once arrested, he is released but must report to court on a regular basis, an event that proves maddening, as nothing is ever resolved. As he grows more uncertain of his fate, his personal life, including work at a bank and his relations with his landlady and a young woman who lives next door, becomes increasingly unpredictable. As Joseph tries to gain control, he succeeds only in accelerating his own excruciating downward spiral.

Видалити Створити тред

ТЕМА

Українська література

Наукові публікації

Сучасні письменники

Фільтрація

Вікторія Клопотенко Тематика: Наукові публікації

Книги не тільки допомагають нам дізнатися про щось нове. А навіть тривалість життя залежить від того, скільки часу людина приділяє книгам. Доведено, що читання здатне згладити такуж заворуванню, як стареє недоумство. А що думаєте ви?

1 коментарів 01.10.2022 02:12

Попередня 1 Наступна

Рисунок 4.16 – Створення нової теми форуму

← Повернутись назад

ФОРУМ БУККРОСИНГ

Сортувати за датою

Сортувати за темою

Сортувати за коментарями

Створити тред

Вікторія Клопотенко

Книги не тільки допомагають нам дізнатися про щось нове. А навіть тривалість життя залежить від того, скільки часу людина приділяє книгам. Доведено, що читання здатне згладити такуж заворуванню, як стареє недоумство. А що думаєте ви?

1 коментарів 01.10.2022 02:12

Євгеній Сірко Тематика: Наукові публікації

A terrifying psychological trip into the life of one Joseph K, an ordinary man who wakes up one day to find himself accused of a crime he did not commit, a crime whose nature is never revealed to him. Once arrested, he is released but must report to court on a regular basis, an event that proves maddening, as nothing is ever resolved. As he grows more uncertain of his fate, his personal life, including work at a bank and his relations with his landlady and a young woman who lives next door, becomes increasingly unpredictable. As Joseph tries to gain control, he succeeds only in accelerating his own excruciating downward spiral.

0 коментарів 02.10.2022 12:12

Фільтрація

Попередня 1 Наступна

Рисунок 4.17 – Сорткування тем форуму

Тему на форумі можна прокоментувати, для цього потрібно натиснути на тему та в полі для коментаря написати текст на рисунку 4.18.

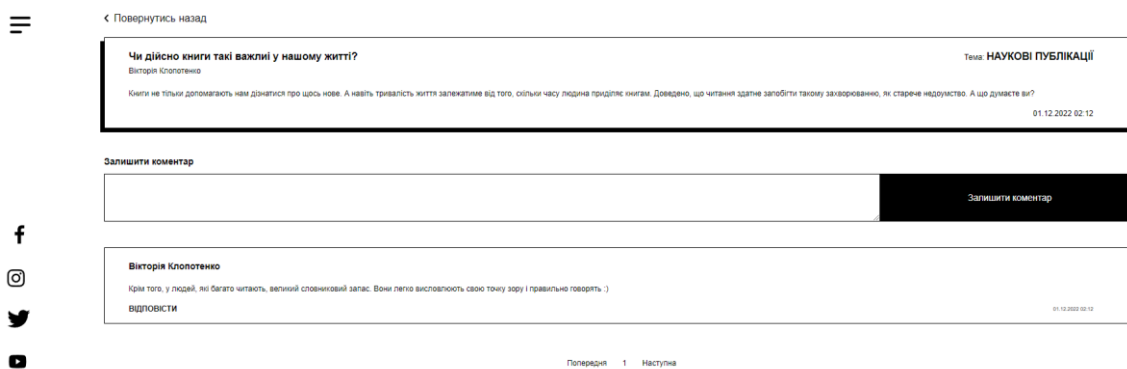


Рисунок 4.18 – Коментар до теми форуму

Коментар можна залишити не лише до теми форуму, а й відповіді на інший коментар, тим самим користувачам надається можливість вести дискусію на рисунку 4.19.



Рисунок 4.19 – Відповідь на коментар іншого користувача

Користувач може переглянути інших користувачів, налаштувати потрібні фільтри та перейти на особисті сторінки на рисунку 4.20.

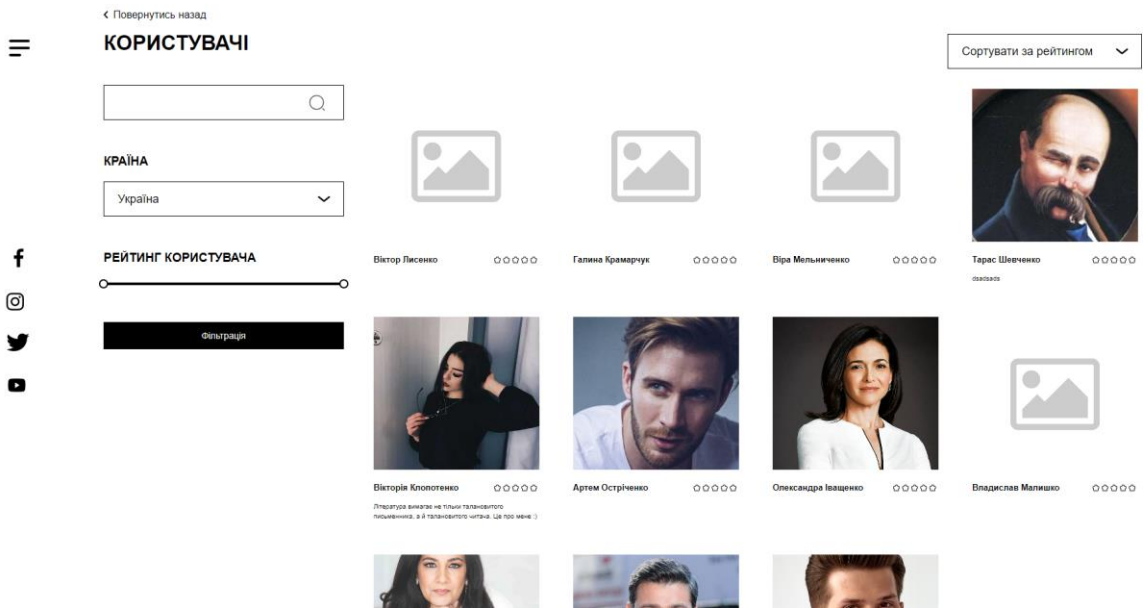


Рисунок 4.20 – Набір зареєстрованих користувачів

На сторінці «Книги користувачів» показано набір книг, які користувачі готові обміняти. Для комфорту, було реалізовано гнучкий інструментарій фільтрації та сортування витворів мистецтва на рисунках 4.21-4.22.

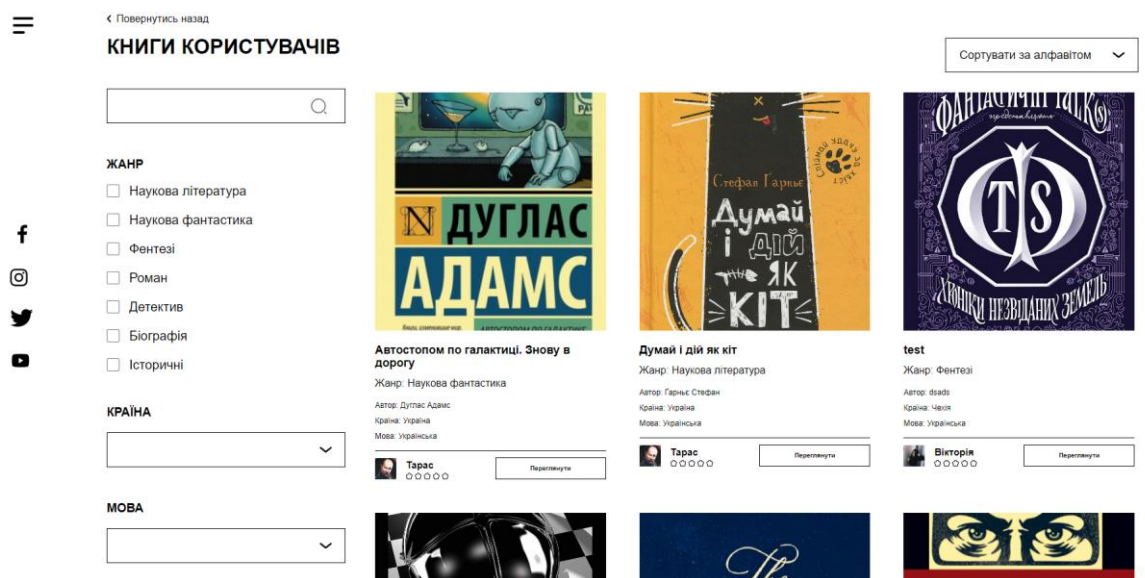


Рисунок 4.21 – Сторінка з набором книг

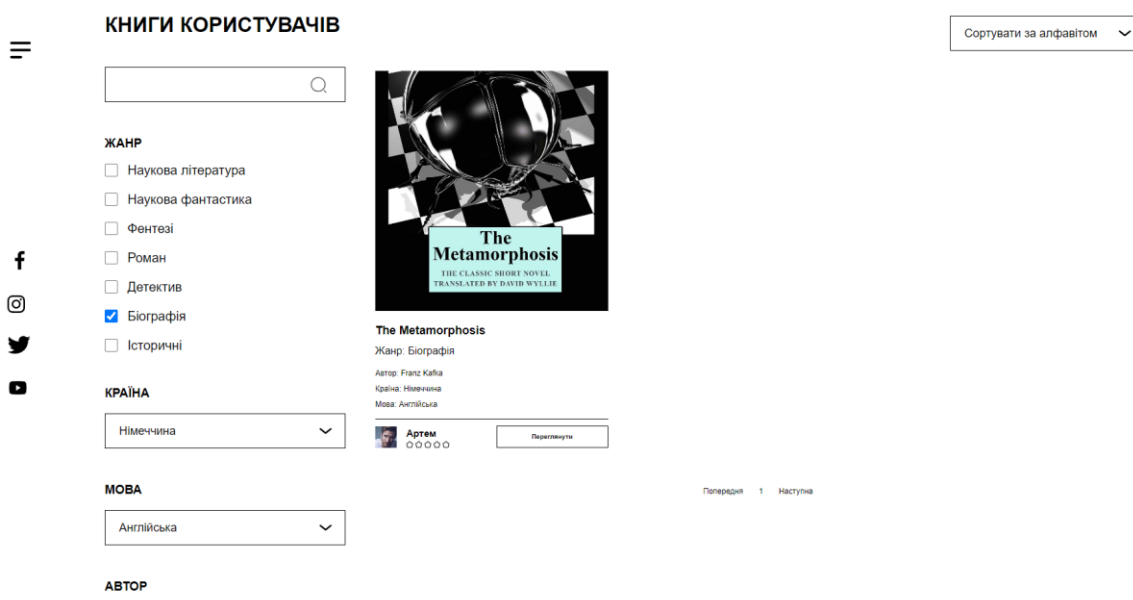


Рисунок 4.22 – Фільтрація та сортування книг

Для створення заявки на обмін книгами потрібно перейти на потрібну книгу через посилання з набором книг, або через особисту сторінку користувача який володіє цією книгою (рис 4.23-4.24).

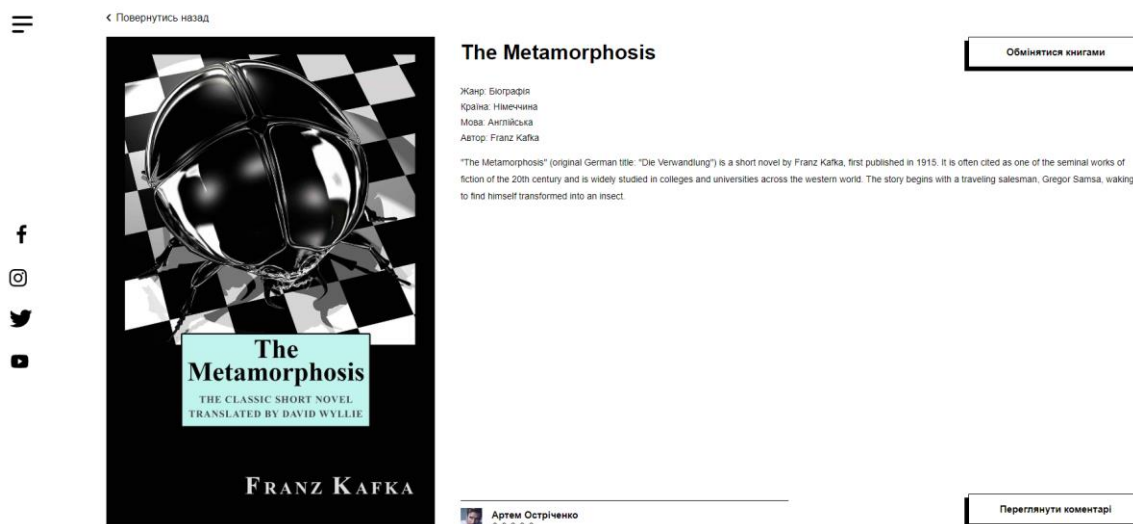


Рисунок 4.23 – Сторінка книги іншого користувача

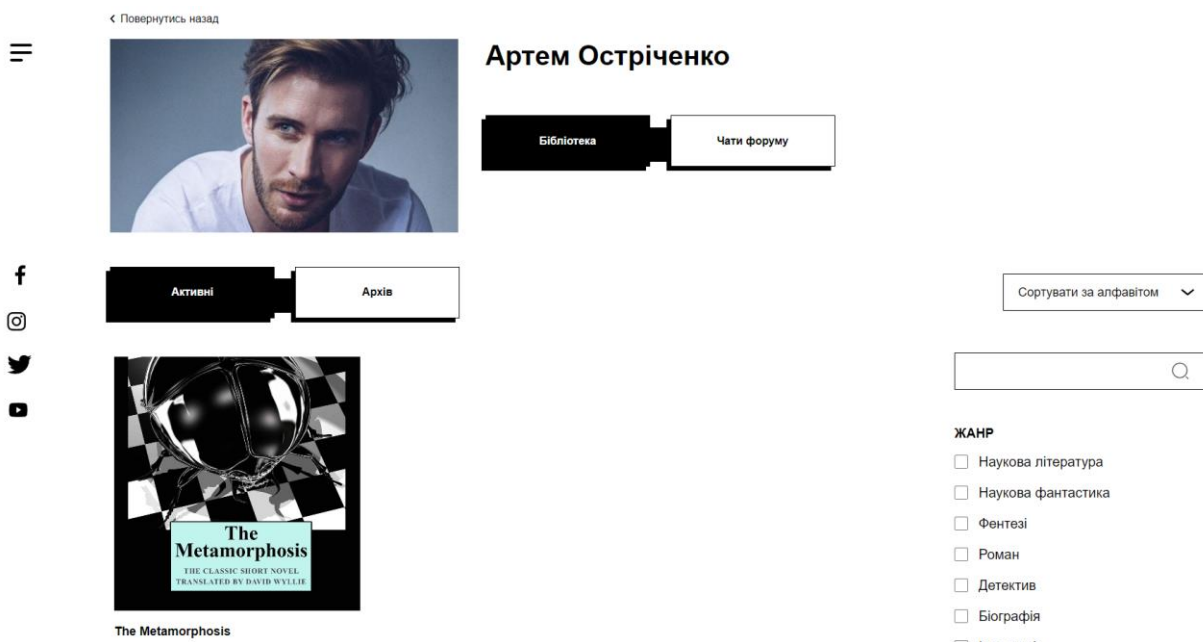


Рисунок 4.24 – Особиста сторінка користувача з потрібною книгою

Якщо користувач не авторизований, то йому також доступне меню інформаційної системи, але недоступний перехід на особисту сторінку та обмін книгами (рис. 4.25).

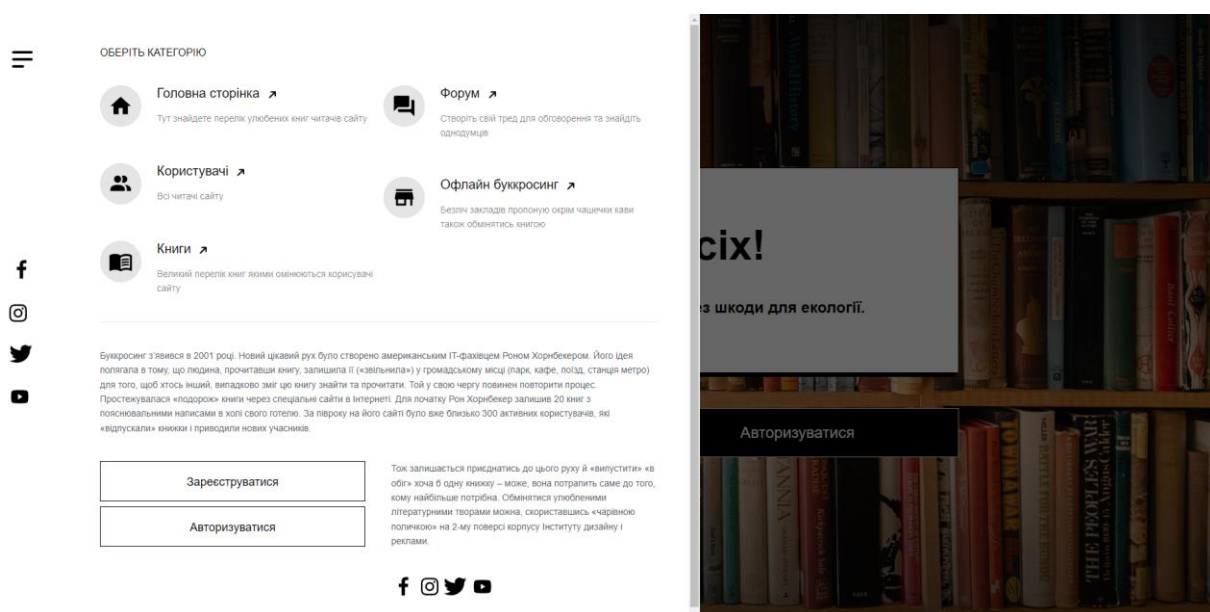


Рисунок 4.25 – Меню інформаційної системи неавторизованного користувача

Для створення успішної заявки на обмін, система пропонує вибрати свою книгу для обміну та обов'язково заповнити поле з текстом (рис. 4.26).

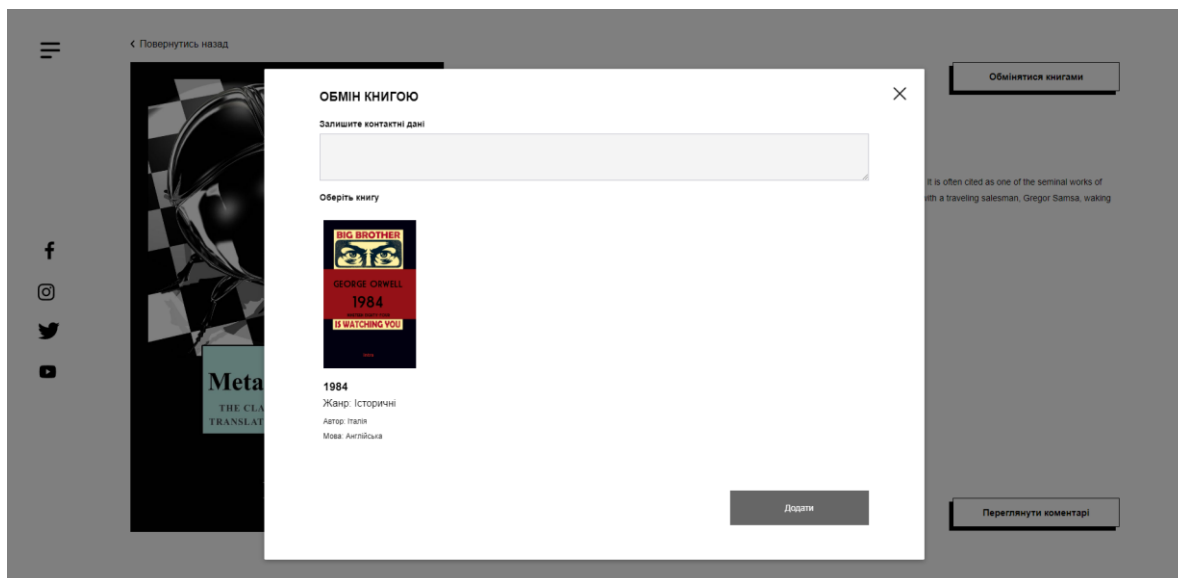


Рисунок 4.26 – Оплата послуги

Після створення пропозиції в іншого користувача з'явиться запит на обмін, яку можна прийняти або відхилити (рис. 4.27).

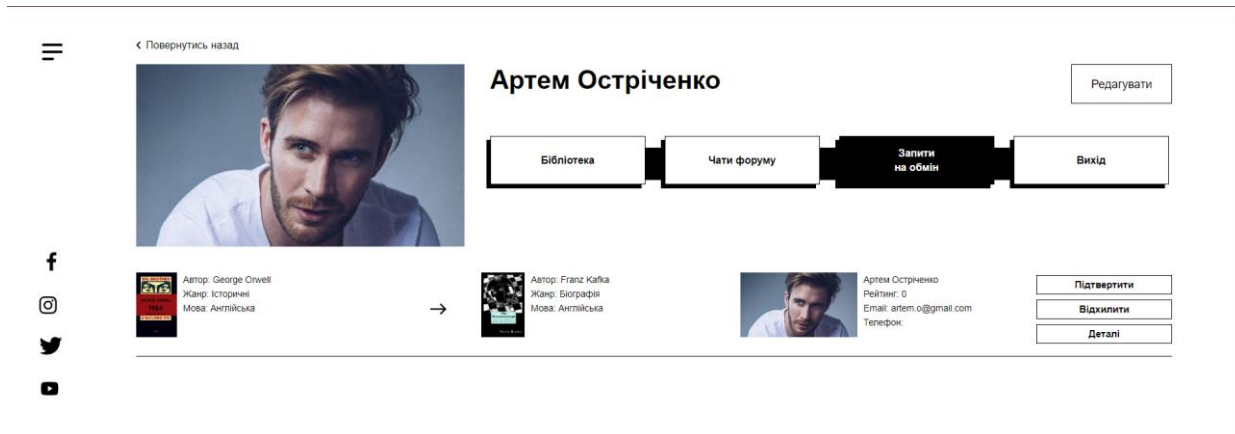


Рисунок 4.27 – Запит на обмін

Після підтвердження запиту, відкривається сторінка обміну (рис. 4.28-4.29).

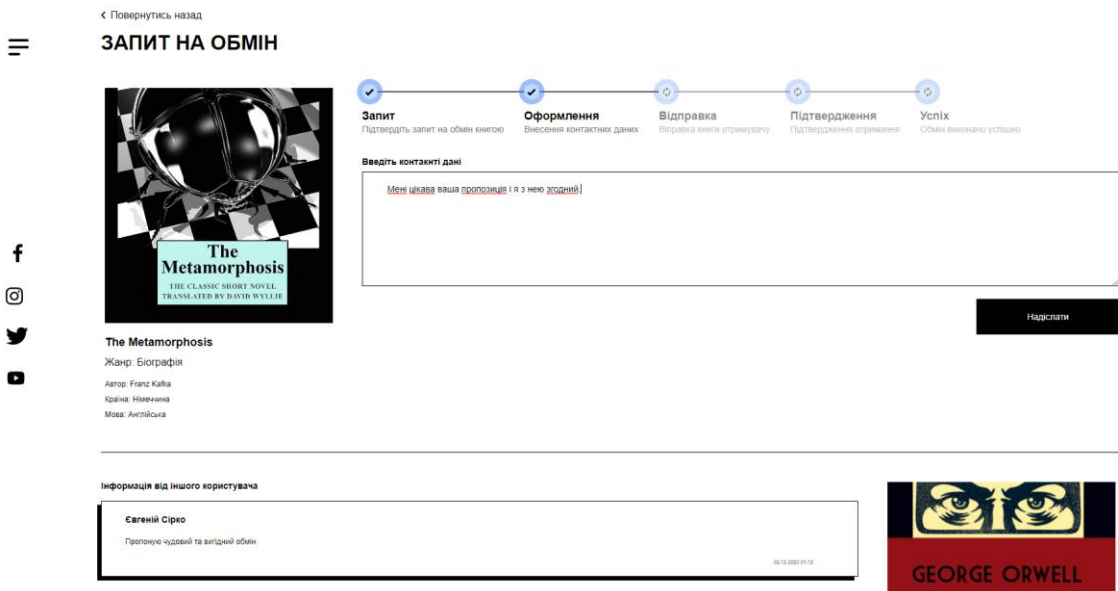


Рисунок 4.28 – Сторінка запиту на обмін

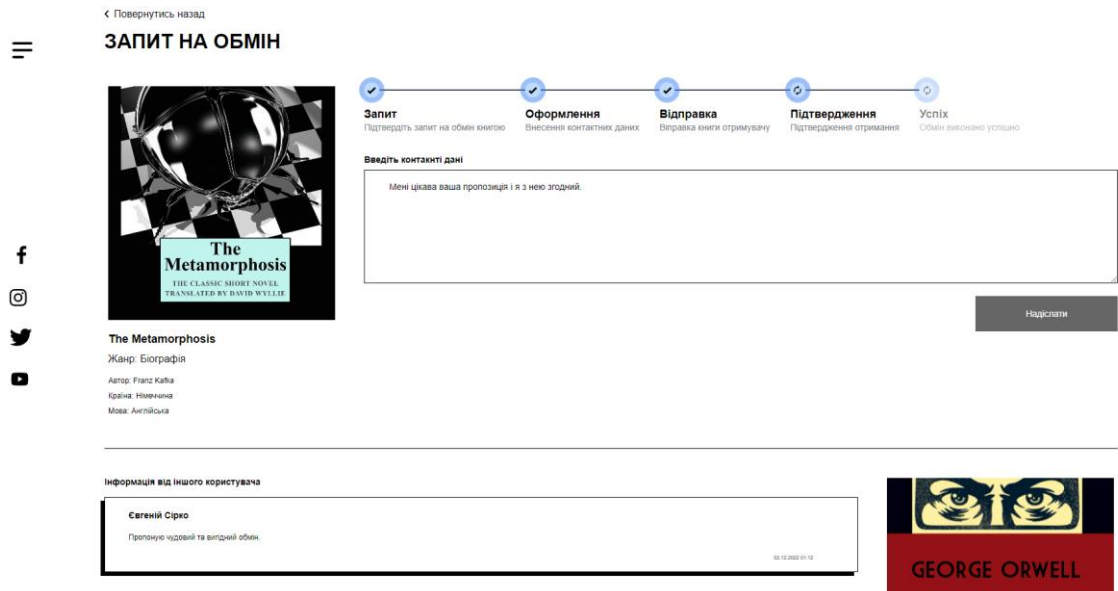


Рисунок 4.29 – Сторінка успішного обміну книгами

Після успішної операції обміну, для учасників обміну відкривається можливість оцінити операцію та підняти рейтинг один одному (рис. 4.30).

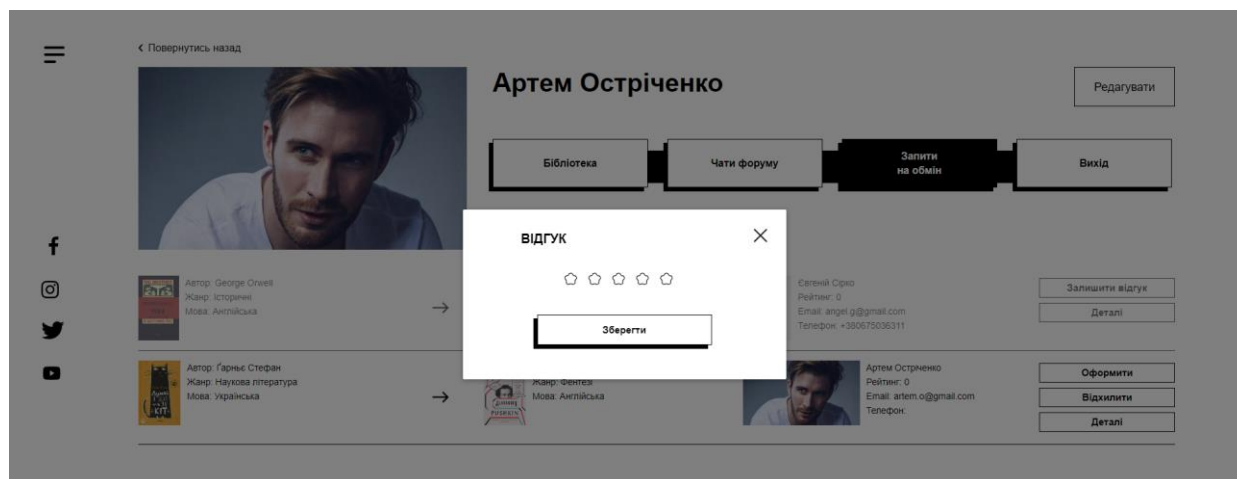


Рисунок 4.30 – Оцінка взаємодії

Для комфорту користувачів та незареєстрованих відвідувачів, офлайн буккросинг доступний для перегляду всім відвідувачам, де всі хто бажає можуть переглянути заклади з можливістю обміну книгами. (рис. 4.31).

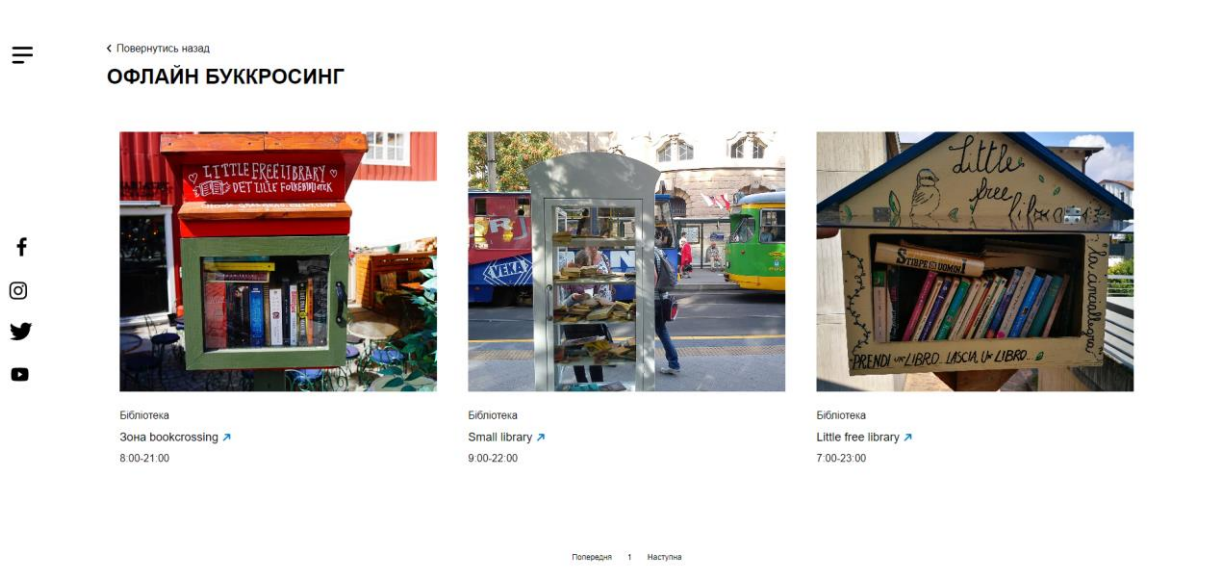


Рисунок 4.31 – Офлайн буккросинг

4.3 Адміністрування інформаційної системи

Для адміністратора виділено спеціальне посилання та перше, що потрібно йому зробити – авторизуватися (рис. 4.32).

Авторизація

Email

name@company.com

Пароль

.....

Вхід

Рисунок 4.32 – Посилання для адміністратора

Адміністратор може з головної сторінки переглядати головну інформацію сайту, редагувати посилання на соціальні мережі та переглядати останні обміни.

На рисунку 4.33 показано головну сторінку з переліченим вище функціоналом.

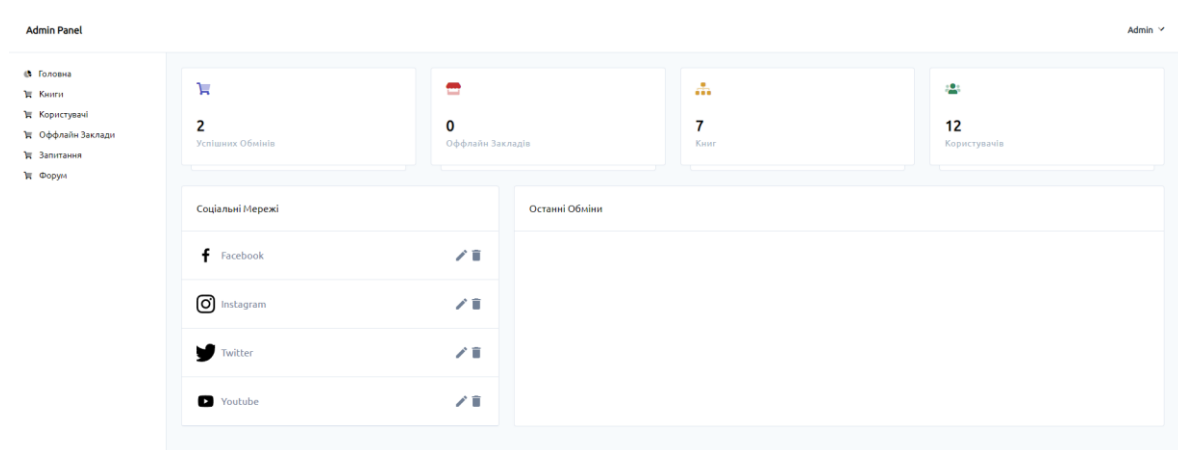


Рисунок 4.33 – Головна сторінка панелі адміністратора

Адміністратор через сторінку «Книги» може видаляти книги зі сторінок користувачів (рис. 4.34).

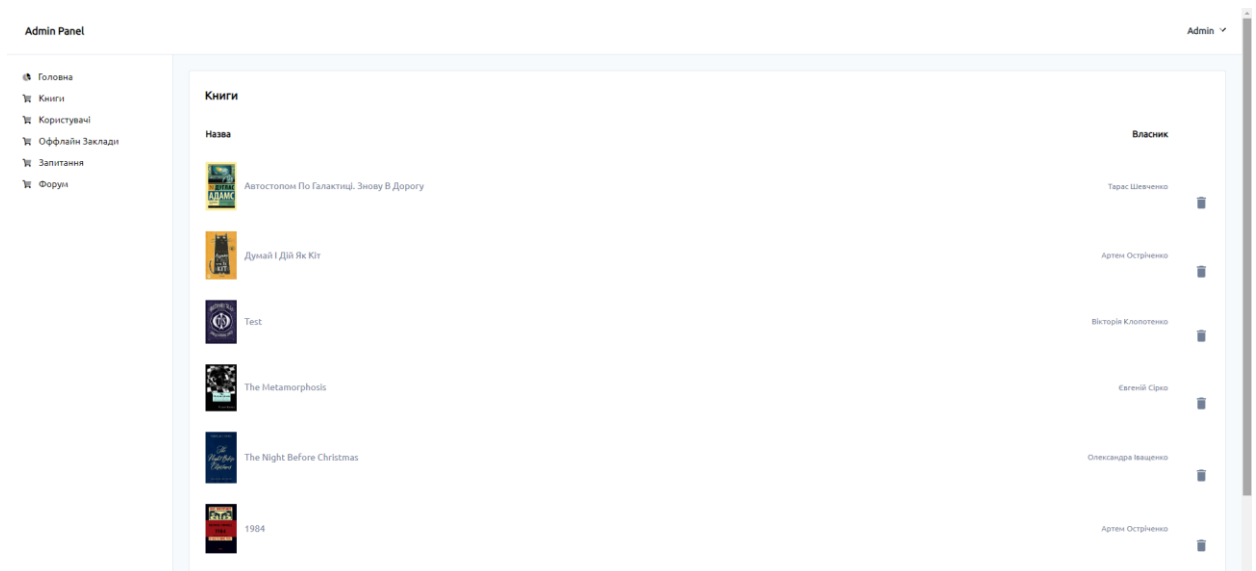


Рисунок 4.34 – Сторінка «Книги»

На сторінці «Користувачі» є можливість блокувати або розблокувати користувачів (рис. 4.35).

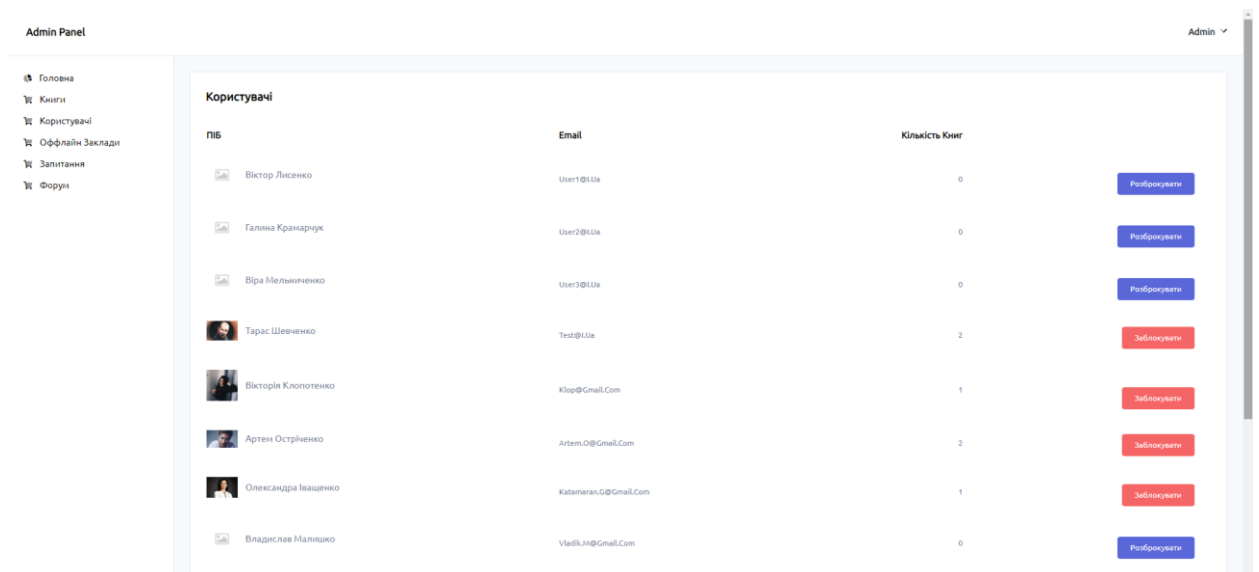


Рисунок 4.35 – Сторінка «Користувачі»

Заклади для офлайн буккросингу може додавати та редагувати лише адміністратор через власну панель на пристосованій для цього сторінці (рис. 4.36).

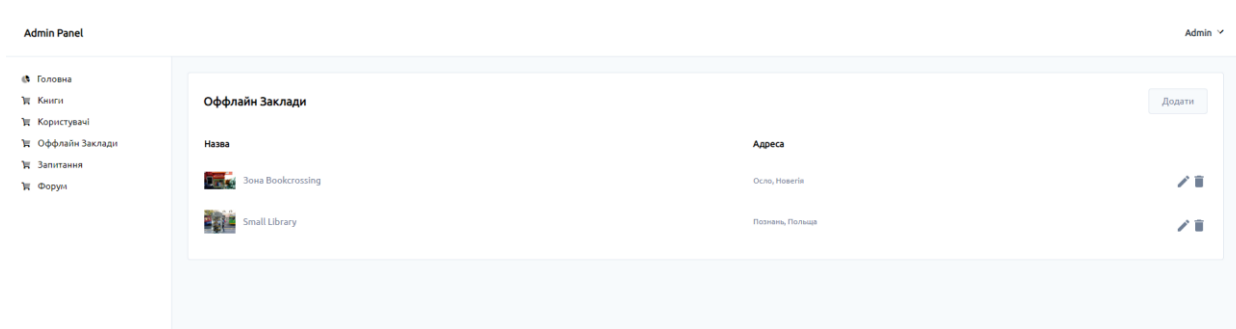


Рисунок 4.36 – Сторінка «Офлайн буккросинг»

Для створення сторінки закладу для офлайн буккросингу, потрібно перейти на спеціальну сторінку, заповнити всі поля та натиснути «Зберегти» (рис. 4.37).

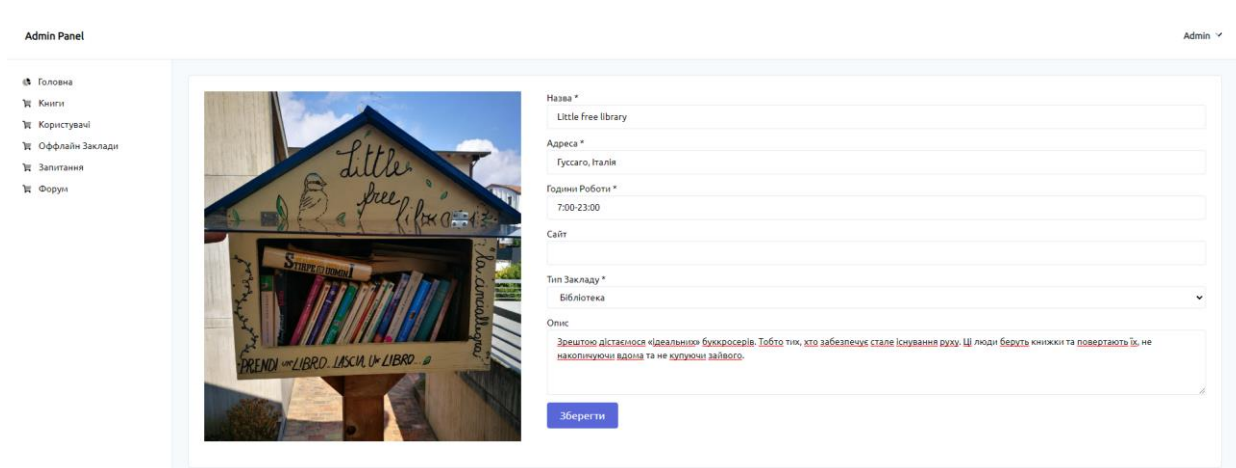


Рисунок 4.37 – Створення сторінки закладу офлайн буккросингу

На сторінці «Запитання» для адміністратора відкривається можливість переглядати, відповідати та видаляти питання, які були задані через спеціальну форму (рис. 4.38).

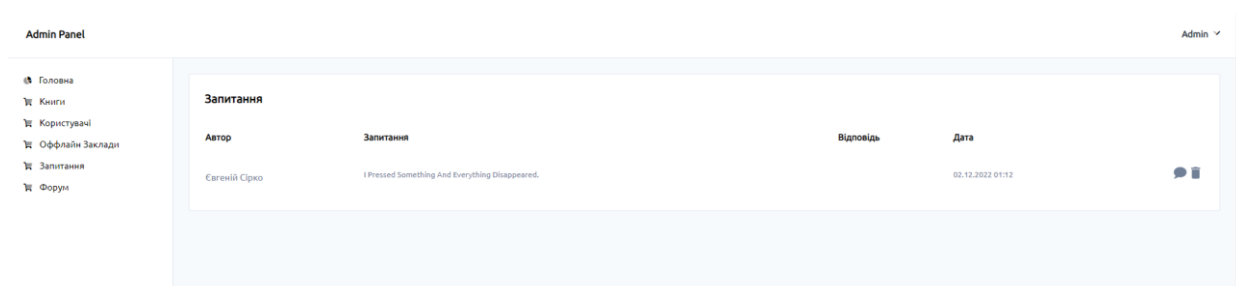


Рисунок 4.38 – Сторінка «Запитання»

Сторінка на рисунку 4.39 відповідає за перегляд та контроль від неналежної поведінки користувачів. Якщо створена тема чи коментар порушує правила корпоративної ввічливості, то їх можна видалити.

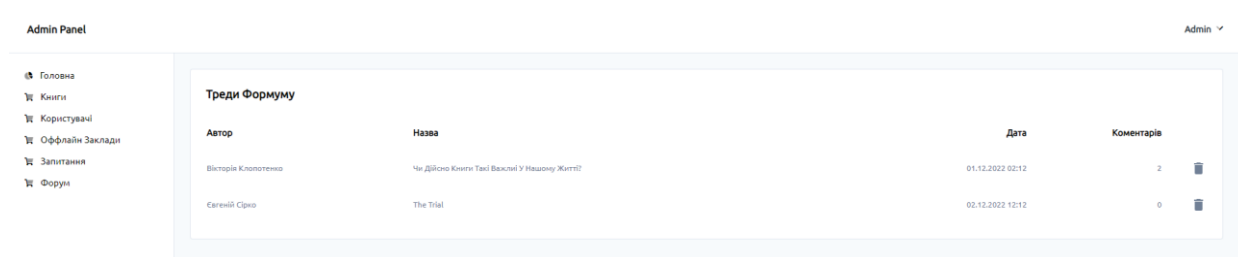


Рисунок 4.39 – Сторінка «Форум»

Сторінка на рисунку 4.39 показано адаптив інформаційної системи підтримки буккросингу.

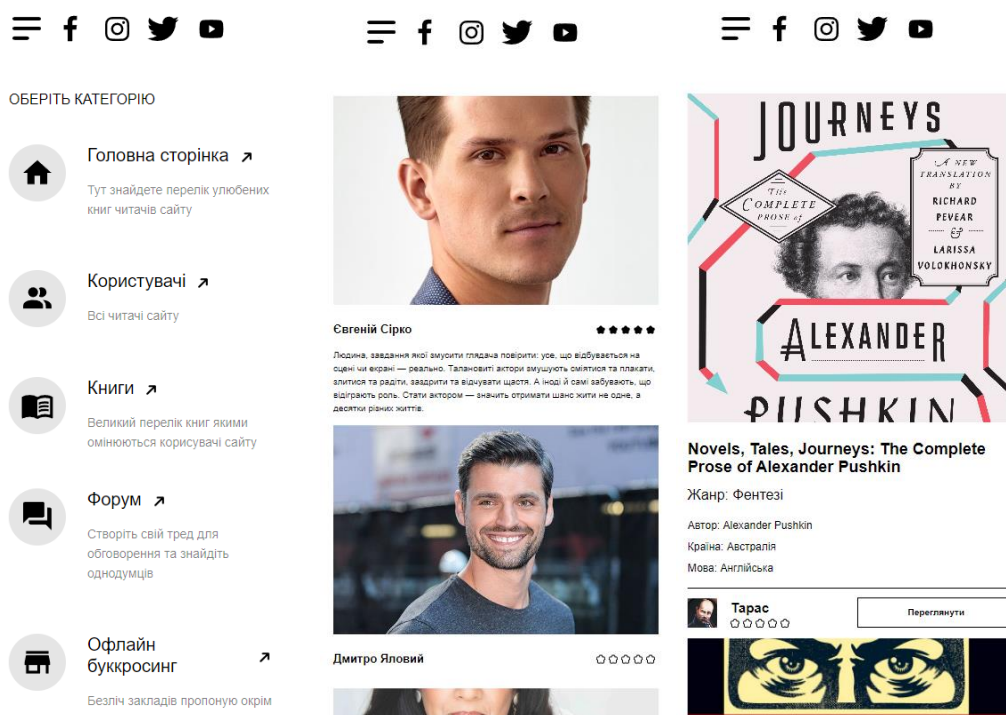


Рисунок 4.40 – Адаптив інформаційної системи

ВИСНОВКИ

При виконанні кваліфікаційної роботи магістра було досліджено та проаналізовано всі головні аспекти інформаційних потоків, які використовуються у тематиці буккросингу. На базі аналогів інформаційних систем були виявлені недоліки та переваги сайтів.

Було проведено пошук та аналіз інформації згідно з тематикою кваліфікаційної роботи. Також я ознайомився з реалізаціями схожих задач, виконав виділення їх переваг та недоліків.

Створено структуру задач по створенню продукту, організаційну структуру, матрицю відповідальності, часові межі представлені у вигляді часової шкали та діаграми Ганта, проаналізовано можливі ризики та шляхи їх попередження або усунення (Додаток А).

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Everything You Should Know About Paper And Ecology [Електронний ресурс] – Режим доступу до ресурсу: <https://contemporarylynx.co.uk/everything-you-should-know-about-paper-and-ecology> (дата звернення: 18.09.2022);
2. Tons of paper produced [Електронний ресурс] – Режим доступу до ресурсу: <https://www.theworldcounts.com/challenges/consumption/other-products/environmental-impact-of-paper> (дата звернення: 18.09.2022);
3. Why Do Print Books Still Dominate the Reading Marketplace in 2022? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.scanningpens.blog/news/2022/4/25/why-do-print-books-still-dominate-the-reading-marketplace-in-2022> (дата звернення: 18.09.2022);
4. Print book market has its best start to the year in a decade [Електронний ресурс] – Режим доступу до ресурсу: <https://www.thebookseller.com/bestsellers/print-book-market-has-its-best-start-to-the-year-in-a-decade> (дата звернення: 18.09.2022);
5. What is BookCrossing? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.languagehumanities.org/what-is-bookcrossing.htm> (дата звернення: 02.10.2022);
6. D. J. (Ohio S. U. Staley, Futures Thinking for Academic Librarians: Scenarios for the Future of the Book. Association of College & Research Libraries, 36 (2012)
7. Modernism’s Print Cultures (Bloomsbury Academic, 2015)
8. The Bookcrossing space is enriched with new books. Come and have a look! [Електронний ресурс] – Режим доступу до ресурсу: <https://cla.unipd.it/en/bookcrossing/> (дата звернення: 01.10.2022);
9. Y. Takemon, Ecology and Civil Engineering as a science of habitat. Ecol. Civil Eng. 10(1), 41-46, 2007. Ecology and Civil Engineering. 10, 41–46 (2007).
10. Insect Bioecology and Nutrition for Integrated Pest Management (CRC Press, 2012).
11. J. A. Flannery, K. M. Smith, Eco-Library Design (Springer Netherlands, 2014).

12. S. Drosopoulos, M. F. Claridge, *Insect sounds and communication: Physiology, behaviour, ecology, and evolution* (CRC Press, 2005).

13. *Role of Print Media in Coverage of Environment Issues* [Электронный ресурс] – Режим доступа до ресурсу: <https://edubirdie.com/examples/role-of-print-media-in-coverage-of-environment-issues/> (дата звернення: 02.10.2022);

14. *ENVIRONMENTAL IMPACT OF PRINTING* [Электронный ресурс] – Режим доступа до ресурсу: <https://www.davidbrimble.com/environmental-impact-of-printing/> (дата звернення: 02.10.2022);

15. *The Impact of Printing on The Environment* [Электронный ресурс] – Режим доступа до ресурсу: <https://www.capitalmds.com/the-impact-of-printing-on-the-environment/> (дата звернення: 02.10.2022);

16. Сайт аналог «Bookcrossing-ua.at.ua» [Электронный ресурс] – Режим доступа до ресурсу: <https://bookcrossing-ua.at.ua> (дата звернення: 02.10.2022);

17. Сайт аналог «Bookcrossing.com» [Электронный ресурс] – Режим доступа до ресурсу: <https://www.bookcrossing.com/bookmaps> (дата звернення: 02.10.2022);

18. Facebook група «Bookcrossing Lviv» [Электронный ресурс] – Режим доступа до ресурсу: <https://www.facebook.com/groups/657575231011490> (дата звернення: 02.10.2022);

19. Martin the Robert: *Clean Code: A Handbook of Agile Software Craftsmanship* / Pearson, 2017. – 1214 с.

20. Martin Fowler: *Refactoring: Improving the Design of Existing Code* / Addison-Wesley Professional, 2018. – 448 с.

21. V. Anton Spraul: *Think Like a Programmer: An Introduction to Creative Problem Solving* / No Starch Press, 2018. – 256 с.

22. Merih Taze: *Engineers Survival Guide: Advice, tactics, and tricks After a decade of working at Facebook, Snapchat, and Microsoft* Paperback / Merih Taze, 2021. – 245 с.

23. Sam Newman, Theodore O'Brien: *Building Microservices: Designing Fine-Grained Systems* Audible Logo Audible Audiobook – Unabridged / Upfront Books, 2022. – 534 с.

24. Nicole Forsgren PhD, Jez Humble, Gene Kim: Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations / IT Revolution Press, 2018. – 289 c.
25. Camille Fournier: The Manager's Path: A Guide for Tech Leaders Navigating Growth and Change / O'Reilly Media, 2019. – 241 c.
26. Eric Freeman, Elisabeth Robson: Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software / O'Reilly Media, 2021. – 669 c.
27. Scott Rogers: Level Ups! The Guide to Great Video Game Design / Wiley, 2020. – 560 c.
28. Will Larson, Shaula Evans: Staff Engineer: Leadership Beyond the Management Track Audible Logo Audible / Will Larson, 2021. – 112 c.
29. Kathy Sierra, Trisha Gee, Bert Bates: Head First Java: A Brain-Friendly Guide / O'Reilly Media, 2022. – 752 c.
30. Adam Bellemare: Building Event-Driven Microservices: Leveraging Organizational Data at Scale / O'Reilly Media, 2020. – 324 c.

ДОДАТОК А

ПЛАНУВАННЯ РОБІТ для розробки кваліфікаційної роботи магістра «Інформаційні технології проектування»

A.1 ІДЕНТИФІКАЦІЯ МЕТИ ІТ-ПРОЕКТУ

Головна мета проекту – розробка зручного інструментарію по обміну книгами на території України.

Розробимо таблицю та деталізацію за допомогою методу SMART (табл.А.1).

Таблиця А.1 – Деталізація мети методом SMART

Specific (конкретна)	Розробка інформаційної системи, за допомогою якої можна буде виконувати поставлені цілі та задачі, та виконувати обмін книгами.
Measurable (вимірювана)	Прогрес та якість буде вимірюватися в кількості користувачах та кількості позитивних відгуків.
Achievable (досяжна)	Данна інформаційна система буде розроблятися з використанням HTML, CSS, JavaScript, Vue.js та PHP, MySQL, Laravel.
Relevant (реалістична)	Для виконання проекту є всі необхідні засоби та знання.
Time-framed (обмежена у часі)	Для проекту визначений дедлайн та побудована Діаграма Ганта для відстеження часу виконання.

A.2 ПЛАНУВАННЯ ЗМІСТУ СТРУКТУРИ РОБІТ ІНФОРМАЦІЙНОЇ СИСТЕМИ

WBS - це розбиття проекту на конкретні результати, які повинні бути досягнуті для досягнення цілей проекту. Як правило, на верхньому рівні вказується сам проект, під ним (на першому рівні) - основні результати, кожен з яких, в свою чергу, деталізується, тобто наступний рівень завжди менше попереднього за обсягом робіт і, як правило, включає 2 і більше пакетів робіт. При цьому в різних гілках WBS може бути різна кількість рівнів в залежності від потрібного ступеня деталізації.

На рис.А.1 приведена WBS-структура проекту.



Рисунок А.1 – WBS-структура

Після побудови WBS розробляють організаційну структуру виконавців. OBS-структура проекту – організаційна структура виконавців проекту. Визначається за переліком пакетів робіт нижнього рівня кожної гілки WBS-структури.

Організаційна структура проекту є ієрархічною структурою управління проектом і показує відносини між учасниками проекту. Організаційна структура

проекту створюється на рівні підприємства. Її елементи призначаються на рівні EPS, проекти, пакети робіт структури декомпозиції робіт (WBS).

На рис.А.2 приведена OBS-структура проекту.

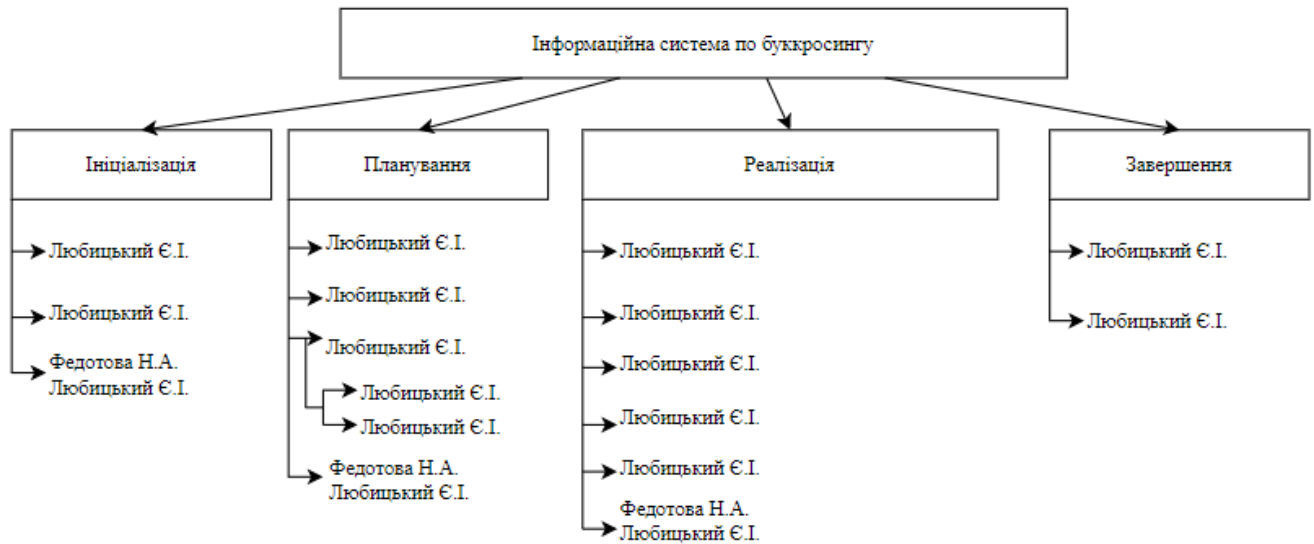


Рисунок А.2 – OBS-структура

А.3 ПОБУДОВА КАЛЕНДАРНОГО ГРАФІКУ ВИКОНАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

На підставі OBS та WBS структур будується матриця відповідальності проекту, яка закріплює за кожною роботою свого виконавця. Використовується для контролю відповідності розподілу ролей цілям проекту.

В таблиці А.2 приведена матриця відповідальності проекту.

Таблиця А.2 – Матриця відповідальності

Назва етапу		Студент	Викладач
1	Ініціалізація	+	+
1.1	Ознайомлення з предметної областю проекту	+	-
1.2	Аналіз конкурентів	+	-
1.3	Ідентифікація ідей	+	+
2	Планування	+	-
2.1	Підготовка матеріалу	+	-
2.2	Вивчення вимог проекту	+	-
2.3	Розробка календарного плану	+	-
2.4	Планування ризиків	+	-
3	Реалізація	+	+
3.1	Розробка прототипу	+	-
3.2	Підготовка макету дизайну сайту	+	-
3.3	Верстка	+	-
3.4	Робота над програмною частиною	+	-
3.5	Наповнення сайту контентом	+	-
3.6	Тестування	+	+
4	Завершення	+	-

Назва етапу	Студент	Викладач
4.1 Підготовка документації	+	-
4.1 Перенесення сайту на сервер	+	-

Діаграма Ганта – це горизонтальна діаграма з тимчасовою шкалою, яка використовується для ілюстрації плану робіт за проектом з прив'язкою до часу.

Дана діаграма проекту зображена на рисунках А.3 та А.4

	Длительнс ▾	Название задачи ▾
1	57 дней	Інформаційна система для тестування та підбору кандидатів на вакантні місця
2	5 дней	1 Ініціалізація
3	2 дней	1.1 Ознайомлення з предметної областю проекту
4	2 дней	1.2 Аналіз конкурентів
5	1 день	1.3 Ідентифікація ідей
6	7 дней	2. Планування
7	2 дней	2.1. Підготовка матеріалу
8	2 дней	2.2. Вивчення вимог проекту
9	1 день	2.3. Планування ризиків
10	2 дней	2.4. Розробка календарного плану
11	1 день	2.4.1. Діаграма Ганта
12	1 день	2.4.2. Мережевий графік
13	45 дней	3. Реалізація
14	5 дней	3.1. Розробка прототипу
15	3 дней	3.2. Підготовка макету дизайну сайту
16	15 дней	3.3. Верстка
17	15 дней	3.4. Робота над програмною частиною
18	2 дней	3.5. Наповнення сайту контентом
19	5 дней	3.6. Тестування
20	2 дней	4. Завершення
21	1 день	4.1. Підготовка документації
22	1 день	4.2. Перенесення сайту на сервер

Рисунок А.3 – Діаграма Ганта (інформація)

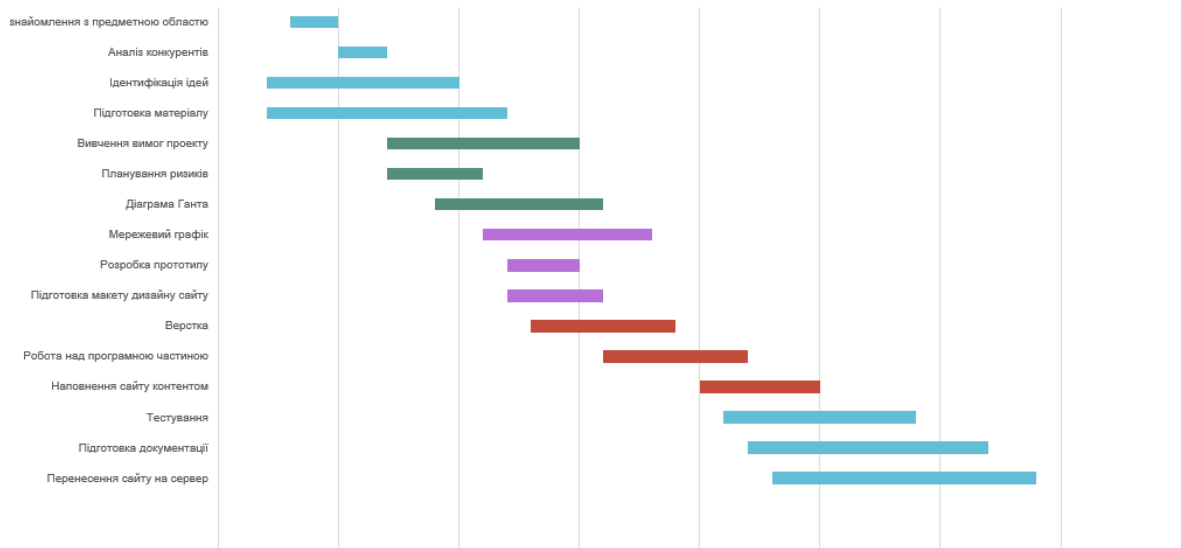


Рисунок А.4 – Діаграма Ганта (графік)

А.4 ПЛАНУВАННЯ РИЗИКІВ ПРОЕКТУ

Будь-яке управлінське рішення приймається в умовах ризику, викликаного неповнотою інформації про об'єкт управління і навколишнього його середовищі і обмеженням часу на його прийняття. Середовище прийняття рішень варіюється в залежності від ступеня ризику. Умови визначеності існують тільки тоді, коли керівник точно знає результат, який буде мати кожен вибір.

В умовах ризику ймовірність результату кожного рішення можна визначити тільки з відомою вірогідністю. Якщо інформації недостатньо для прогнозування рівня ймовірності результатів в залежності від вибору, умови прийняття рішення є невизначеними. В умовах невизначеності керівник на основі аналізу ризиків повинен встановити допустимість можливих ризиків і їх наслідків.

Управління і ризик не віддільні. Ризики управління організацією це ризики цілепокладання, маркетингу та менеджменту організації. Ризик цілепокладання - це можливість неправильного визначення цілей організації. При неправильно визначених цілях і поставлена мета діяльності організації не може бути успішною.

Ризик маркетингу - це можливість відхилень в результатах діяльності організації при неправильному визначенні ринкових умов - виборі ніші і позиціонування організації та її продукції на ринку.

Ризики менеджменту - це можливість неправильних дій в процесі досягнення поставлених цілей. Ризик менеджменту явно чи не явно спочатку присутній у всіх стандартах на системи менеджменту мінімум як запобіжні дії.

У ризик-менеджменті прийнято виділяти кілька ключових етапів:

1. Виявлення ризику, його аналіз і оцінка ймовірності його реалізації і масштабу наслідків.
2. Вибір методів та інструментів управління виявленим ризиком.
3. Розробка ризик-стратегії з метою зниження ймовірності реалізації ризику і мінімізації можливих негативних наслідків.
4. Реалізація ризик-стратегії.

Таблиця А.3 – Шкала оцінювання ймовірності виникнення та впливу ризику на виконання проекту

Оцінка	Ймовірність виникнення	Вплив ризику
1	Низька	Низький
2	Середня	Середній
3	Висока	Високий

3	RS_2	RS_3,	RS_5, RS_9
2	RS_1, RS_13	RS_4, RS_6	RS_7, RS_14
1	RS_12	RS_8, RS_11	RS_10, RS_15
	1	2	3

Рисунок А.5 – Матриця ймовірності виникнення ризиків та впливу ризику

На підставі отриманого значення індексу ризику класифікують: за рівнем ризику, що знаходиться в табл. А.4.

Таблиця А.4 – Оцінка ймовірності виникнення, величини витрат та індекс ризику

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_1	Відкритий	Непорозуміння між розробником та замовником	Низька	Середній	3	<ol style="list-style-type: none"> Провести діалог для визначення конкретної проблеми через яку виникло непорозуміння. Прийняти заходи для усунення проблеми. 	Попередження	<ol style="list-style-type: none"> Визначити ступінь непорозуміння. Ігнорувати його при низькому рівні.
RS_2	Відкритий	Поява альтернативного продукту	Низька	Високий	3	<ol style="list-style-type: none"> Визначити переваги та недоліки альтернативного продукту. Використати отриману інформацію для вдосконалення власного продукту 	Прийняття	Провести агресивний маркетинг свого продукту
RS_3	Відкритий	Нечітке завдання на розробку	Середня	Високий	6	<ol style="list-style-type: none"> Уточнити інформацію у замовника щодо поставлених вимог. 	Попередження	Зробити правки зважаючи на

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
						2. Зажати чіткого технічного завдання у замовника.		вимоги замовника
RS_4	Відкритий	Помилки проектування	Висока	Високий	9	Детально аналізувати ТЗ, та робити уточнення в замовника при виникненні непорозумінь	Пом'якшення	Створювати план дій на кожен етап роботи
RS_5	Відкритий	Збої в роботі програмного забезпечення	Низька	Високий	3	Визначити можливу причину збоїв та прийняти заходи	Попередження	
RS_6	Відкритий	Відсутність резервних копій даних	Низька	Середній	2	Позначити в плані робіт резервне збереження даних в прийнятну періодичність	Попередження	Зберігати копії даних на різних носіях
RS_7	Відкритий	Реалізація непотрібної функціональності	Низька	Низький	1	Проаналізувати доцільність реалізації непотрібної функціональності та прийняти заходи	Використання	Визначити затрати на реалізацію та

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
								обдумати актуальність
RS_8	Відкритий	Невиконання моніторингу проекту	Середня	Низький	2	Виділення окремого персоналу для моніторингу проекту	Перенос	Аналіз результатів проекту після кожного етапу роботи

ДОДАТОК Б. КОД РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

BooksController.php

Контролер книг.

```

<?php

namespace App\Http\Controllers;

use App\Models\Books;
use App\Models\Orders;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class BooksController extends Controller
{
    protected $publicStorage = "/uploads/book/";
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index(Request $request)
    {
        $model = Books::with('ganre', 'country', 'language', 'owner');
        if($request->owner) {
            $model = $model->whereHas('owner', function($query) use ($request) {
                $query->where('id', $request->owner);
            });
        }
        if($request->name) {
            $model = $model->where('title', 'LIKE', '%' . $request->name . '%');
        }
        if($request->author) {
            $model = $model->where('author', 'LIKE', '%' . $request->author . '%');
        }
        if($request->country) {
            $model = $model->where('country_id', $request->country);
        }
        if($request->language_id) {
            $model = $model->where('language_id', $request->language_id);
        }
        if(isset($request->ganre) && count($request->ganre) > 0) {
            $model = $model->whereIn('ganre_id', $request->ganre);
        }
        if(isset($request->archive)) {
            $model = $model->where('archive', $request->archive);
        }
        $data = $model->get();
        if($request->rate) {
            $data = $data->filter(function($model) use ($request) {
                return $model->owner['rate'] >= $request->rate[0] && $model->owner['rate'] <= $request->rate[1];
            });
        }
    }
}

```

```

    }
    return response()->json($data);
}

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Request $request)
{
    $model = new Books();
    $data = $request->all();
    $data['owner_id'] = Auth::id();
    $data['archive'] = $request['archive'] == 'true' ? 1 : 0;
    if (isset($request['new_photo']) && $request['new_photo'] != "null" && $request['new_photo'] != "undefined") {
        $name = $this->publicStorage . uniqid() . '.' . $request['new_photo']->getClientOriginalExtension();
        $request['new_photo']->move(public_path() . $this->publicStorage, $name);
        $data['photo'] = $name;
    }
    $response = $model->create($data);
    return response()->json([
        "id" => $response->id
    ]);
}

/**
 * Display the specified resource.
 *
 * @param \App\Models\Books $books
 * @return \Illuminate\Http\Response
 */
public function show($id)
{
    $model = Books::with('genre', 'country', 'language', 'owner', 'review.user')->find($id);
    $model->orders = Orders::with('bookFrom', 'bookTo')->where('books_id_from', $id)->orWhere('books_id_to', $id)-
>get();
    return response()->json($model);
}

/**
 * Update the specified resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param \App\Models\Books $books
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, $id)
{
    $model = Books::find($id);
    $data = $request->all();
    if (isset($request['new_photo']) && $request['new_photo'] != "null" && $request['new_photo'] != "undefined") {
        $name = $this->publicStorage . uniqid() . '.' . $request['new_photo']->getClientOriginalExtension();
        $request['new_photo']->move(public_path() . $this->publicStorage, $name);
        $data['photo'] = $name;
    }
    $model->update($data);
}

```

```

}

/**
 * Remove the specified resource from storage.
 *
 * @param \App\Models\Books $books
 * @return \Illuminate\Http\Response
 */
public function destroy($id)
{
    Books::find($id)->delete();
}
}

```

ForumController.php

Контролер форуму.

<?php

```

namespace App\Http\Controllers;

use App\Models\Forum;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class ForumController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index(Request $request)
    {
        $model = Forum::with('user', 'category')->withCount('allComments');
        if($request->users_id) {
            $model = $model->whereHas('user', function($query) use ($request) {
                $query->where('id', $request->users_id);
            });
        }
        if($request->name) {
            $model = $model->where('title', 'LIKE', '%' . $request->name . '%');
        }
        if(isset($request->theme) && count($request->theme) > 0) {
            $model = $model->whereIn('forum_theme_id', $request->theme);
        }
        $data = $model->get();
        return response()->json($data);
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)

```

```

{
    $data = $request->all();
    $data['users_id'] = Auth::id();
    $model = new Forum();
    $model->create($data);
}

/**
 * Display the specified resource.
 *
 * @param \App\Models\Forum $forum
 * @return \Illuminate\Http\Response
 */
public function show($id)
{
    $model = Forum::with('user', 'category', 'comments.comments', 'comments.user')->find($id);
    return response()->json($model);
}

/**
 * Remove the specified resource from storage.
 *
 * @param \App\Models\Forum $forum
 * @return \Illuminate\Http\Response
 */
public function destroy($id)
{
    Forum::find($id)->delete();
}
}

```

OrdersController.php

Контролер обміну книгами.

```
<?php
```

```

namespace App\Http\Controllers;

use App\Models\Orders;
use App\Models\Books;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class OrdersController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $model = Orders::get();
        return response()->json($model);
    }

    public function myOrders() {

```

```

    $userId = Auth::id();
    $model = Orders::with(
        'bookFrom.owner',
        'bookFrom.ganre',
        'bookFrom.language',
        'bookTo.owner',
        'bookTo.ganre',
        'bookTo.language'
    )->whereHas('bookFrom', function($query) use ($userId) {
        $query->where('owner_id', $userId);
    })->orWhereHas('bookTo', function($query) use ($userId) {
        $query->where('owner_id', $userId);
    })->get();
    return response()->json($model);
}

public function rejectOrders($id) {
    Orders::find($id)->update([
        'order_status_id' => 3
    ]);
}

public function acceptOrders($id) {
    Orders::find($id)->update([
        'order_status_id' => 2
    ]);
}

public function approveOrders(Request $request, $id) {
    Orders::find($id)->update([
        'order_status_id' => 4,
        'description_to' => $request->description_to
    ]);
}

public function sendOrders($id) {
    Orders::find($id)->update([
        'order_status_id' => 5
    ]);
}

public function takenOrders($id) {
    $model = Orders::with('bookTo', 'bookFrom')->find($id);

    Books::find($model->books_id_to)->update([
        'owner_id' => $model->bookFrom['owner_id']
    ]);

    Books::find($model->books_id_from)->update([
        'owner_id' => $model->bookTo['owner_id']
    ]);

    $model->update([
        'order_status_id' => 6
    ]);
}

/**

```

```

* Show the form for creating a new resource.
*
* @return \Illuminate\Http\Response
*/
public function lastOrders()
{
    $data = Orders::with('bookFrom', 'bookTo')->where('order_status_id', 10)->limit(5)->get();
    return response()->json($data);
}

/**
* Store a newly created resource in storage.
*
* @param \Illuminate\Http\Request $request
* @return \Illuminate\Http\Response
*/
public function store(Request $request)
{
    $data = $request->all();
    $model = new Orders();
    $model->create($data);
}

/**
* Display the specified resource.
*
* @param \App\Models\Orders $orders
* @return \Illuminate\Http\Response
*/
public function show($id)
{
    $model = Orders::with(
        'bookFrom.owner',
        'bookFrom.ganre',
        'bookFrom.language',
        'bookFrom.country',
        'bookTo.owner',
        'bookTo.ganre',
        'bookTo.country',
        'bookTo.language'
    )->find($id);
    return response()->json($model);
}
}

```

ApplicationController.php

Контролер для роботи з заявками.

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

use App\Models\TournamentTeamApplications;
use App\Models\TournamentUserApplications;

```



```

class ApplicationController extends Controller
{
    function teamApplication(Request $request, $id) {
        $model = new TournamentTeamApplications();
        $model->create([
            'tournaments_id' => $id,
            'teams_id' => $request->teams_id,
            'application_status_id' => 1
        ]);
    }

    function updateTeamApplication(Request $request, $id) {
        $model = TournamentTeamApplications::find($id);
        $data = $request->all();
        $model->update($data);
    }

    function userApplication(Request $request, $id) {
        $model = new TournamentUserApplications();
        $model->create([
            'tournaments_id' => $id,
            'users_id' => Auth::id(),
            'application_status_id' => 1
        ]);
    }

    function updateUserApplication(Request $request, $id) {
        $model = TournamentUserApplications::find($id);
        $data = $request->all();
        $model->update($data);
    }
}

```

Books.php

Модель книги.

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Books extends Model
{
    use HasFactory;

    protected $table = 'books';

    protected $fillable = [
        'title',
        'owner_id',
        'description',
        'photo',
        'ganre_id',
    ];
}

```

```

        'archive',
        'country_id',
        'language_id',
        'author'
    ];

    function genre() {
        return $this->belongsTo('App\Models\Genres', 'genre_id');
    }

    function country() {
        return $this->belongsTo('App\Models\Country', 'country_id');
    }

    function language() {
        return $this->belongsTo('App\Models\Languages', 'language_id');
    }

    function owner() {
        return $this->belongsTo('App\Models\User', 'owner_id');
    }

    function review() {
        return $this->hasMany('App\Models\Reviews');
    }
}

```

User.php

Модель користувачів.

```

<?php

namespace App\Models;

// use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Laravel\Sanctum\HasApiTokens;

class User extends Authenticatable
{
    use HasApiTokens, HasFactory, Notifiable;

    /**
     * The attributes that are mass assignable.
     *
     * @var array<int, string>
     */
    protected $fillable = [
        'name',
        'surname',
        'email',
        'password',
        'phone',
        'description',
        'citate',
        'photo',
    ];
}

```

```

        'best_book',
        'role',
        'country_id',
        'is_ban'
    ];

    /**
     * The attributes that should be hidden for serialization.
     *
     * @var array<int, string>
     */
    protected $hidden = [
        'password',
        'remember_token',
    ];

    /**
     * The attributes that should be cast.
     *
     * @var array<string, string>
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];

    protected $appends = [
        'rate'
    ];

    function getRateAttribute() {
        return $this->evaluation->sum('rate') > 0 ? round($this->evaluation->sum('rate') / count($this->evaluation)) : $this->evaluation->sum('rate');
    }

    public function evaluation()
    {
        return $this->hasMany('App\Models\UsersRate', 'users_id_who');
    }

    function books() {
        return $this->hasMany('App\Models\Books', 'owner_id');
    }

    function country() {
        return $this->belongsTo('App\Models\Country', 'country_id');
    }
}

```

Forum.php

Модель форума.

<?php

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

```

class Forum extends Model
{
    use HasFactory;

    protected $table = 'forum';

    protected $fillable = [
        'title',
        'theme',
        'users_id',
        'forum_theme_id'
    ];

    protected $casts = [
        'created_at' => 'datetime:d.m.Y h:m',
    ];

    function user() {
        return $this->belongsTo('App\Models\User', 'users_id');
    }

    function category() {
        return $this->belongsTo('App\Models\Theme', 'forum_theme_id');
    }

    function comments() {
        return $this->hasMany('App\Models\ForumComments')->whereNull('forum_comments_id');
    }

    function allComments() {
        return $this->hasMany('App\Models\ForumComments');
    }
}

```

Orders.php

МОДЕЛЬ ЗАМОВЛЕНЬ.

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Orders extends Model
{
    use HasFactory;

    protected $table = 'order';

    protected $fillable = [
        'order_status_id',
        'books_id_from',
        'books_id_to',
        'description_from',
        'description_to'
    ];
}

```

```

protected $casts = [
    'created_at' => 'datetime:d.m.Y h:m',
];

function bookFrom() {
    return $this->belongsTo('App\Models\Books', 'books_id_from');
}

function bookTo() {
    return $this->belongsTo('App\Models\Books', 'books_id_to');
}
}

```

app.js

Головний файл який підключає всі бібліотеки та компоненти.

```

import './bootstrap';
import './sass/app.scss';

import 'material-design-icons-iconfont/dist/material-design-icons.css'

import Notifications from 'notiwind';

import { createApp } from 'vue';

import app from './Page/app.vue'
import router from './router/app.js'
import store from './store.js';

createApp(app)
    .use(router)
    .use(store)
    .use(Notifications)
    .mount("#app");

```

router/app.js

Файл маршрутизації vue.js.

```

import { createWebHistory, createRouter } from "vue-router";
import store from './store';

import Books from './Page/profile/Books.vue';
import Forum from './Page/profile/Forum.vue';
import Orders from './Page/profile/Orders.vue';

function lazyLoad(view) {
    return () => import(`./Page/${view}.vue`);
}

const routes = [
    {
        path: "/",
        name: "welcome",
        component: lazyLoad("Welcome"),
    },
    {
        path: "/login",

```

```

name: "login",
component: lazyLoad("Login"),
},
{
  path: "/register",
  name: "register",
  component: lazyLoad("Register"),
},
{
  path: "/:any",
  component: lazyLoad("Main"),
  children: [
    {
      path: "/home",
      name: "books",
      component: lazyLoad("Books")
    },
    {
      path: "/users",
      name: "users",
      component: lazyLoad("Users")
    },
    {
      path: "/offline",
      name: "offline",
      component: lazyLoad("Offline")
    },
    {
      path: "/forum",
      name: "forum",
      component: lazyLoad("Forum")
    },
    {
      path: "/forum/:id",
      name: "forum-item",
      component: lazyLoad("ForumItem")
    },
    {
      path: "/profile/:child?",
      name: "profile",
      component: lazyLoad("Profile"),
      meta: {
        requiresAuth: true
      },
      children: [
        {
          path: "/profile/",
          name: "profile-book",
          component: Books
        },
        {
          path: "/profile/forum",
          name: "profile-forum",
          component: Forum
        },
        {
          path: "/profile/orders",
          name: "profile-orders",

```

```

        component: Orders
      },
    ],
  },
  {
    path: "/edit",
    name: "edit",
    component: lazyLoad("EditProfile"),
    meta: {
      requiresAuth: true
    },
  },
  {
    path: "/add",
    name: "add",
    component: lazyLoad("CreateBook"),
    meta: {
      requiresAuth: true
    },
  },
  {
    path: "/order/:id",
    name: "order",
    component: lazyLoad("Order")
  },
  {
    path: "/book/:id",
    name: "book",
    component: lazyLoad("BookItem")
  },
  {
    path: "/book/:id/edit",
    name: "book-edit",
    component: lazyLoad("CreateBook"),
    meta: {
      requiresAuth: true
    },
  },
  {
    path: "/user/:id",
    name: "user",
    component: lazyLoad("Profile"),
    children: [
      {
        path: "/user/:id",
        name: "user-book",
        component: Books
      },
      {
        path: "/user/:id/forum",
        name: "user-forum",
        component: Forum
      },
      {
        path: "/user/:id/orders",
        name: "user-orders",
        component: Orders
      },
    ],
  },

```

```

    ]
  },
]
}
];

const router = createRouter({
  history: createWebHistory(),
  routes,
});

router.beforeEach((to, from, next) => {
  if(to.matched.some(record => record.meta.requiresAuth)) {
    if (store.getters.isLoggedIn) {
      store.dispatch('update');
      next();
    } else {
      next('/login');
    }
  } else {
    if((to.name === 'welcome' || to.name === 'login' || to.name === 'register') && store.getters.isLoggedIn) {
      store.dispatch('update');
      next({
        path: '/home',
        params: { nextUrl: to.fullPath }
      });
    } else {
      store.dispatch('update');
      next();
    }
  }
})

export default router;

```

Books.vue

КОМПОНЕНТ сторінки з книгами.

```

<template>
  <div class="wrapper">
    <Back/>
    <div class="flex justify-between title-head">
      <div class="title-page">Книги користувачів</div>
      <div class="select-group head">
        <select>
          <option value="1">Сортувати за алфавітом</option>
          <option value="1">Сортувати за роком</option>
          <option value="1">Сортувати за рейтингом автора</option>
        </select>
      </div>
    </div>
    <div class="grid grid-cols-12 md:gap-10 gap-0">
      <div class="md:col-span-3 col-span-12 filter">
        <div class="filter-grout">
          <input v-model="filter.name" type="text">
        </div>

```



```

<div class="filter-grout">
  <div class="title">Жанр</div>
  <div class="checkbox-group" v-for="(item, i) in ganres" :key="i">
    <input type="checkbox" v-model="filter.ganre" :id="`category-${i}`" :value="item.id">
    <label :for="`category-${i}`">{{ item.title }}</label>
  </div>
</div>
<div class="filter-grout">
  <div class="title">Країна</div>
  <div class="select-group">
    <select v-model="filter.country">
      <option v-for="(item, index) in country" :value="item.id">{{ item.title }}</option>
    </select>
  </div>
</div>
<div class="filter-grout">
  <div class="title">Мова</div>
  <div class="select-group">
    <select v-model="filter.language_id">
      <option v-for="(item, index) in languages" :value="item.id">{{ item.title }}</option>
    </select>
  </div>
</div>
<div class="filter-grout">
  <div class="title">Автор</div>
  <div class="select-group">
    <select v-model="filter.author">
      <option v-for="(item, index) in authors" :value="item.author">{{ item.author }}</option>
    </select>
  </div>
</div>
<div class="filter-grout">
  <div class="title">Рейтинг користувача</div>
  <vue-slider
    :min="0"
    :max="5"
    :interval="1"
    v-model="filter.rate"
  />
</div>
<div class="filter-grout">
  <button @click="getData()">Фільтрація</button>
</div>
</div>
<div class="md:col-span-9 col-span-12">
  <div class="grid grid-cols-12 md:gap-10 gap-0">
    <div class="md:col-span-4 col-span-12" v-for="(item, index) in books" :key="index">
      <AdsItem :item="item" />
    </div>
  </div>
  <Pagination />
</div>
</div>
</div>
</template>
<script>
import AdsItem from '../Components/AdsItem.vue'
import Pagination from '../Components/Pagination.vue'

```

```

import Back from "../Components/Back.vue";

import { reactive, toRefs } from 'vue'
import VueSlider from 'vue-slider-component'
import 'vue-slider-component/theme/antd.css'

import country from "../mixins/country";
import ganres from "../mixins/ganres";
import languages from "../mixins/language";
import authors from "../mixins/authors";

export default {
  mixins: [country, ganres, languages, authors],
  components: {
    AdsItem,
    Pagination,
    VueSlider,
    Back
  },
  data() {
    return {
      books: [],
      authors: [],
      filter: {
        rate: [0, 5],
        name: "",
        ganre: [],
        country: null,
        author: null,
        language_id: null
      }
    }
  },
  created() {
    this.getData();
  },
  methods: {
    getData() {
      axios.get('/api/books', {
        params: this.filter
      })
      .then((response) => {
        this.books = response.data;
      });
    }
  },
  setup() {
    const data = reactive({ value: 0 })
    return toRefs(data)
  },
}
</script>
<style lang="scss" scoped>

</style>

```

Users.vue

Компонент сторінки користувачів.

```

<template>
  <div class="wrapper">
    <Back/>
    <div class="flex justify-between title-head">
      <div class="title-page">Користувачі</div>
      <div class="select-group head">
        <select >
          <option value="1">Сортувати за рейтингом</option>
          <option value="1">Сортувати за датою реєстрації</option>
          <option value="1">Сортувати за алфавітом</option>
        </select>
      </div>
    </div>
    <div class="grid grid-cols-12 md:gap-10 gap-0">
      <div class="md:col-span-3 col-span-12 filter">
        <div class="filter-grout">
          <input v-model="filter.name" type="text">
        </div>
        <div class="filter-grout">
          <div class="title">Країна</div>
          <div class="select-group">
            <select v-model="filter.country">
              <option v-for="(item, index) in country" :value="item.id">{{ item.title }}</option>
            </select>
          </div>
        </div>
      </div>
      <div class="filter-grout">
        <div class="title">Рейтинг користувача</div>
        <vue-slider
          :min="0"
          :max="5"
          :interval="1"
          v-model="filter.rate"
        />
      </div>
      <div class="filter-grout">
        <button @click="getData()">Фільтрація</button>
      </div>
    </div>
    <div class="md:col-span-9 col-span-12">
      <div class="grid grid-cols-12 md:gap-10 gap-0">
        <div class="md:col-span-3 col-span-12" v-for="(item, index) in users" :key="index">
          <div class="user-item">
            <router-link :to="/user/${item.id}">
              
            </router-link>
            <div class="name">
              <span>{{ item.name }} {{ item.surname }}</span>
              <Stars :data="item.rate" />
            </div>
            <div class="description">{{ item.description }}</div>
          </div>
        </div>
      </div>
    </div>
    <Pagination />
  </div>

```

```

    </div>
  </div>
</template>
<script>
import Pagination from '../Components/Pagination.vue'
import Stars from '../Components/Stars.vue'
import Back from "../Components/Back.vue";

import { reactive, toRefs } from 'vue'
import VueSlider from 'vue-slider-component'
import 'vue-slider-component/theme/antd.css'

export default {
  components: {
    Pagination,
    VueSlider,
    Stars,
    Back
  },
  data() {
    return {
      country: [],
      users: [],
      filter: {
        rate: [0, 5],
        name: "",
        country: null
      }
    }
  },
  created() {
    this.getCountry();
    this.getUsers();
  },
  methods: {
    getCountry() {
      axios.get('/api/country')
        .then((response) => {
          this.country = response.data;
        });
    },
    getUsers() {
      var params = Object.assign(this.filter, {
        is_ban: false
      });
      axios.get('/api/users', {
        params
      })
        .then((response) => {
          this.users = response.data;
        });
    }
  },
  setup() {
    const data = reactive({ value: 0 })
    return toRefs(data)
  },
}

```

```

</script>
<style lang="scss" scoped>
  .wrapper {
    color: #000;
  }
  .user-item {
    border: 1px solid #fff;
    transition: 0.3s;
    padding: 5px;
    img {
      width: 100%;
      height: 236px;
      object-fit: cover;
      margin-bottom: 17px;
    }
    .name {
      font-weight: 700;
      font-size: 13px;
      display: flex;
      justify-content: space-between;
      align-items: center;
    }
    .description {
      font-weight: 400;
      font-size: 10px;
      line-height: 15px;
      margin-top: 10px;
    }
    &:hover {
      border: 1px solid #000000;
      box-shadow: -6px 6px 0px #000000;
    }
  }
</style>

```

Forum.vue

КОМПОНЕНТ ФОРУМУ.

```

<template>
  <div class="wrapper">
    <Back/>
    <div class="grid grid-cols-12 md:gap-2 gap-0">
      <div class="md:col-span-9 col-span-12">
        <div class="flex justify-between title-head">
          <div class="title-page">Форум буккросинг</div>
          <div class="select-group head">
            <select>
              <option value="1">Сортувати за датою</option>
              <option value="1">Сортувати за темою</option>
              <option value="1">Сортувати за коментарями</option>
            </select>
          </div>
        </div>
        <div class="grid grid-cols-2 md:gap-5 gap-0 form" v-if="activeForm">
          <div class="form-group">
            <label for="">Заголовок</label>
            <input v-model="theme.title" type="text">
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

</div>
<div class="form-group">
  <label for="">Тема</label>
  <select v-model="theme.forum_theme_id">
    <option v-for="(item, index) in themes" :key="index" :value="item.id">{{ item.title }}</option>
  </select>
</div>
<div class="col-span-2 form-group">
  <label for="">Опис</label>
  <textarea v-model="theme.theme" rows="5"></textarea>
  <div class="button flex justify-end">
    <button @click="activeForm = !activeForm">Видалити</button>
    <button @click="createTheme()" class="active">Створити тред</button>
  </div>
</div>
</div>
<div class="item" v-for="(item, index) in forums" :key="index">
  <router-link :to="`/forum/${item.id}`">
    <div class="head">
      <router-link :to="`/user/${item.user.id}`">
        <span>{{ item.user.name }} {{ item.user.surname }}</span>
      </router-link>
      <span>Тематика: {{ item.category.title }}</span>
    </div>
    <p>{{ item.theme }}</p>
    <div class="flex justify-between pt-4">
      <span class="comments">{{ item.all_comments_count }} коментарів</span>
      <span class="date">{{ item.created_at }}</span>
    </div>
  </router-link>
</div>
<Pagination />
</div>
<div class="md:col-span-3 col-span-12 filter">
  <div class="filter-grout create">
    <button @click="activeForm = !activeForm" v-if="authUser()">Створити тред</button>
  </div>
  <div class="filter-grout mb-5">
    <input v-model="filter.name" type="text">
  </div>
  <div class="filter-grout">
    <div class="title">Тема</div>
    <div class="checkbox-group" v-for="(item, i) in themes" :key="i">
      <input type="checkbox" v-model="filter.theme" :id="`category-${i}`" :value="item.id">
      <label :for="`category-${i}`">{{ item.title }}</label>
    </div>
  </div>
  <div class="filter-grout">
    <button @click="getData()">Фільтрація</button>
  </div>
</div>
</div>
</div>
</div>
</template>
<script>
import Pagination from '../Components/Pagination.vue'
import Back from '../Components/Back.vue';

```

```

import { mapGetters } from 'vuex';

export default {
  components: {
    Pagination,
    Back
  },
  data() {
    return {
      activeForm: false,
      forums: [],
      themes: [],
      theme: {
        title: "dsadsa",
        theme: "dsadsa",
        forum_theme_id: null
      },
      filter: {
        name: "",
        theme: []
      }
    }
  },
  created() {
    this.getData();
    this.getThemes();
  },
  methods: {
    ...mapGetters(["authUser"]),
    getData() {
      axios.get('/api/forum', {
        params: this.filter
      })
      .then((response) => {
        this.forums = response.data;
      });
    },
    getThemes() {
      axios.get('/api/themes')
      .then((response) => {
        this.themes = response.data;
      });
    },
    createTheme() {
      axios.post('/api/forum', this.theme)
      .then(() => {
        this.getData();
        this.activeForm = false;
      });
    }
  }
}
</script>
<style lang="scss" scoped>
.form {
  border: 1px solid #000;
  padding: 20px;
  margin-bottom: 40px;

```

```

.form-group {
  label {
    font-weight: 600;
    font-size: 14px;
  }
  input, select, textarea {
    border: 1px solid #000000;
    padding: 15px 20px;
    color: #000;
    font-weight: 600;
    font-size: 12px;
    width: 100%;
  }
  select {
    -webkit-appearance: none;
    -moz-appearance: none;
    text-indent: 1px;
    text-overflow: "";
    background: url('/img/direction-down.svg') no-repeat;
    background-position: right 10px center;
  }
}
.buttonnts {
  button {
    padding: 10px 60px;
    border: 1px solid #000000;
    font-weight: 500;
    font-size: 14px;
    margin-left: 10px;
  }.active {
    background: #000;
    color: #fff;
  }
}
.item {
  padding: 21px 70px 14px 37px;
  border: 1px solid #000;
  font-weight: 400;
  font-size: 12px;
  line-height: 15px;
  margin-bottom: 20px;
  transition: 0.3s;
  .head {
    display: flex;
    justify-content: space-between;
    font-weight: 600;
    font-size: 14px;
    margin-bottom: 20px;
    width: 100%;
  }
  .date {
    font-weight: 300;
    font-size: 8px;
  }
  .comments {
    font-weight: 700;
    font-size: 12px;
  }
}

```



```

    line-height: 15px;
    &:hover {
      text-decoration-line: underline;
    }
  }
  &:hover {
    border: 1px solid #000000;
    box-shadow: -6px 6px 0px #000000;
  }
}
.filter-grout.create {
  margin-bottom: 30px;
  button {
    height: 54px;
  }
}
@media screen and (max-width: 600px) {
  .title-head {
    margin-bottom: 20px;
  }
  .item {
    padding: 20px;
  }
}
</style>

```

Order.vue

КОМПОНЕНТ ДЛЯ ОБМІНУ КНИГАМИ.

```

<template>
  <div class="wrapper">
    <modal v-if="modalOpen" @close="modalOpen = false" :size="500" style="position: fixed;">
      <template v-slot:header>Залиште відгук</template>
      <template v-slot:body>
        <div class="add-review">
          <div class="stars">
            
          </div>
          <div class="flex justify-center">
            <button @click="addRate()">Зберегти</button>
          </div>
        </div>
      </template>
    </modal>
    <Back/>
    <div class="title-page">Запит на обмін</div>
    <div class="grid grid-cols-12 gap-10">
      <div class="md:col-span-3 col-span-12">
        <Book :item="order.book_to" />
      </div>
      <div class="md:col-span-9 col-span-12">
        <ol class="items-center sm:flex mb-6">
          <li class="relative mb-6 sm:mb-0">
            <div class="flex items-center">
              <div
                class="flex z-10 justify-center items-center w-6 h-6 bg-blue-200 rounded-full ring-0 sm:ring-8 shrink-
0">

```

```

    <div class="icon-step active"></div>
  </div>
  <div class="hidden sm:flex w-full bg-gray-200 h-0.5 dark:bg-gray-700"></div>
</div>
<div class="mt-3 sm:pr-8">
  <h3 class="text-lg font-semibold">Запит</h3>
  <time
    class="block mb-2 text-sm font-normal leading-none text-gray-400 dark:text-gray-500">Підтвердіть
    запит на обмін книгою</time>
  </div>
</li>
<li :class="['relative', 'mb-6', 'sm:mb-0', order.order_status_id == 2 || order.order_status_id == 4 ||
order.order_status_id == 5 || order.order_status_id == 6 ? '' : 'opacity']">
  <div class="flex items-center">
    <div
      class="flex z-10 justify-center items-center w-6 h-6 bg-blue-200 rounded-full ring-0 sm:ring-8 shrink-
0">
      <div :class="['icon-step', order.order_status_id == 2 || order.order_status_id == 4 || order.order_status_id
== 5 || order.order_status_id == 6 ? 'active' : '']"></div>
    </div>
    <div class="hidden sm:flex w-full bg-gray-200 h-0.5 dark:bg-gray-700"></div>
  </div>
  <div class="mt-3 sm:pr-8">
    <h3 class="text-lg font-semibold">Оформлення</h3>
    <time
      class="block mb-2 text-sm font-normal leading-none text-gray-400 dark:text-gray-500">Внесення
      контактних даних</time>
    </div>
  </li>
  <li :class="['relative', 'mb-6', 'sm:mb-0', order.order_status_id == 4 || order.order_status_id == 5 ||
order.order_status_id == 6 ? '' : 'opacity']">
    <div class="flex items-center" @click="sended()">
      <div
        class="flex z-10 justify-center items-center w-6 h-6 bg-blue-200 rounded-full ring-0 sm:ring-8 shrink-
0">
        <div :class="['icon-step', order.order_status_id == 5 || order.order_status_id == 6 ? 'active' : '']"></div>
      </div>
      <div class="hidden sm:flex w-full bg-gray-200 h-0.5 dark:bg-gray-700"></div>
    </div>
    <div class="mt-3 sm:pr-8">
      <h3 class="text-lg font-semibold">Відправка</h3>
      <time
        class="block mb-2 text-sm font-normal leading-none text-gray-400 dark:text-gray-500">Віправка
        книги отримувачу</time>
      </div>
    </li>
    <li :class="['relative', 'mb-6', 'sm:mb-0', order.order_status_id == 5 || order.order_status_id == 6 ? '' : 'opacity']">
      <div class="flex items-center" @click="order.order_status_id == 5 ? taken() : "">
        <div
          class="flex z-10 justify-center items-center w-6 h-6 bg-blue-200 rounded-full ring-0 sm:ring-8 shrink-
0">
          <div :class="['icon-step', order.order_status_id == 6 ? 'active' : '']"></div>
        </div>
        <div class="hidden sm:flex w-full bg-gray-200 h-0.5 dark:bg-gray-700"></div>
      </div>
      <div class="mt-3 sm:pr-8">
        <h3 class="text-lg font-semibold">Підтвердження</h3>
        <time

```

```

        class="block mb-2 text-sm font-normal leading-none text-gray-400 dark:text-gray-
500">Підтвердження
        отримання</time>
      </div>
    </li>
    <li :class="['relative', 'mb-6', 'sm:mb-0', order.order_status_id == 6 ? '' : 'opacity']">
      <div class="flex items-center">
        <div
          class="flex z-10 justify-center items-center w-6 h-6 bg-blue-200 rounded-full ring-0 sm:ring-8 shrink-
0">
          <div :class="['icon-step', order.order_status_id == 6 ? 'active' : '']"></div>
        </div>
      </div>
      <div class="mt-3 sm:pr-8">
        <h3 class="text-lg font-semibold">Успіх</h3>
        <time
          class="block mb-2 text-sm font-normal leading-none text-gray-400 dark:text-gray-500">Обмін
виконано успішно</time>
        </div>
      </li>
    </ol>

    <div class="add-review">
      <label for="">Введіть контактні дані</label>
      <textarea v-model="order.description_to" :disabled="(order.order_status_id != 2)"></textarea>
      <div class="flex justify-end">
        <button @click="approve()" :disabled="(order.order_status_id != 2)">Надіслати</button>
      </div>
    </div>
  </div>
</div>
<hr>
<div class="grid grid-cols-12 gap-10">
  <div class="md:col-span-9 col-span-12">
    <div class="add-review">
      <label for="">Інформація від іншого користувача</label>
      <div class="another-user mt-3">
        <div class="head">
          <span>{{ order.book_from.owner.name }} {{ order.book_from.owner.surname }}</span>
        </div>
        <p>{{ order.description_from }}</p>
        <div class="date">{{ order.created_at }}</div>
      </div>
    </div>
  </div>
  <div class="md:col-span-3 col-span-12">
    <Book :item="order.book_from" />
  </div>
</div>
</div>
</template>
<script>
import Book from './Components/AdsItem.vue';
import Back from './Components/Back.vue';
import Modal from './Components/Modal.vue'
export default {
  components: {
    Book,

```

```

    Back,
    Modal
  },
  data() {
    return {
      modalOpen: false,
      rate: 0,
      order: {
        description_to: "",
        book_from: {
          ganre: {},
          language: {},
          owner: {},
          country: {}
        },
        book_to: {
          ganre: {},
          language: {},
          owner: {},
          country: {}
        }
      }
    }
  },
  created() {
    this.getData();
  },
  methods: {
    approve() {
      axios.post('/api/orders/' + this.$route.params.id + '/approve', {
        description_to: this.order.description_to
      })
        .then(() => {
          this.getData();
        });
    },
    sended() {
      axios.post('/api/orders/' + this.$route.params.id + '/sended')
        .then(() => {
          this.getData();
        });
    },
    taken() {
      axios.post('/api/orders/' + this.$route.params.id + '/taken')
        .then(() => {
          this.getData();
          this.modalOpen = true;
        });
    },
    getData() {
      axios.get('/api/orders/' + this.$route.params.id)
        .then((response) => {
          this.order = response.data;
        });
    },
    addRate() {
      axios.post('/api/users/rate', {
        rate: this.rate,

```

```

        users_id_who: this.order.book_to.owner_id
    })
    .then(() => {
        window.location.href = '/profile/orders';
    })
}
}
}
</script>
<style lang="scss" scoped>
.add-review {
    button {
        background: none;
        width: 270px;
        display: inline-block;
        height: 47px;
        font-weight: 700;
        font-size: 15px;
        color: #000;
        border: 1px solid #000000;
        box-shadow: -6px 6px 0px #000000;
        margin-top: 15px;
    }
}
.stars {
    width: 100%;
    display: flex;
    justify-content: center;
    margin-bottom: 30px;
    margin-top: 40px;
}
.stars img {
    margin-right: 17px;
    width: 20px;
    height: 20px;
}
.icon-step {
    width: 100%;
    height: 100%;
    background: url('/img/autorenew.svg') no-repeat;
    background-size: 70%;
    background-position: center center;
    cursor: pointer;
    &:hover {
        background: url('/img/check-bold.svg') no-repeat;
        background-size: 70%;
        background-position: center center;
    }
}
.active {
    background: url('/img/check-bold.svg') no-repeat;
    background-size: 70%;
    background-position: center center;
    cursor: auto;
}
.opacity {
    opacity: 0.5;
}
}

```

```
hr {
  height: 2px;
  background: #000;
  margin: 40px 0;
}

.wrapper {
  color: #000;
}

.add-review {
  label {
    color: #000;
    font-weight: 600;
    font-size: 14px;
    line-height: 24px;
    margin-bottom: 4px;
    display: block;
    width: 100%;
  }

  textarea {
    background: #fff;
    border: 1px solid #000;
    color: rgb(0, 0, 0);
    font-weight: 300;
    font-size: 14px;
    padding: 15px 39px;
    display: block;
    width: 100%;
    transition: 0.3s;
    height: 180px;

    &:focus {
      border: 1px solid #000;
    }
  }

  button {
    width: 225px;
    height: 56px;
    color: #fff;
    background: #000;
    margin-top: 20px;
  }

  margin-bottom: 23px;
}

.select-group select {
  padding: 10px;
}

.another-user {
  padding: 21px 70px 14px 37px;
  border: 1px solid #000000;
  box-shadow: -6px 6px 0px #000000;
  font-weight: 400;
}
```

```
font-size: 12px;
line-height: 15px;
margin-bottom: 20px;
transition: 0.3s;

.head {
  display: flex;
  justify-content: space-between;
  font-weight: 600;
  font-size: 14px;
  margin-bottom: 20px;
  width: 100%;
}

.date {
  font-weight: 300;
  font-size: 8px;
  margin-top: 17px;
  text-align: end;
}
}
</style>
```