

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра інтелектуальних технологій


**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
БАКАЛАВРА**

на тему:

«Рекомендаційний модуль для шкільної електронної
бібліотеки»

Галузь знань **12 «Інформаційні технології»**
Спеціальність **122 «Комп'ютерні науки»**
Освітня програма **«Комп'ютерні науки»**
Освітній рівень: бакалавр

Виконала: студентка 4 курсу, групи КН- 41

Юр Анна Юріївна 

Керівник: Красовська Г.В. 

Кандидат технічних наук, доцент, доцент
кафедри інтелектуальних технологій

Випускна кваліфікаційна робота бакалавра допущена до захисту
рішенням кафедри *інтелектуальних технологій*
Протокол № 11 від 06.06.2022 р.
Зав. кафедри _____ доц. Іларіонов О.Є.

Київ - 2022

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА
ШЕВЧЕНКА**

Факультет інформаційних технологій
Кафедра інтелектуальних технологій
Спеціальність 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
Завідувач кафедри
інтелектуальних технологій
Іларіонов О.Є.

“ ___ ” _____ 2022 р.

**ЗАВДАННЯ
НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ**

_____ Юр Анні Юріївні _____

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи): “Рекомендаційний модуль для шкільної електронної бібліотеки”, затверджена протоколом засідання кафедри від 23 грудня 2021 р. № 4
2. Термін здачі студентом закінченого проекту (роботи): 29 травня 2022 року
3. Вихідні дані до проекту (роботи): Закон України “Про авторське право та суміжні права”, Закон України “Про захист персональних даних.
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
Вступ, системний аналіз технології обліку книжок та надання рекомендацій щодо вибору літератури у шкільній електронній бібліотеці, проектування рекомендаційного модуля для шкільної електронної бібліотеки, розробка рекомендаційного модуля для шкільної електронної бібліотеки, висновки.
5. Перелік презентаційного матеріалу (з точним зазначенням обов'язкових презентацій)
 1. Актуальність теми (1 слайд);
 2. Мета, предмет та об'єкт дослідження (1 слайд);
 3. Системний аналіз технології обліку книжок та надання рекомендацій щодо вибору літератури у шкільній електронній бібліотеці (8 слайдів)
 4. Проектування рекомендаційного модуля для шкільної електронної бібліотеки (6 слайдів)
 5. Розробка рекомендаційного модуля для шкільної електронної бібліотеки (2 слайди)
 6. Перевірка працездатності програмного модулю (6 слайдів)
 7. Висновки по роботі (1 слайд)

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

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 15 лютого 2022 року

Керівник



/ Красовська Г. В. /

(підпис)

(ПІБ)

Завдання прийняв до виконання



/ Юр А. Ю. /

(підпис)

(ПІБ)

КАЛЕНДАРНИЙ ПЛАН

Пор. №	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів випускної кваліфікаційної роботи	Примітка
1	Дослідження джерел та збір інформації для випускної кваліфікаційної роботи.	15.02.2022 - 1.03.2022	
2	Системний аналіз технології обліку книжок та надання рекомендацій щодо вибору літератури у шкільній електронній бібліотеці.	1.03.2022 - 15.03.2022	
3	Проектування рекомендаційного модуля для шкільної електронної бібліотеки.	15.03.2022 - 05.04.2022	
4	Розробка рекомендаційного модуля для шкільної електронної бібліотеки.	05.04.2022 - 09.05.2022	
5	Налагодження та тестування рекомендаційного модуля.	09.05.2022 - 16.05.2022	
6	Оформлення пояснювальної записки та підготовка презентаційних матеріалів.	16.05.2022 - 29.05.2022	

Студент-дипломник



/ Юр А. Ю. /

(підпис)

Керівник випускної кваліфікаційної роботи



/ Красовська Г. В. /

(підпис)

(ПІБ)

Анотація

Юр Анна Юріївна виконала випускню кваліфікаційну роботу на тему: «Рекомендаційний модуль для шкільної електронної бібліотеки» за спеціальністю 122 «Комп'ютерні науки».

У випускній кваліфікаційній роботі проведено аналіз актуальності розробки рекомендаційного модуля для шкільної електронної бібліотеки. Створено відповідний модуль рекомендацій для електронної шкільної бібліотеки, котрий дозволяє пропонувати учням нові книжки до прочитання враховуючи їх інтереси.

Ключові слова: шкільна бібліотека, електронна бібліотека, рекомендаційний модуль, рекомендації.

Summary

The degree project: «Recommendation module for school digital library» has completed by Yur Anna specialty 122 – «Computer Sciences».

In this graduation thesis the analysis of urgency of use for the recommendation module for the school digital library is carried out. An appropriate module of recommendations for the digital school library has been created, which allows the offering of new books to read for students, taking into account their interests.

Keywords: school library, digital library, recommendation module, recommendations.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ТЕХНОЛОГІЇ ОБЛІКУ КНИЖОК ТА НАДАННЯ РЕКОМЕНДАЦІЙ ЩОДО ВИБОРУ ЛІТЕРАТУРИ У ШКІЛЬНІЙ ЕЛЕКТРОННІЙ БІБЛІОТЕЦІ	9
1.1 Аналіз технології обліку книжок у шкільній бібліотеці	9
1.2 Цілі створення рекомендаційного модуля для шкільної електронної бібліотеки	12
1.3 Аналіз підходів до побудови рекомендаційних систем	13
1.3.1 Рекомендаційні системи: основні поняття та визначення	13
1.3.2 Підходи до побудови рекомендаційних систем	14
1.3.2.1 Колаборативна фільтрація	14
1.3.2.2 Фільтрування на основі вмісту	15
1.3.2.3 Гібридна система фільтрації	15
1.3.3 Порівняння основних стратегій створення рекомендаційних систем	16
1.4. Вимоги до рекомендаційного модулю шкільної електронної бібліотеки	17
РОЗДІЛ 2. ПРОЕКТУВАННЯ РЕКОМЕНДАЦІЙНОГО МОДУЛЯ ДЛЯ ШКІЛЬНОЇ ЕЛЕКТРОННОЇ БІБЛІОТЕКИ	21
2.1 Функціональний аналіз рекомендаційного модуля для шкільної електронної бібліотеки	21
2.2 Діаграми аналізу процесу формування рекомендацій	25
2.3 Архітектура інформаційної системи рекомендаційного модулю шкільної електронної бібліотеки	26
2.4 Інформаційне забезпечення рекомендаційного модуля електронної шкільної бібліотеки	28
2.3 Висновки до другого розділу	31
РОЗДІЛ 3. РОЗРОБКА РЕКОМЕНДАЦІЙНОГО МОДУЛЯ ДЛЯ ШКІЛЬНОЇ ЕЛЕКТРОННОЇ БІБЛІОТЕКИ	31
3.1 Вибір мови розробки	31
3.2 Розробка та опис програмного інтерфейсу з користувачем (керівництво користувача) рекомендаційного модуля	32
3.3 Перевірка працездатності програмного рекомендаційного модуля	37
ВИСНОВКИ	41
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	42

ДОДАТКИ

ВСТУП

У сучасному світі інформаційних технологій люди щоденно стикаються з надзвичайно великим обсягом інформації. В таких умовах дуже важко зорієнтуватись та обробити всі запропоновані дані для знаходження необхідної інформації. Кожен електронний ресурс являє собою набір даних котрі необхідно обробляти. Сервіси зацікавлені в особистих рекомендаціях та рекламі для користувачів, адже це значно збільшує прибуток компанії. Як результат, інтерес до розробки рекомендаційних систем за останні роки дуже сильно зріс. Передові компанії якнайшвидше впроваджують їх у свою роботу, тож завдання розробки ефективних рекомендаційних систем є актуальним.

Рекомендаційна система – це програма, метою якої є фільтрація інформації та формування рекомендаційного списку з різних об'єктів (книг, фільмів, товарів та ін.). Рекомендаційна система забезпечує персоналізовану інформацію, вивчаючи інтереси користувача, дані профілю, а також аналізуючи його дії.

Метою розробки рекомендаційного модуля для шкільної електронної бібліотеки є можливість усунення проблеми відсутності та недоступності бажаних книг для користувача в даний момент, покращення якості навчання учнів шляхом швидкого та постійного доступу до книг, сприяння збільшенню мотивації та заохочення учнів до ознайомлення з бібліотекою, а також підвищення ступеню автоматизації процесу роботи шкільної бібліотеки.

Об'єктом дослідження є процес формування рекомендацій читачам шкільної бібліотеки щодо вибору літератури.

Предмет дослідження - методи надання рекомендацій для вибору літератури в електронній шкільній бібліотеці.

В першому розділі було проведено аналіз технологій обліку книжок в бібліотеках та методи надання рекомендацій щодо вибору літератури у електронній бібліотеці, описано предметну область, виконано аналіз та порівняння існуючих підходів до побудови рекомендаційних систем. Також було визначено вимоги до рекомендаційного модулю шкільної електронної бібліотеки.

В другому розділі визначено функціональні вимоги до рекомендаційного модулю шкільної електронної бібліотеки, побудовано діаграми функціонування та діаграми декомпозиції основних функцій модулю. Створено архітектуру рекомендаційного модулю, проведено розробку інформаційного забезпечення рекомендаційного модулю.

В третьому розділі описано вибір мови програмування застосунку, виконана розробка та опис програмного інтерфейсу з користувачем (керівництво користувача) розробленого модулю, виконана перевірка працездатності розробленого модуля.

В результаті виконання випускної кваліфікаційної роботи розроблено прототип програмного рекомендаційного модуля для шкільної електронної бібліотеки, який дозволить забезпечувати учнів постійним доступом до бібліотеки та надавати рекомендації по вибору нових книжок відповідно до запиту учня. Модуль може бути впроваджений до використання у шкільних бібліотеках.

РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ТЕХНОЛОГІЇ ОБЛІКУ КНИЖОК ТА НАДАННЯ РЕКОМЕНДАЦІЙ ЩОДО ВИБОРУ ЛІТЕРАТУРИ У ШКІЛЬНІЙ ЕЛЕКТРОННІЙ БІБЛІОТЕЦІ

1.1 Аналіз технології обліку книжок у шкільній бібліотеці

Бібліотека, або книгозбірня — культурно-освітній заклад, установа, котра виконує збір та зберігання друкованих і рукописних матеріалів, проводить їхнє опрацювання та внесення у каталоги, обслуговує читачів.

Шкільна бібліотека – це бібліотека, котра діє в структурі школи, та призначена для забезпечення учнів навчальною та художньою літературою. Шкільна бібліотека забезпечує організацію навчальної діяльності, дозвілля і розвитку учнів. Шкільна бібліотека в першу чергу складається з фонду підручників, фонду художньої літератури, а також навчально-виховної, методичної, довідково-енциклопедичної літератури. Основними засобами інформації є книги, довідкові та наукові видання, журнали, газети.

Сучасна шкільна бібліотека має поєднувати традиційні бібліотечні функції з застосуванням комп'ютерних технологій та доступом до інформаційних ресурсів.

Пріоритетні завдання в роботі шкільної бібліотеки:

- інформаційне забезпечення навчально-виховного процесу школи;
- формування культури читання в учнів;
- сприяння самоосвіті учнів;
- організація інформаційного обслуговування читачів;
- формування, використання, зберігання бібліотечного фонду;
- популяризація літератури та книги.

Основні види подання інформації у бібліотеках:

1. текстовий варіант книги;
2. відео матеріал;
3. аудіо варіант книги;



Рисунок 1.1 – Основні види подання інформації в бібліотеці

Організація роботи учнів у шкільній бібліотеці виконана наступним чином: кожен школяр має власний читацький квиток або електронний кабінет у бібліотеці. Учні забезпечуються навчальними матеріалами з фонду підручників, а також мають змогу в будь-який час звернутися до бібліотеки для пошуку необхідної додаткової або художньої літератури. Видача книжок може фіксуватися в читацькому квитку або особистому кабінеті. Якщо учень звертається до онлайн ресурсів бібліотеки та використовує електронні варіанти книжок, це не фіксується бібліотекою.

На даний момент існують альтернативні сайти онлайн бібліотек, такі як: “ЛитМир”, “MyBook”, YakaBoo.ua та інші. Всі платформи бібліотек мають одну й ту саму мету, а саме онлайн-бібліотека з книгами, які розподілені за жанрами, з доступом до зручного пошуку та перегляду книжок.

Важливим є доступ до повного переліку жанрів - історія, фантастика, наукова література, психологія, біографія, війна, фольклор та велика кількість інших, які існують сьогодні та цікаві користувачам. Всі онлайн бібліотеки мають свій каталог жанрів та пошукову систему, яка допоможе знайти необхідне.

Також є різні застосунки реалізації користуванням та засвоєнням інформації, в тому числі де користувачі з обмеженими можливостями, або ті, які з певних причин не можуть читати, повинні мати можливість слухати та засвоювати інформацію через аудіо-файли.

Порівняємо популярні існуючі електронні бібліотеки за різними критеріями у таблиці 1.1.

Таблиця 1.1 Порівняння існуючих сайтів електронних бібліотек

	“НЭБ”	“ЛитМир”	Yakaboo.ua	УкрЛіб
Різноманітний вибір за жанрами	+	+	+	-
Алгоритм рекомендацій	-	-	+	-
Зручний інтерфейс	-	-	+-	-
Адаптивність	+	+	+	+
Кроссбраузерний сайт	+	+	+	+

Виходячи з наведених порівнянь, більшість електронних бібліотек орієнтуються на різноманітність жанрів. Проте мало з них мають окрім цього зручний застосунок та рекомендаційну систему. Також, розглянуті сайти електронних бібліотек не спеціалізуються на шкільному матеріалі, тож саме учням важко знайти необхідні посібники у онлайн-доступі на таких сайтах. Виходячи з цього, розробка шкільної електронної бібліотеки з рекомендаційним модулем є надзвичайно актуальною на сьогоднішній день.

1.2 Цілі створення рекомендаційного модуля для шкільної електронної бібліотеки

На сьогоднішній день, в умовах світового карантину, воєнного стану та сучасних технологій - необхідно створити альтернативу фізичній шкільній бібліотеці, а саме бібліотеку в смартфоні або комп'ютері. Під час проведення онлайн-навчання в багатьох учнів немає можливості звернутися до бібліотеки в школі та отримати необхідні матеріали. Саме тому необхідно впровадити модуль електронної бібліотеки, аби учні незалежно від часу та місця перебування могли отримувати якісний онлайн-доступ до підручників, художньої літератури та всього фонду шкільної бібліотеки.

В той же час рекомендаційний модуль сприятиме розвитку читацької компетентності в учнів, підвищить рівень зацікавленості в новій літературі, оскільки буде зручним способом отримувати книжки цілодобово.

Завдання передбачає створення рекомендаційного модулю для шкільної електронної бібліотеки. Дана система повинна налагодити та автоматизувати процес самостійної, віддаленої роботи шкільної бібліотеки. В даній системі відсутній персонал, який видає книжки, тільки технічне обслуговування та збільшення каталогу книг адміністратором. Таким чином це значно спростить роботу працівникам бібліотек та заощадить їх час. Також система має

можливість заохочувати користувача читати книжки саме за тими напрямкам, які його цікавлять.

Основним мотивом створення даного додатку є бажання забезпечити учнів можливістю отримати книги для навчання та дозвілля через онлайн-доступ. На даний момент існує дуже мала кількість вітчизняного продукту на який не треба заходити через VPN-сервіси, підтримуючи цим іноземні компанії, а тим паче це стосується шкільних бібліотек. Нашим учням необхідна українська онлайн-бібліотека, котрою зможуть користуватися всі освітні заклади, і першим кроком до цього може стати моя рекомендаційна система для одної шкільної електронної бібліотеки.

1.3 Аналіз підходів до побудови рекомендаційних систем

1.3.1 Рекомендаційні системи: основні поняття та визначення

Рекомендаційна система – це програма, метою якої є фільтрація інформації та формування рекомендаційного списку з різних об'єктів (книг, фільмів, товарів, новин, відео, музики та ін.). Рекомендаційна система забезпечує персоналізовану інформацію, вивчаючи інтереси користувача, дані профілю, а також аналізуючи його дії. На основі цих даних формується граф інтересів і обчислюється список рекомендацій для конкретного користувача.

Рекомендаційні системи використовуються в багатьох сферах нашого життя, і шкільна освіта не є виключенням.

Перше застосування рекомендаційної системи було виконане сервісом Amazon [1]. Компанія Amazon охоплює більше 30 категорій товарів, у тому числі електронні книги, і рекомендує їх. Підґрунтям для рекомендацій є інформація про те, що клієнти купували, переглядали, поставлені відгуки та рейтинги. Обравши товар для купівлі, рекомендаційний модуль на основі цього

рекомендує користувачу інші продукти, котрі були придбані користувачами (за допомогою матриці покупки наступного товару на основі його схожості з попередньою покупкою). Компанія Amazon запатентувала цей підхід під назвою Item-To-Item Collaborative Filtering (колаборативна фільтрація від елемента до елемента). Метод спільної фільтрації (CF) збільшивши продажі Amazon та спровокував вибух досліджень рекомендаційних систем. [2]

На даний момент існує багато методів для створення рекомендацій, проте вони всі мають свої переваги і недоліки. Саме з цієї причини дослідження в даній області актуальні.

1.3.2 Підходи до побудови рекомендаційних систем

Існують основні стратегії створення рекомендаційних систем:

- Колаборативна фільтрація
- Фільтрація на основі вмісту
- Гібридна система фільтрації

1.3.2.1 Колаборативна фільтрація

Техніки колаборативної фільтрації найчастіше використовуються у соціальних мережах та медіа ресурсах, що пропонують цікаві спільні ресурси. Метою є оцінка індивідуальних преференції й пропозиція посилань на інші ресурси, системи, продукти, які вірогідно будуть цікаві користувачу.

Основна концепція колаборативної фільтрації полягає в тому, що двоє або більше людей, які мають подібні інтереси в одній області, також схильні до схожих товарів або продуктів з іншої області. Схожість між користувачами можна виявити за їх поведінкою в перегляді (показник кліків), шаблоном перегляду та оцінками (явні, неявні). Враховуючи поведінку інших

користувачів, колаборативна фільтрація оперує знаннями про групи, аби сформувані рекомендації на основі спільних рис. Наприклад, за допомогою колаборативної фільтрації музичний додаток може спрогнозувати, яка музика сподобається користувачу та підібрати схожі нові треки.

Прогнози складаються індивідуально для кожного користувача, хоч і використовується інформація зібрана від багатьох учасників.

Обмеження даної системи полягають в тому, що вона об'єднує користувачів в групи, котрим притаманний схожий стиль і пропонує їм одне й те саме, не враховуючи особливості кожного.

1.3.2.2 Фільтрування на основі вмісту

Системи фільтрування на основі вмісту засновані на концепції «Покажи мені більше того, що мені сподобалося». Основна ідея даних систем полягає в рекомендації предметів або продуктів конкретному користувачеві, схожих на ті, які вже сподобалися йому в минулому. Подібність між двома або кількома предметами можна обчислити на основі їх схожих ознак. [3]

Профілі взаємодії представлені векторами, які отримують характеризуючі атрибути об'єктів або користувачів. Обмеження даної системи полягає в тому, що рекомендації для всіх користувачів генеруються дуже одноманітні та не персоналізовані. Рекомендуються схожі між собою елементи на основі їх опису чи функцій.

1.3.2.3 Гібридна система фільтрації

Гібридна система фільтрації комбінує техніки колаборативної фільтрації та фільтрацію на основі вмісту. Існує багато варіантів гібридних рекомендаційних систем, але основними є:

- Колаборативна фільтрація з доданими можливостями взятими у техніки фільтрації на основі вмісту
- фільтрація на основі вмісту з використаними техніками колаборативної фільтрації.

Гібридну систему фільтрації використовують для подолання певних обмежень в методах колаборативної фільтрації та фільтрації на основі вмісту, шляхом поєднання кращих характеристик обох підходів. Поєднувати ці методи можна різноманітними способами. Використання гібридних моделей забезпечує більш точні рекомендації ніж чисті колаборативні методи та методи на основі вмісту.

1.3.3 Порівняння основних стратегій створення рекомендаційних систем

Рекомендаційні системи з колаборативною фільтрацією для своєї роботи аналізують дані про користувачів, а системи з фільтрацією на основі вмісту - дані про елементи системи. Системам, які мають повний опис метаданих елементів доцільніше використовувати фільтрацію на основі вмісту, а от системам з відсутніми або обмеженими даними - колаборативну фільтрацію.

Проблемою рекомендаційних систем є холодних старт. Холодний старт виникає при додаванні нового користувача або об'єкту до системи через відсутність інформації для формування рекомендацій.[4]

Нижче наведено таблицю з порівнянням методів фільтрації у сучасних рекомендаційних системах.

Таблиця 1.2 Порівняння методів фільтрації рекомендаційних систем

Тип фільтрації	Переваги	Недоліки
Колаборативна фільтрація	Забезпечує кращі результати якщо кількість користувачів і рейтингів доступна у великій кількості.	Проблема з холодним стартом у випадку як нових користувачів, так і нових елементів.
Фільтрування на основі вмісту	Не залежить від уподобань інших користувачів. Можливе порівняння між елементами.	Через користувачів із тисячами покупок або різноманітними смаками важко зробити оптимальні рекомендації. Холодний старт для нових користувачів.

Як видно з таблиці, найпоширенішою проблемою рекомендаційних систем є холодний старт, а фільтрування на основі вмісту більш гнучке та не прив'язане до оцінок користувача. Для нашого рекомендаційного модулю буде використано фільтрування на основі вмісту. Це дозволить формувати рекомендації, подаючи до списку схожі між собою елементи (на основі їх опису).

1.4. Вимоги до рекомендаційного модулю шкільної електронної бібліотеки

За допомогою рекомендаційної системи, що розробляється, процес пошуку нових книг, котрі задовольняють інтереси читача, значно спроститься:

необхідно буде лише зайти на сайт та вибрати з рекомендованого системою списку.

Головною задачею рекомендаційного модулю шкільної електронної бібліотеки є формування індивідуального додаткового списку літератури для кожного учня на основі його даних та інтересів. Даний модуль створює список пропонованої літератури учню на основі вхідної інформації про:

- клас;
- обраний предмет;
- обрану тему;

На виході ми отримуємо список підібраних рекомендацій для читання. З кожним новим пошуком або оновленням сторінки, список рекомендацій буде змінюватись. Список відображається середніми піктограмами з фотографією обкладинки книги, автором книги, кнопкою переходу до читання. Користувач також має можливість додати книгу до списку “улюблених”.

Рекомендації для учнів будуються за двома напрямками: навчальна література та художня література. Учень має змогу обрати з меню бажаний вид та виконати підбір рекомендацій по ньому.

Основна задача - розробка адаптивного та кросбраузерного сайту електронної шкільної бібліотеки з якісним рекомендаційним модулем. Головними критеріями сайту є сучасні матеріали, які можна замінити, додати або редагувати, використання всіх необхідних та доступних вимог для сучасної роботи з книгами.

Проаналізувавши наявний ринок електронних бібліотек, а також підходи до побудови рекомендаційних систем, було виявлено необхідні критерії для зручного користування сайтом електронної шкільної бібліотеки.

Основні характеристики системи:

- підтримка особистого кабінету;
- пошуковий модуль;
- перегляд книг;
- рекомендації нових книг за предметом, класом або темою;
- додавання книг до розділу “улюблені”;

Складові елементи модулю:

- підсистема авторизації;
- підсистема обліку читачів;
- підсистема обліку книг;
- рекомендаційний модуль;

Опис профілів зацікавлених сторін:

1. Внутрішні зацікавлені сторони:
 - розробник модулю;
 - власники та керівники школи;
 - учні;

2. Зовнішні зацікавлені сторони:
 - конкуренти;
 - партнери;
 - міністерство освіти і науки України;

Функціональні вимоги для сайту онлайн бібліотеки:

1. Авторизація на сайті

Для кожного учня створюється особистий кабінет в системі. В особистому кабінеті є можливість зберігання книг до розділу “улюблені”, доступ до формування рекомендацій, пошуку та перегляду книг.

2. Збереження інформації користувача

Для більш коректної роботи рекомендаційного модулю система буде записувати дані про улюблені книги користувача і надавати з них список улюблених жанрів.

3. Редактор сайту

Для редактора сайту необхідно мати всі інструменти додавання, редагування або видалення книги, додавання нових жанрів або піджанрів, перегляд списку користувачів, їх даних.

4. Зручна сторінка контенту

Сторінка повинна бути зручною, інтуїтивно зрозумілою та мати сучасний варіант перегляду книжок.

РОЗДІЛ 2. ПРОЕКТУВАННЯ РЕКОМЕНДАЦІЙНОГО МОДУЛЯ ДЛЯ ШКІЛЬНОЇ ЕЛЕКТРОННОЇ БІБЛІОТЕКИ

2.1 Функціональний аналіз рекомендаційного модуля для шкільної електронної бібліотеки

На діаграмі (рисунок 2.1). зображено контекстну діаграму. На вході діаграми ми маємо дані про користувача та дані про запит користувача, тобто що він шукає. Процес пошуку книг включає в себе пошук за назвою, класом, жанром та підбір рекомендацій для користувача. На виході ми отримуємо шукану книгу для відкриття та підібрані рекомендації для читання.

Елементами керування даного процесу є закон України “Про авторські права”, оскільки в доступі мають бути лише легальні книги. Другим елементом керування є закон України про захист персональних даних.

Механізмами у даному випадку є сайт бібліотеки та редактор сайту.



Рисунок 2.1 Контекстна діаграма

Розглянемо “дерево функцій” рекомендаційного модулю шкільної бібліотеки (рис.2.2). Головні функції даного модулю поділяються на два види: клієнтський модуль та адміністративний модуль.

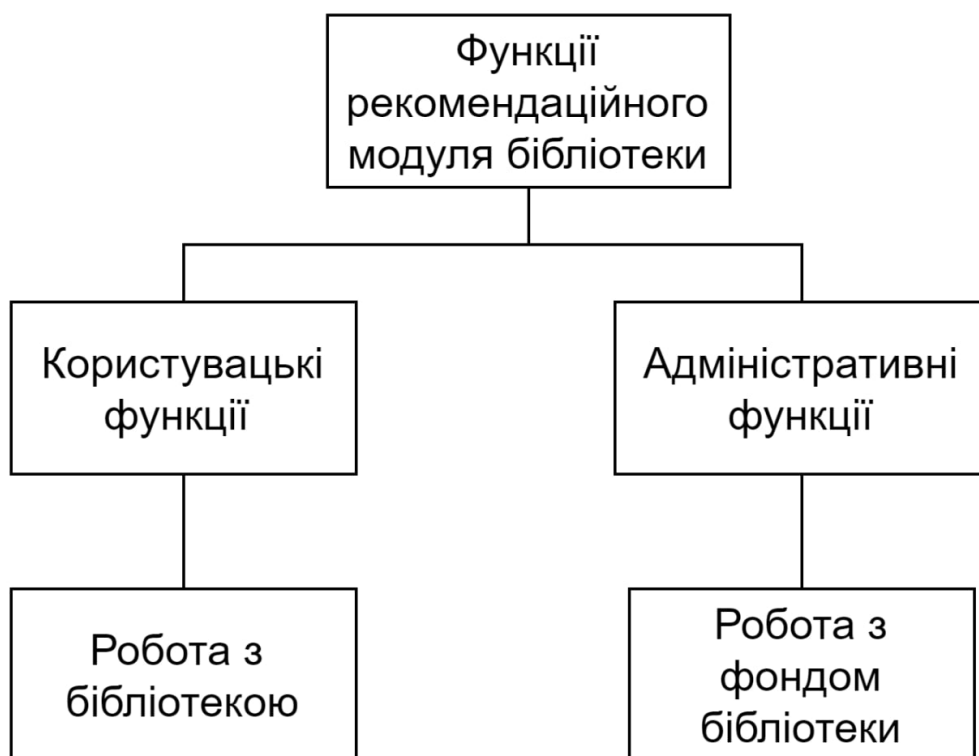


Рисунок 2.2 “Дерево функцій” модулю рекомендаційної системи бібліотеки

Адміністративний модуль є частиною розробки рекомендаційної системи, включає функції редагування модулю, підсистеми обліку книг, підсистеми обліку учнів, подальшої роботи з контентом: додавання нових книг до сайту електронної шкільної бібліотеки, редагування даних у вже існуючих книгах, видалення існуючих книжок з сайту.

Клієнтський модуль дає можливість використання рекомендаційної системи читачам, використовуючи інтерфейс системи.

Опишемо функціонування сайту шкільної електронної бібліотеки, а саме клієнтський функціонал (рис. 2.3). Початком запуску модулю є відкриття веб застосунку користувачем в браузері. Перед користувачем відкривається

можливість авторизації - вхід до особистого кабінету (логін). Дані про користувача зберігаються в підсистемі обліку читачів. Функція логіну доступна для вже зареєстрованих адміністратором користувачів. Після авторизації читачу відкривається можливість почати пошук книг та отримати необхідні запити.

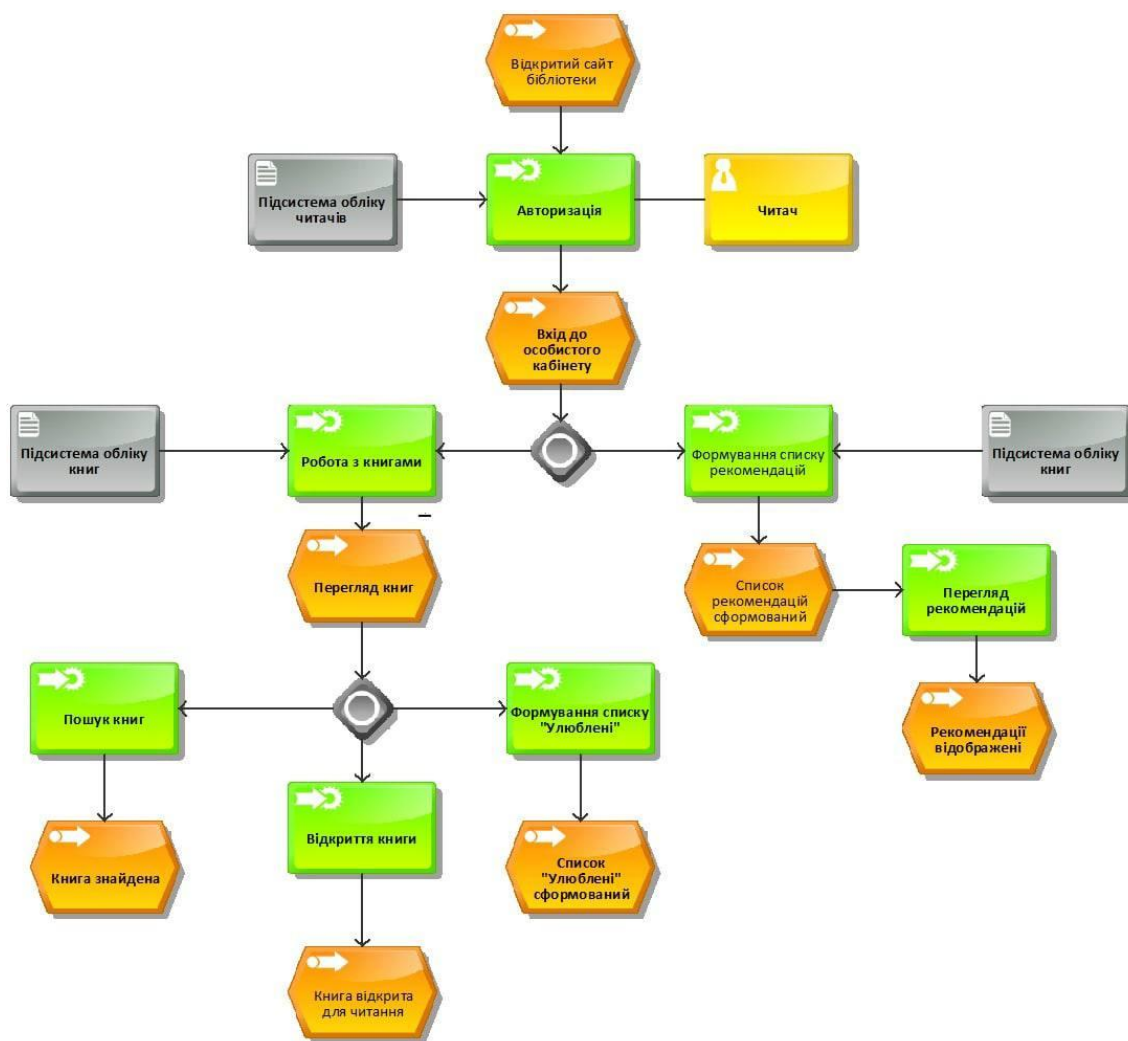


Рисунок 2.3 Діаграма Event-Driven Process Chain функціонування клієнтської частини модулю

Розглянемо діаграму функціонування модулю на рис.2.4. На найвищому рівні знаходиться процес “Робота з бібліотекою” котрий має три підпроцеси: “Авторизація на сайті” та “Користування бібліотекою”.

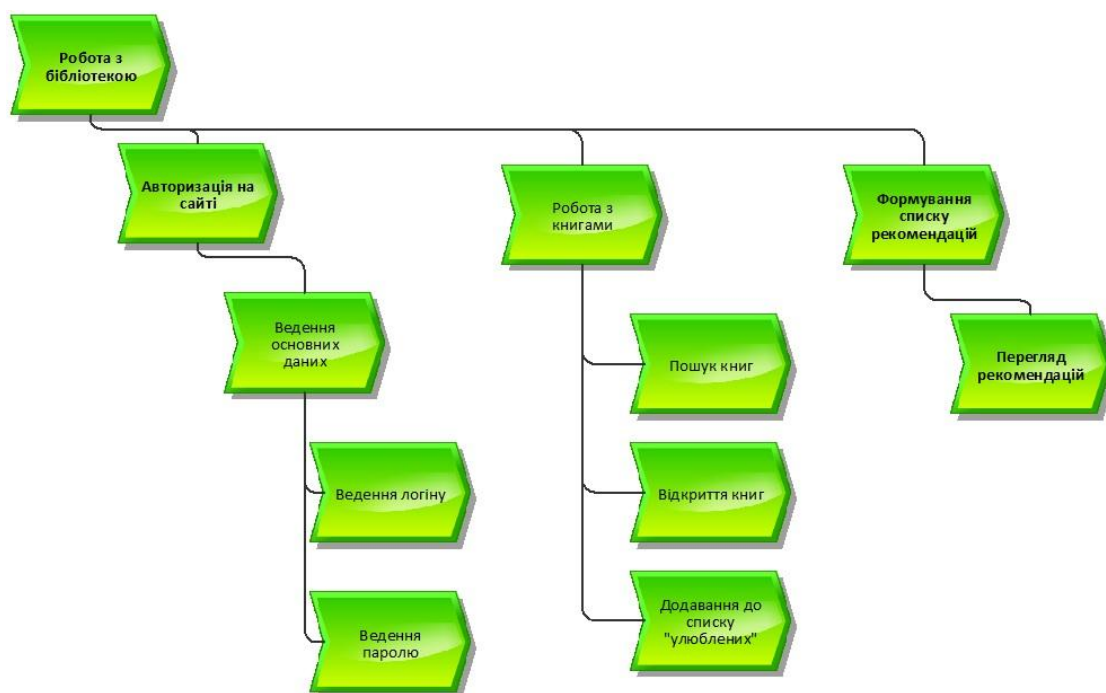


Рисунок 2.4 Діаграма Value Added Chain Diagram для функціонування клієнтської частини модулю

В авторизацію входить підпроцес введення основних даних, а саме введення логіну та паролю для особистого кабінету читача. Користування бібліотекою включає функціонал роботи з книжками - пошук книг (за назвою, видом літератури, класом), режим відкриття книжок та додавання до списку “улюблені”. Формування списку рекомендацій включає перегляд наданих рекомендацій.

Опишемо функціонування адміністративного модулю сайту на діаграмі (рис.2.5). На найвищому рівні знаходиться процес “Робота з фондом бібліотеки”, котрий має підпроцеси: “Додавання нових книг”, “Редагування існуючих книг” та “Видалення книг”.



Рисунок 2.5 Діаграма Value Added Chain Diagram для функціонування адміністративної частини модулю

2.2 Діаграми аналізу процесу формування рекомендацій

Представимо виконувані процеси у предметному середовищі за допомогою діаграм. Виконаємо декомпозицію основних процесів клієнтської частини модуля та зобразимо на діаграмі (рис.2.6).

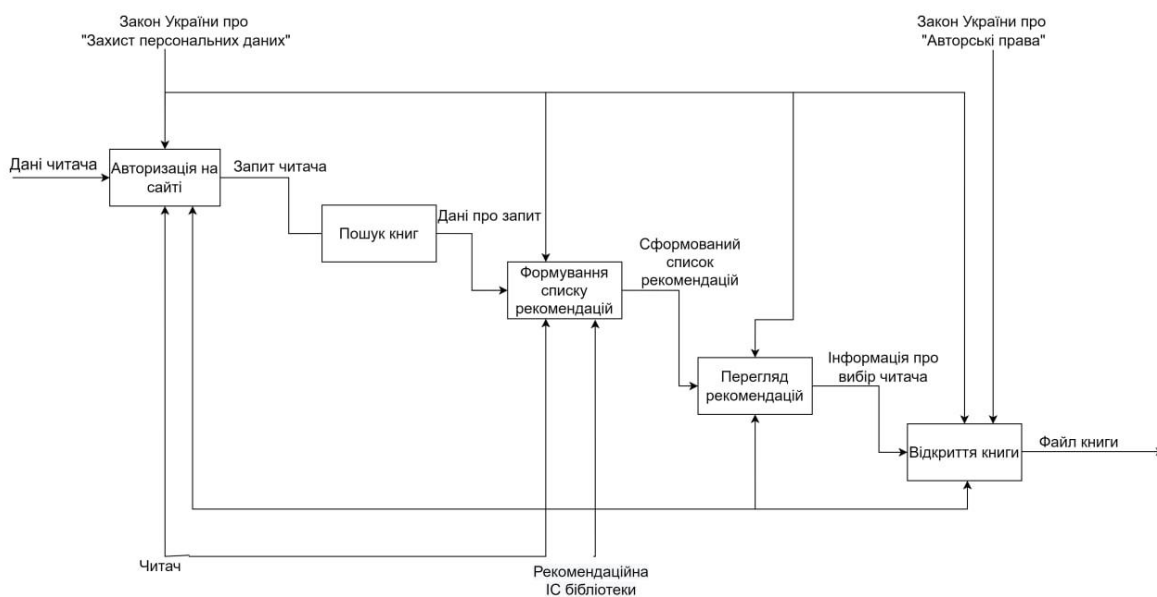


Рисунок 2.6 Декомпозиція основних процесів клієнтської частини модуля

Першим процесом є авторизація читача у системі, далі він задає запит на пошук необхідної книги, система отримує дані, проводить пошук з підсистеми обліку книг для читача і формує список рекомендацій, видає як результат

рекомендаційний список книг, які читач може переглянути і відкрити бажану книгу.

Виконаємо декомпозицію функції пошуку книг. Деталізована діаграма IDEF3 на рисунку 2.7 демонструє процес пошуку книг в підсистемі обліку книг.

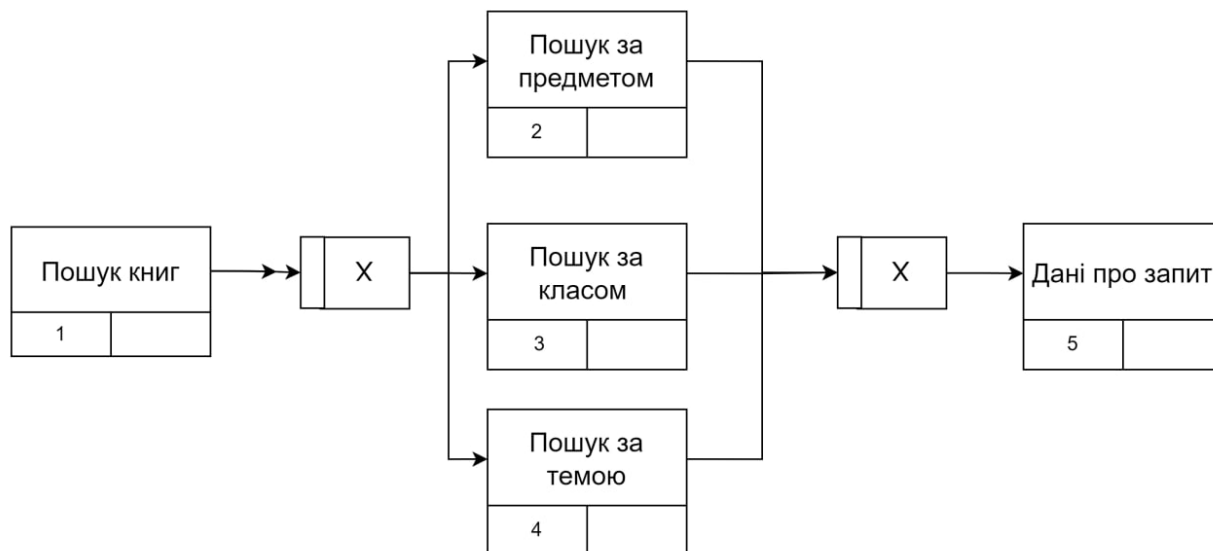


Рисунок 2.7 Діаграма IDEF3 для процесу пошуку книг

2.3 Архітектура інформаційної системи рекомендаційного модулю шкільної електронної бібліотеки

Наступний етап - побудова архітектури розроблюваної інформаційної системи.

Рекомендаційна система книг розподіляється на дві основних частини:

- модуль користувача;
- модуль адміністратора.

Користувацький модуль вміщує в собі підфункції:

- авторизація в особистий кабінет;
- робота з книгами;

- формування списку рекомендацій

Робота з книгами включає в себе: пошук книг, їх відкриття та роботу з функцією “додати до улюблених книг”. Формування списку рекомендацій включає в себе перегляд рекомендацій. Пошук книг та формування рекомендацій є основними функціями сайту. Знаходження книг відбувається за назвою, предметом, класом або темою. Формування рекомендацій - за класом учня, темою або предметом.

Ці функції дозволяють швидко та зручно знайти необхідну книгу для користувача.

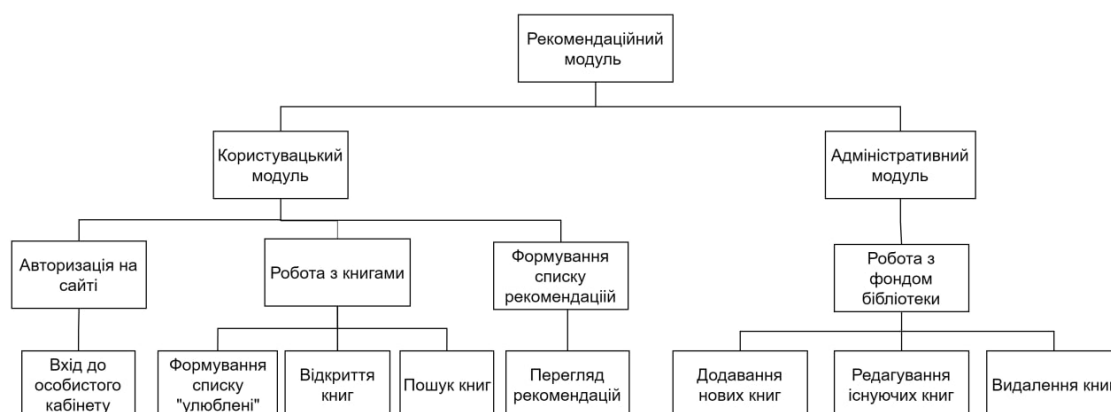


Рисунок 2.8 – Архітектура системи

Адміністративний модуль включає в собі роботу з фондом шкільної бібліотеки. Робота з фондом шкільної бібліотеки має підфункції:

- додавання нових книг;

Адміністратор має право додавати нові книги, що буде забезпечувати оновлення бібліотеки.

- редагування існуючих книг ;

Адміністратор має можливість змінювати існуючі книги з метою редагування людського фактору помилки.

- видалення книг;

Адміністратор має право видаляти книги, котрі були заборонені на території України або мають бути видалені з певних причин.

Формування рекомендацій у розроблювальній рекомендаційній системі відбуватиметься на основі даних про клас, предмет та тему. Обчислення відбуватиметься на основі схожих ознак книг, тобто фільтрування на основі вмісту. Таким чином користувач зможе виконати підбір особистих рекомендацій. Обравши улюблену книгу, він має можливість запросити підбір списку рекомендацій виходячи з побажань: схожі за класом, навчальним предметом або темою книги.

2.4 Інформаційне забезпечення рекомендаційного модуля електронної шкільної бібліотеки

Як вже було зазначено раніше, дана інформаційна система, для правильної роботи, використовує певні дані користувача.

Розробимо концептуальну модель рекомендаційного модуля (рис. 2.9). Для заданої предметної області відображено наступні сутності:

- Читач
- Книга

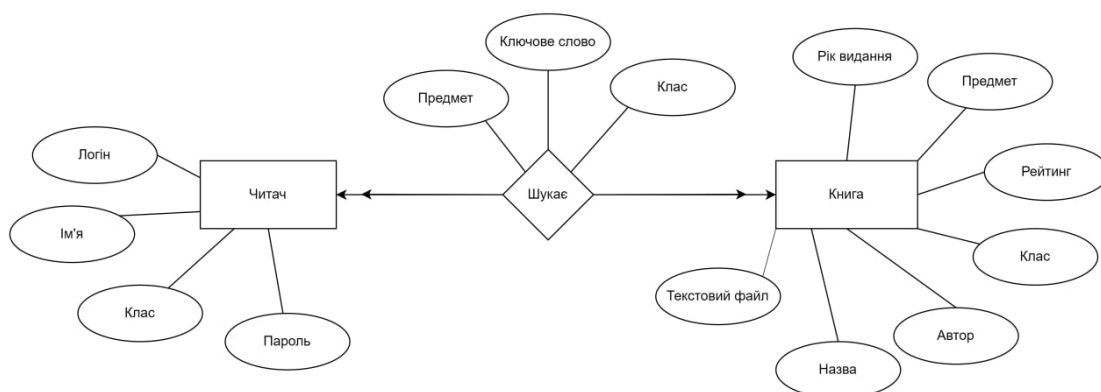


Рисунок 2.9 Концептуальна модель рекомендаційного модуля

Кожна з сутностей має свої атрибути, тобто майбутні поля у таблицях. Наприклад, «Читач» має логін, пароль, ім'я та клас, які вводяться адміністратором при реєстрації акаунту, а «Книга» містить в собі поля назви, автору, року видання, тощо.

Далі опишемо у таблиці 2.1 зв'язки між сутностями.

Таблиця 2.1 Зв'язки між сутностями

№	Сутності, що утворюють зв'язок	Тип зв'язку	Пояснення
1	Читач - Книга	Багато до багатьох	Одному читачу може рекомендуватись багато книг. Одна книга може рекомендуватись багатьом читачам.

Наступною розробимо логічну модель (рис. 2.10). Визначаємо первинні ключі та зовнішні ключі, які пов'язують таблиці між собою.



Рисунок 2.10 Логічна модель бази даних

Опишемо обмеження та типи даних атрибутів у таблиці.

Таблиця 2.2 Склад та характеристика атрибутів таблиць

п/п	Назва елемента даних	Тип елемента даних	Обов'язков е значення	Обмеження	Ключ
Таблиця «Читач»					
	ІД читача	Ціле число	так	30	ПК
	Пароль	Символьний	так		
	Логін	Символьний	так		
	ПІБ	Символьний	так	20	
	Клас	Ціле число	так	10	
Таблиця «Книга»					
	ІД книги	Ціле число	так	30	ПК
	Автор	Символьний	так		
	Назва	Символьний	так		
	Предмет	Символьний	так		
	Рік видання	Дата	так		
	Клас	Ціле число	так		
	Рейтинг	Ціле число	так	10	

Останньою є фізична модель бази даних (рис. 2.11), розроблена на основі даталогічної.



Рисунок 2.11 Фізична модель бази даних

Дана модель відображає як дані насправді зберігаються в базі даних. На ній можна побачити детальніші дані про атрибути, а саме – тип даних та обмеження для деяких з них.

2.3 Висновки до другого розділу

В результаті проведення етапу проектування рекомендаційного модуля електронної бібліотеки було спроектовано систему для подальшої її реалізації.

Було проведено функціональний аналіз, побудовано дерево функцій, архітектуру системи. Також було розроблено діаграми Event-Driven Process Chain та Value Added Chain Diagram для функціонування модуля користувача та адміністратора. На даних схемах було зображено та описано логіку роботи програми. Створена концептуальна, фізична модель та логічна модель рекомендаційного модулю.

РОЗДІЛ 3. РОЗРОБКА РЕКОМЕНДАЦІЙНОГО МОДУЛЯ ДЛЯ ШКІЛЬНОЇ ЕЛЕКТРОННОЇ БІБЛІОТЕКИ

3.1 Вибір мови розробки

Для розробки веб сторінок використовується JavaScript. JavaScript — динамічна, об'єктно-орієнтована мова програмування. Найчастіше використовується як частина браузера, що надає можливість коду на стороні клієнта (такому, що виконується на пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки, проте його також використовують багато небраузерних середовищ, наприклад node.js та Apache CouchDB. JavaScript є прототип-орієнтованою, динамічною скриптовою мовою з декількома парадигмами із підтримкою об'єктно-орієнтованого, імперативного й декларативного (функціональне програмування) стилів. Це третій шар в ієрархії стандартних веб-технологій, два з яких HTML і CSS.

HTML — це мова розмітки, яку ми використовуємо для структурування та надання сенсу до нашого веб-вмісту, наприклад, визначення абзаців, заголовків і таблиць даних або вбудовування зображень на сторінку.

CSS — це мова правил стилю, яку ми використовуємо для застосування стилів до нашого HTML-вмісту, наприклад, встановлення кольорів фону та шрифтів, а також розміщення нашого вмісту в кількох стовпцях.

Представлення HTML-документа виконуємо у вигляді дерева тегів за допомогою об'єктна модель документа (Document Object Model). Таке дерево потрібно для правильного відображення сайту та внесення змін на сторінках за допомогою JavaScript. DOM – це "дерево вузлів", що має єдиний корінь, який розділяється на безліч дочірніх гілок, кожна з яких теж може розділитися і закінчується "листя". Корінь – це елемент html, а гілки – вкладені елементи.

3.2 Розробка та опис програмного інтерфейсу з користувачем (керівництво користувача) рекомендаційного модуля

Інтерфейс з користувачем рекомендаційного модулю для шкільної бібліотеки реалізується як багатосторінковий веб-додаток.

На початку роботи користувач першим кроком має авторизуватися за допомогою логіну та паролю.

Рисунок 3.1 Авторизація на початковій сторінці рекомендаційного модуля

Після того, як користувач авторизувався, він потрапляє на головну сторінку, де відображено наявні книги бібліотеки в довільному порядку (рис.3.2).

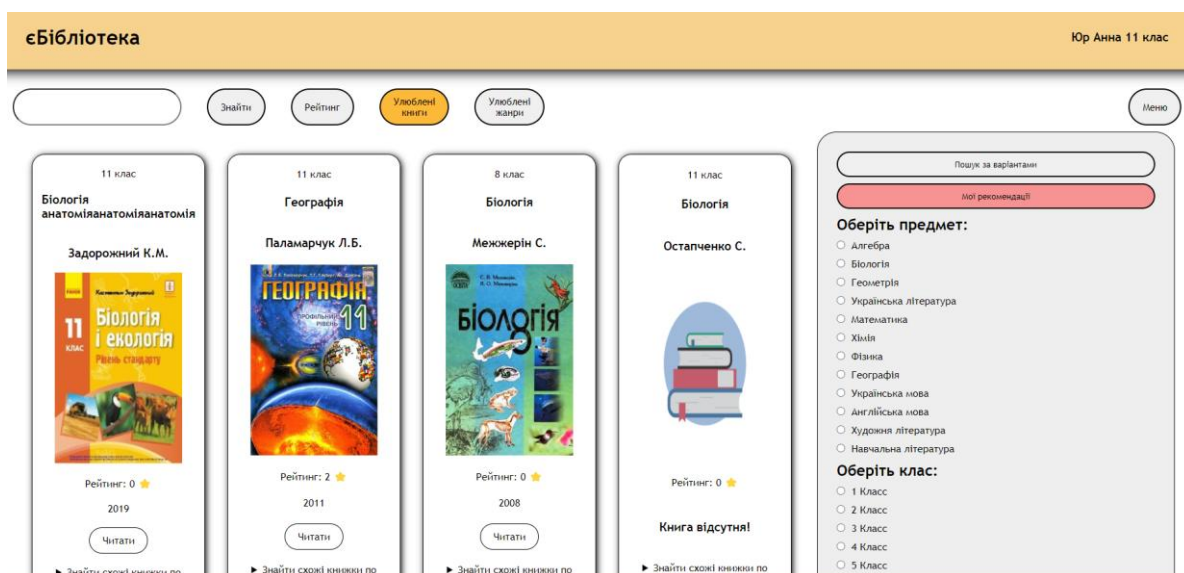


Рисунок 3.2 Головна сторінка рекомендаційного модуля

На головній сторінці користувач має можливість вибору з переліку функцій, а саме (рис.3.3):

1. Виконати пошук книг за назвою.

2. Переглянути шкільний рейтинг книг.
3. Переглянути власний список “улюблених книг”.
4. Відобразити власний список “улюблених жанрів”.

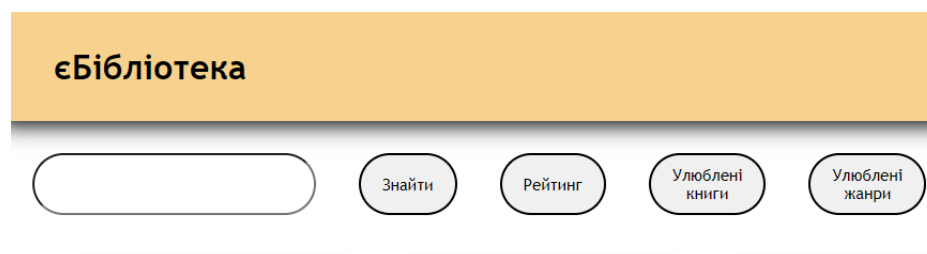


Рисунок 3.3 Функції головного меню рекомендаційного модуля

Також на головній сторінці користувач має змогу розгорнути вкладку “Меню”. В даній вкладці користувач отримує наступний функціонал (рис.3.4):

1. Пошук за варіантами:
 - обравши предмет;
 - обравши клас;
 - обравши те, що не цікавить користувача;
 - обравши декілька підпунктів;
2. Формування “Моїх рекомендацій” .

Формування списку “Мої рекомендації” відбувається з використанням даних акаунту про клас та використовуючи шкільний рейтинг книг.

Пошук за варіантами

Мої рекомендації

Оберіть предмет:

- Алгебра
- Біологія
- Геометрія
- Українська література
- Математика
- Хімія
- Фізика
- Географія
- Українська мова
- Англійська мова
- Художня література
- Навчальна література

Оберіть клас:

- 1 Клас
- 2 Клас
- 3 Клас
- 4 Клас
- 5 Клас
- 6 Клас
- 7 Клас
- 8 Клас
- 9 Клас
- 10 Клас
- 11 Клас

Не цікавить

- Алгебра

Рисунок 3.4 Робота з меню пошуку книг

Обравши бажаним методом книгу, користувачу надається наступний функціонал (рис.3.5):

- відкрити для читання;
- додати книгу до списку “улюблені книги”;
- знайти схожі книги по: предмету/класу/темі;

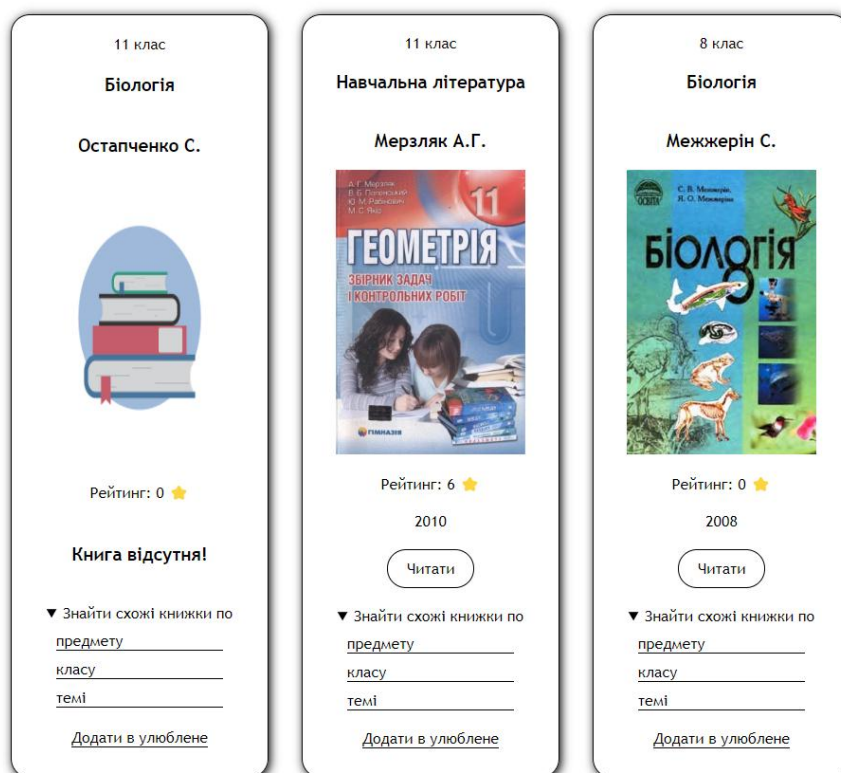


Рисунок 3.5 Робота з книгою

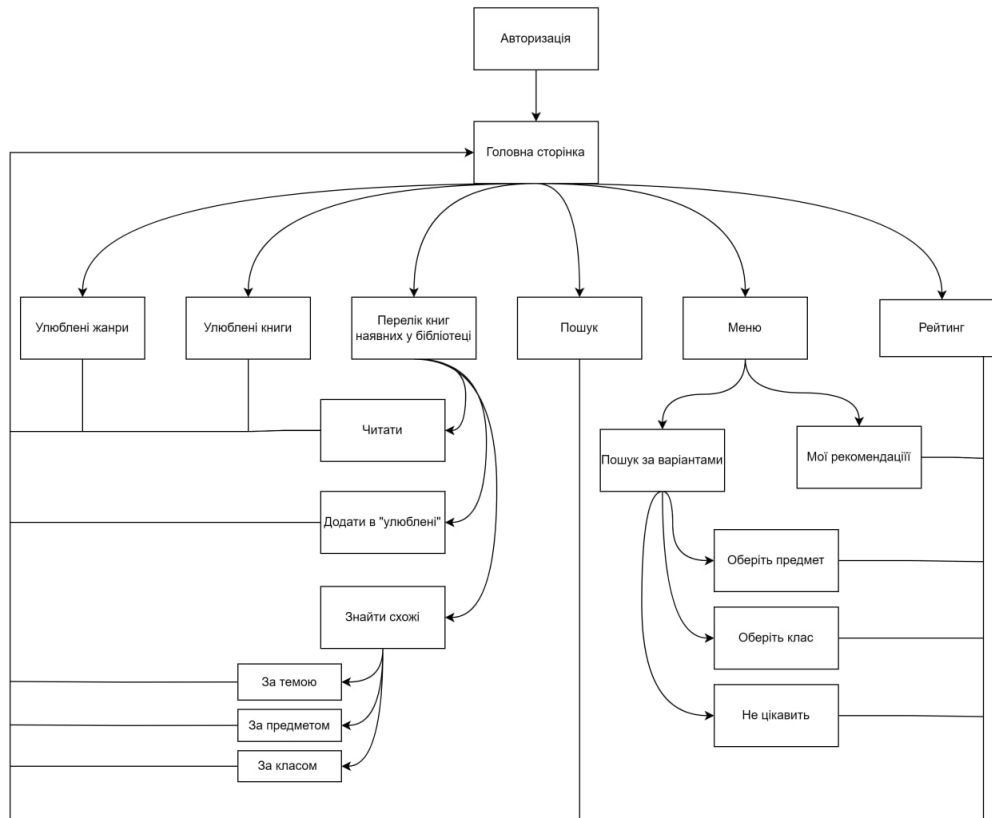


Рисунок 3.6 Граф переходів між сторінками рекомендаційного модуля

3.3 Перевірка працездатності програмного рекомендаційного модуля

Перевіримо працездатність рекомендаційного модуля. Обравши у меню бажані критерії, виконаємо пошук: клас - 11, предмет - художня література, отримуємо результат (рис.3.7).

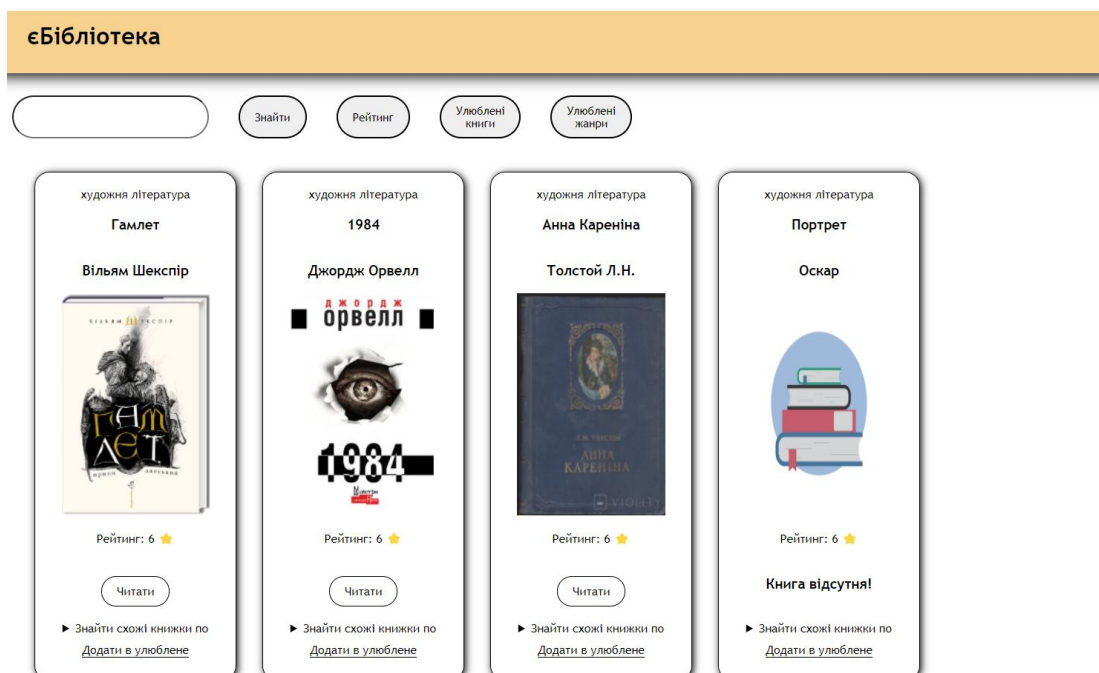


Рисунок 3.7 Результат за пошуком

Додамо книгу до списку “Улюблені книги”. Обираємо бажану книгу з переліку, до прикладу “Портрет Доріана Грея” та додаємо до улюблених. Тепер дану книгу ми можемо швидко знайти у розділі “Улюблені”. Для наочності порівняємо рисунок 3.8., де список улюблених книг пустий, та рисунок 3.9., зроблений після додавання нами необхідної книги.

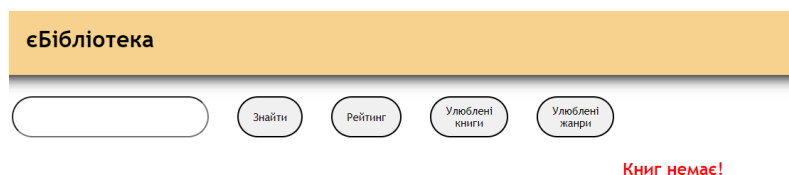


Рисунок 3.8 Список “улюблені книги” до виконання додавання

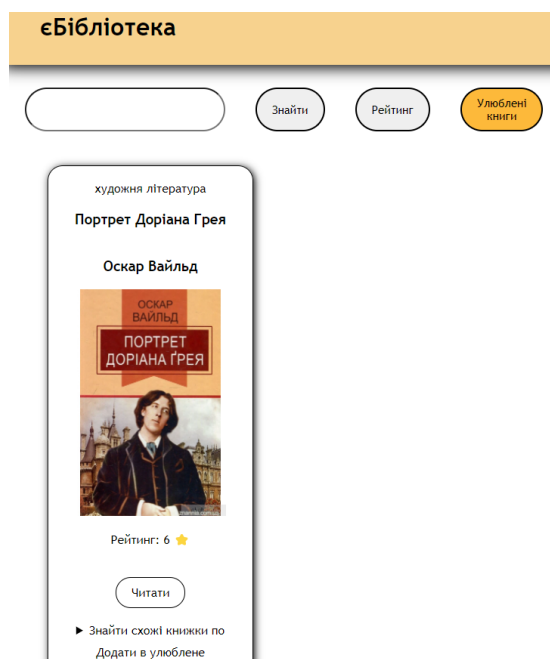


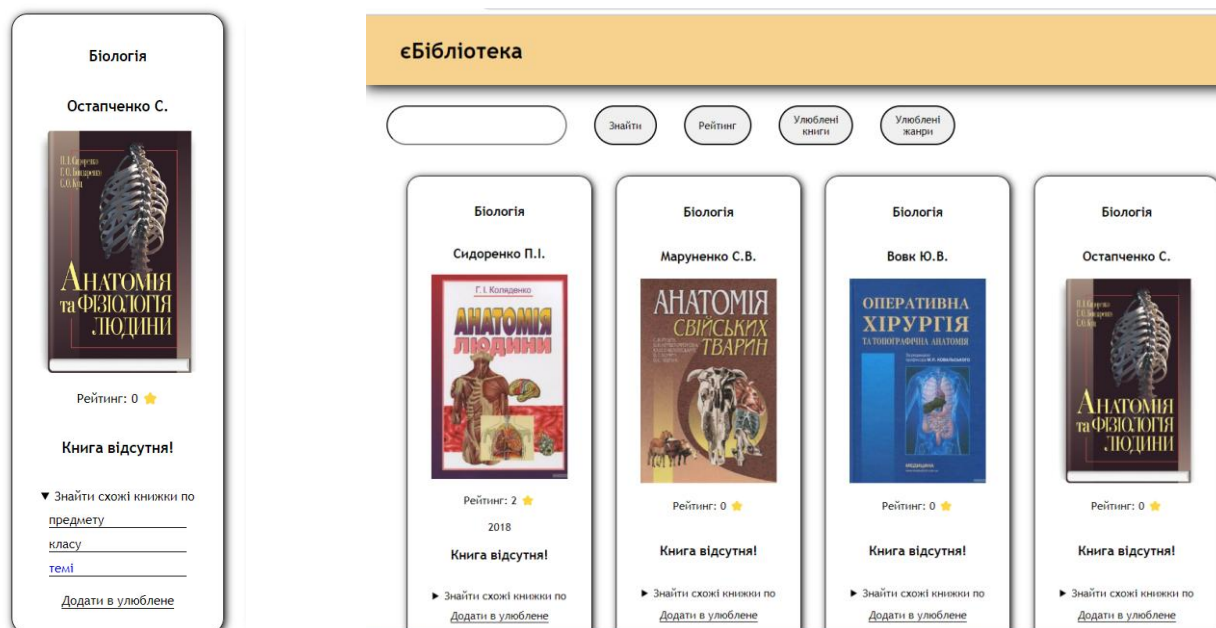
Рисунок 3.9 Список “улюблені книги” після виконання додавання

Перевіримо формування рекомендацій за допомогою функції “Знайти схожі книги по предмету/класу/темі”. Обираємо книгу з біології та виконаємо підбір рекомендацій на основі предмету (рис.3.10).



Рисунок 3.10 - (а) Обрана книга для виконання підбору рекомендацій;
(б) Результат виконання підбору рекомендацій схожих книг за предметом.

Обираємо знову книгу з біології та виконаємо підбір рекомендацій на основі теми. Специфікою даної книги є те, що вона має вузьке направлення - “анатомія”, тож отримати ми маємо рекомендації не загальних підручників з біології, а саме книг з анатомії (рис.3.11).



(a)

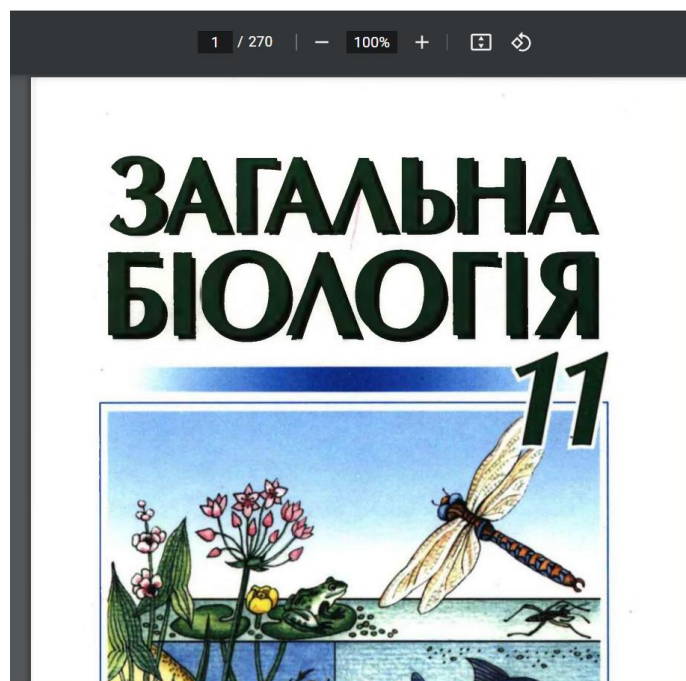
(б)

Рисунок 3.11 - (а) Обрана книга для виконання підбору рекомендацій;
(б) Результат виконання підбору рекомендацій схожих книг за темою.

Також перевіримо працездатність відкриття самих книжок (рис.3.12).



(a)



(б)

Рисунок 3.12 - (а) Обрана книга для прочитання

(б) Відкриття книги для прочитання

ВИСНОВКИ

Модуль рекомендацій шкільної бібліотеки не тільки зменшує інформаційне навантаження при пошуку навчальних матеріалів, рекомендуючи відповідну літературу для учнів, а також надає їм варіанти для поглибленого вивчення предметів та позашкільного літературного розвитку. В дипломній роботі було проаналізовано існуючі методи рекомендацій, розглянуто використання рекомендаційних методів в існуючих системах та представлено модуль рекомендацій. Було розглянуто підхід до розробки рекомендаційної системи, метою якої є забезпечення учнів навчальними та додатковими матеріалами для покращення рівня знань та заохочення до читання більшої кількості книг.

Загалом рекомендаційна система шкільної бібліотеки має високі перспективи, оскільки кожен користувач, в нашому випадку учень, має індивідуальні запити, потреби та конкретні вимоги. Вмотивовані учні прагнуть відкривати для себе нову літературу та поглиблено вивчати той чи інший напрям. З щоденним ростом комп'ютерних та інтернет технологій багато шкіл використовують або прагнуть почати використовувати систему електронної бібліотеки для підтримки викладання та навчання. Система поширює різні навчальні матеріали, такі як посібники, авторські статті викладачів, додаткову літературу, тощо. І для ефективного користування даними матеріалами, впровадження підтримки персоналізованих рекомендацій є найкращим варіантом.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Qomariyah, N. N. Pairwise Preferences Learning for Recommender Systems (Doctoral dissertation, University of York). 2018. С. 7-11.
2. Types of recommendation systems: [Електронний ресурс] // Режим доступу: <https://www.victoria.lviv.ua/library/students/sss/theme9.html>
3. Evolution of Recommender Systems from Ancient Times to Modern Era: A Survey Richa Sharma* and Rahul Singh University Institute of Engineering, Chandigarh University, Gharuan, Mohali - 140413, Punjab, India; Vol 9 (20) | May 2016. С 2-5.
4. Системи управління, навігації та зв'язку, 2018, випуск 4(50) // Є.В. Мелешко // Центральноукраїнський національний технічний університет, Кропивницький, Україна // ПРОБЛЕМИ СУЧАСНИХ РЕКОМЕНДАЦІЙНИХ СИСТЕМ ТА МЕТОДИ ЇХ РІШЕННЯ. С 120-121.
5. G., Linden, B., Smith, and J., York, “Amazon.com recommendations: item-to-item collaborative filtering,” Internet Computing 7:1, pp. 76–80, 2003.
6. Jure Leskovec, Anand Rajaraman, Jeffrey David Ullman, “Mining of Massive Datasets”, Chapter 9 Recommendation Systems, Published online by Cambridge University Press, pp. 319-353, 2020.
7. Електронна бібліотека Yakaboo: [Електронний ресурс] // Режим доступу: <https://www.yakaboo.ua/>
8. Електронна бібліотека української літератури: [Електронний ресурс] // Режим доступу: <https://www.ukrlib.com.ua/>
9. Електронна бібліотека ЛітМір: [Електронний ресурс] // Режим доступу: <https://www.litmir.me/>
10. Електронна бібліотека “Національна електронна бібліотека”: [Електронний ресурс] // Режим доступу: <https://rusneb.ru/>
11. Електронна бібліотека “Національна електронна бібліотека”: [Електронний ресурс] // Режим доступу: <https://html5book.ru/>

12. HTML & CSS design and build websites // Jon Duckett: [Електронний ресурс] // Режим доступу: <https://wtf.tw/ref/duckett.pdf>
13. JavaScript & JQuery: Interactive Front-End Web Development // Jon Duckett
14. Mastering JavaScript Functional Programming.: Write clean, robust, and maintainable web and server code using functional JavaScript, 2nd Edition // Federico Kereki. 2020.
15. Recommendation systems: Principles, methods and evaluation // F.O.Isinkaye, Y.O.Folajimi, B.A.Ojokoh // Egyptian Informatics Journal Volume 16, Issue 3, November 2015, Pages 261-273.
16. Том 4 № 50 (2018): Системи управління, навігації та зв'язку // Полтавський національний технічний університет імені Юрія Кондратюка // С. 120-125.

ДОДАТКИ

Додаток А

Програмний код index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <link href="style/main.css" rel="stylesheet">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <div class="root">
    </div>
</body>
<script src="script/script.js"></script>
</html>
```

Програмний код script.js

```

// Користувачі
var user = [
  {
    name: "Анна",
    lastName: "Юр",
    login: "1",
    password: "1",
    dateLogin: "",
    date: "",
    class: 11,
  },
  {
    name: "Поліна",
    lastName: "Станченко",
    login: "2",
    password: "2",
    dateLogin: "",
    date: "",
    class: 10,
  },
]

var userId;
var url = "img/"
var pdf = "pdf/"

// Предмети
var lessonName = [
  { ukr: "Алгебра", eng: "algebra" },
  { ukr: "Біологія", eng: "biology" },
  { ukr: "Геометрія", eng: "geometry" },
  { ukr: "Українська література", eng: "ukrainianLiterature" },
  { ukr: "Математика", eng: "mathematics" },
  { ukr: "Хімія", eng: "chemistry" },
  { ukr: "Фізика", eng: "physics" },
  { ukr: "Географія", eng: "geography" },
  { ukr: "Українська мова", eng: "ukrainianLanguage" },
  { ukr: "Англійська мова", eng: "englishLanguage" },
  { ukr: "Художня література", eng: "fiction" },
  { ukr: "Навчальна література", eng: "educationalLiterature" },
]

var myFavoriteBooks = [

]

var myFavoriteGenre = []

// БД
const libraryBook = [
  {
    id: 0,
    name: "Біологія",
    grade: "11 клас",
    lessonTitle: "biology", //biology, algebra, geometry, mathematics, chemistry, geography,
    ukrainianLanguage, englishLanguage
    author: "Задорожний К.М.",
    class: 11,
    img: url + "bio11.jpg",
  }
]

```

```

        pdf: pdf + "bio11.pdf",
        year: "2019",
        rating: 0,
        topic: "загальна",
    },
    {
        id: 1,
        name: "Географія",
        grade: "11 клас",
        lessonTitle: "geography", //biology, algebra, geometry, mathematics, chemistry, geography,
        ukrainianLanguage, englishLanguage
        author: "Паламарчук Л.Б.",
        class: 11,
        img: url + "geo11.jpg",
        pdf: pdf + "geo11.pdf",
        year: "2011",
        rating: 2,
        topic: "",
    },
    {
        id: 2,
        name: "Геометрія",
        grade: "11 клас",
        lessonTitle: "geometry", //biology, algebra, geometry, mathematics, chemistry, geography,
        ukrainianLanguage, englishLanguage
        author: "Мерзляк А.Г.",
        class: 11,
        img: url + "geom11.jpg",
        pdf: pdf + "geom11.pdf",
        year: "2019",
        rating: 4,
        topic: "",
    },
    {
        id: 3,
        name: "Українська мова",
        grade: "11 клас",
        lessonTitle: "ukrainianLanguage", //biology, algebra, geometry, mathematics, chemistry, geography,
        ukrainianLanguage, englishLanguage
        author: "Бондаренко Н.В.",
        class: 11,
        img: url + "ukr11.jpg",
        pdf: pdf + "ukr11.pdf",
        year: "2011",
        rating: 6,
        topic: "",
    },
    {
        id: 4,
        name: "Навчальна література",
        grade: "11 клас",
        lessonTitle: "educationalLiterature", //biology, algebra, geometry, mathematics, chemistry,
        geography, ukrainianLanguage, englishLanguage
        author: "Мерзляк А.Г.",
        class: 11,
        img: url + "geom11dop.jpg",
        pdf: pdf + "geom11dop.pdf",
        year: "2010",
        rating: 6,
        topic: "",
    },
    {

```

```

        id: 6,
        name: "Гамлет",
        grade: "художня література",
        lessonTitle: "fiction", //biology, algebra, geometry, mathematics, chemistry, geography,
ukrainianLanguage, englishLanguage
        author: "Вільям Шекспір",
        class: 11,
        img: url + "hamlet.jpg",
        pdf: pdf + "hamlet.pdf",
        year: "",
        rating: 6,
        topic: "",
    },
    {
        id: 7,
        name: "Портрет Доріана Грея",
        grade: "художня література",
        lessonTitle: "fiction", //biology, algebra, geometry, mathematics, chemistry, geography,
ukrainianLanguage, englishLanguage
        author: "Оскар Вайльд",
        class: 11,
        img: url + "doriangrey.jpg",
        pdf: pdf + "doriangrey.pdf",
        year: "",
        rating: 6,
        topic: "",
    },
    {
        id: 9,
        name: "1984",
        grade: "художня література",
        lessonTitle: "fiction", //biology, algebra, geometry, mathematics, chemistry, geography,
ukrainianLanguage, englishLanguage
        author: "Джордж Орвелл",
        class: 11,
        img: url + "1984.jpg",
        pdf: pdf + "1984.pdf",
        year: "",
        rating: 6,
        topic: "",
    },
    {
        id: 10,
        name: "451 градус за Фаренгейтом",
        grade: "художня література",
        lessonTitle: "fiction", //biology, algebra, geometry, mathematics, chemistry, geography,
ukrainianLanguage, englishLanguage
        author: "Рей Бредбері",
        class: 11,
        img: url + "451.jpg",
        pdf: pdf + "451.pdf",
        year: "",
        rating: 6,
        topic: "",
    },
    {
        id: 14,
        name: "Біологія",
        grade: "11 клас",
        lessonTitle: "biology", //biology, algebra, geometry, mathematics, chemistry, geography,
ukrainianLanguage, englishLanguage
        author: "Соболь В.",
    }

```

```

class: 11,
img: url + "bio11_2.jpg",
pdf: pdf + "bio11_2.pdf",
year: "2019",
rating: 5,
topic: "загальна",
},
{
id: 15,
name: "Біологія",
grade: "11 клас",
lessonTitle: "biology", //biology, algebra, geometry, mathematics, chemistry, geography,
ukrainianLanguage, englishLanguage
author: "Кучеренко М.",
class: 11,
img: url + "bio11_3.jpg",
pdf: pdf + "bio11_3.pdf",
year: "2006",
rating: 2,
topic: "загальна",
},
{
id: 16,
name: "Біологія",
grade: "8 клас",
lessonTitle: "biology", //biology, algebra, geometry, mathematics, chemistry, geography,
ukrainianLanguage, englishLanguage
author: "Магяш М.",
class: 8,
img: url + "bio8.jpg",
pdf: pdf + "bio8.pdf",
year: "2006",
rating: 2,
topic: "загальна",
},
{
id: 17,
name: "Біологія",
grade: "8 клас",
lessonTitle: "biology", //biology, algebra, geometry, mathematics, chemistry, geography,
ukrainianLanguage, englishLanguage
author: "Межжерін С.",
class: 8,
img: url + "bio8_2.jpg",
pdf: pdf + "bio8_2.pdf",
year: "2008",
rating: 0,
topic: "загальна",
},
{
id: 18,
name: "Біологія",
grade: "11 клас",
lessonTitle: "biology", //biology, algebra, geometry, mathematics, chemistry, geography,
ukrainianLanguage, englishLanguage
author: "Остапченко С.",
class: 11,
img: "",
pdf: "",
year: "",
rating: 0,
topic: "загальна",

```

```

    },
    {
        id: 19,
        name: "Біологія",
        grade: "11 клас",
        lessonTitle: "biology", //biology, algebra, geometry, mathematics, chemistry, geography,
ukrainianLanguage, englishLanguage
        author: "Остапченко С.",
        class: 11,
        img: "",
        pdf: "",
        year: "",
        rating: 0,
        topic: "загальна",
    },
    {
        id: 20,
        name: "Біологія",
        grade: "",
        lessonTitle: "biology", //biology, algebra, geometry, mathematics, chemistry, geography,
ukrainianLanguage, englishLanguage
        author: "Сидоренко П.І.",
        class: "",
        img: url + "anat.jpg",
        pdf: "",
        year: "2018",
        rating: 2,
        topic: "анатомія",
    },
    {
        id: 21,
        name: "Біологія",
        grade: "",
        lessonTitle: "biology", //biology, algebra, geometry, mathematics, chemistry, geography,
ukrainianLanguage, englishLanguage
        author: "Остапченко С.",
        class: "",
        img: url + "anato.jpg",
        pdf: "",
        year: "",
        rating: 0,
        topic: "анатомія",
    },
    {
        id: 22,
        name: "Біологія",
        grade: "",
        lessonTitle: "biology", //biology, algebra, geometry, mathematics, chemistry, geography,
ukrainianLanguage, englishLanguage
        author: "Маруненко С.В.",
        class: "",
        img: url + "pic.jpg",
        pdf: "",
        year: "",
        rating: 0,
        topic: "анатомія",
    },
    {
        id: 23,
        name: "Біологія",
        grade: "",

```

```

        lessonTitle: "biology", //biology, algebra, geometry, mathematics, chemistry, geography,
        ukrainianLanguage, englishLanguage
        author: "Вовк Ю.В.",
        class: "",
        img: url + "ann.jpg",
        pdf: "",
        year: "",
        rating: 0,
        topic: "анатомія",
    },
]

//-----//

```

```

const root = document.querySelector(".root")

// Создаем хедер, верхний блок с инфой о юзере
function header(user) {
    const header = document.createElement("div")
    header.className = "header"

    const logo = document.createElement("h1")
    logo.className = "logo"
    logo.innerHTML = "єБібліотека"
    logo.setAttribute("onclick", "books(librayBook)")

    const userName = document.createElement("h3")
    userName.innerHTML = `${user.lastName} ${user.name} ${user.class} клас`

    header.append(logo)
    header.append(userName)

    root.append(header)
}

// Поиск книги по названию
function searchBook(librayBook) {
    const inputSearch = document.querySelector(".inputSearch").value

    if (inputSearch) {
        let newArrLibray = []
        for (let i = 0; i < librayBook.length; i++) {
            if (inputSearch == librayBook[i].name) {
                newArrLibray[i] = librayBook[i]
            }
        }
        newArrLibray = newArrLibray.filter(Boolean);
        books(newArrLibray, "rating")
    } else books(librayBook)
}

// Создаем поле для ввода поиска по названию и кнопка для всплывающего меню
function buttonFunction() {
    const searchDiv = document.createElement("div")
    searchDiv.className = "search"

    const search = document.createElement("input")
    search.className = "inputSearch"
}

```

```

const searchButton = document.createElement("button")
searchButton.innerHTML = "Знайти"
searchButton.setAttribute("onclick", "searchBook(librayBook)")

const searchButtonRating = document.createElement("button")
searchButtonRating.innerHTML = "Рейтинг"
searchButtonRating.setAttribute("onclick", "books(librayBook, 'rating')")

const myBooks = document.createElement("button")
myBooks.innerHTML = "Улюблені книги"
myBooks.setAttribute("onclick", "books(myFavoriteBooks, 'rating')")

const myGenre = document.createElement("button")
myGenre.innerHTML = "Улюблені жанри"
myGenre.setAttribute("onclick", "genre()")

const searchMenuButton = document.createElement("button")
searchMenuButton.innerHTML = "Меню"
searchMenuButton.setAttribute("onclick", "menu(lessonName)")

searchDiv.append(search)
searchDiv.append(searchButton)
searchDiv.append(searchButtonRating)
searchDiv.append(myBooks)
searchDiv.append(myGenre)
searchDiv.append(searchMenuButton)

root.append(searchDiv)
}

// Создаем само меню
function menu(lessonName) {

const searchMenuDiv = document.querySelector(".menu")

if (!searchMenuDiv) {
    const menu = document.createElement("div")
    menu.className = "menu"
    root.append(menu)

    const menuButton = document.createElement("button")
    menuButton.innerHTML = "Пошук за варіантами"
    menuButton.className = "searchMenu"
    menuButton.setAttribute("onclick", "options(librayBook, `button`)")
    menu.append(menuButton)

    const recommendationButton = document.createElement("button")
    recommendationButton.innerHTML = "Мої рекомендації"
    recommendationButton.className = "recommendationMenu"
    recommendationButton.setAttribute("onclick", "optionsUser(librayBook, user)")
    menu.append(recommendationButton)

    const titleYesLessonsMenu = document.createElement("h2")
    titleYesLessonsMenu.innerHTML = "Оберіть предмет:"

    menu.append(titleYesLessonsMenu)

    for (let i = 0; i < lessonName.length; i++) {
        const divLessonName = document.createElement("div")
        divLessonName.className = "divLessonName"

```

```

const inputLesson = document.createElement("input")

inputLesson.className = "lesson"
inputLesson.setAttribute("type", "radio")
inputLesson.setAttribute("name", "lessons")
inputLesson.setAttribute("value", lessonName[i].eng)
inputLesson.setAttribute("id", `lessons${i}`)

const labelLesson = document.createElement("label")
labelLesson.setAttribute("for", `lessons${i}`)
labelLesson.innerHTML = lessonName[i].ukr

divLessonName.append(inputLesson)
divLessonName.append(labelLesson)

menu.append(divLessonName)
}

const titleClassMenu = document.createElement("h2")
titleClassMenu.innerHTML = "Оберіть клас:"
menu.append(titleClassMenu)

for (let i = 1; i < 12; i++) {
  const divClassName = document.createElement("div")
  divClassName.className = "divClassName"

  const inputClass = document.createElement("input")

  inputClass.className = "class"
  inputClass.setAttribute("type", "radio")
  inputClass.setAttribute("name", "class")
  inputClass.setAttribute("id", `class${i}`)
  inputClass.setAttribute("value", i)

  const labelClass = document.createElement("label")
  labelClass.setAttribute("for", `class${i}`)
  labelClass.innerHTML = `${i} Клас`

  divClassName.append(inputClass)
  divClassName.append(labelClass)

  menu.append(divClassName)
}

const titleNoLessonsMenu = document.createElement("h2")
titleNoLessonsMenu.innerHTML = "Не цікавить"
titleNoLessonsMenu.style.color = "red"
menu.append(titleNoLessonsMenu)

for (let i = 0; i < lessonName.length; i++) {
  const divNoLessonName = document.createElement("div")
  divNoLessonName.className = "divLessonName"

  const inputNoLesson = document.createElement("input")

  inputNoLesson.className = "lessonNo"
  inputNoLesson.setAttribute("type", "radio")
  inputNoLesson.setAttribute("name", "lessonsNo")
  inputNoLesson.setAttribute("value", lessonName[i].eng)
  inputNoLesson.setAttribute("id", `lessonsNo${i}`)

```

```

        const labelNoLesson = document.createElement("label")
        labelNoLesson.setAttribute("for", `lessonsNo${i}`)
        labelNoLesson.innerHTML = lessonName[i].ukr

        divNoLessonName.append(inputNoLesson)
        divNoLessonName.append(labelNoLesson)

        menu.append(divNoLessonName)
    }

} else searchMenuDiv.remove()

}

// Главный блок по созданию книг
function books(librayBook, type) {

    if (!type == "rating") librayBook.sort(() => Math.random() - 0.5);
    else if (!type) librayBook.sort(() => Math.random() - 0.5);
    else {
        librayBook.sort(function (a, b) {
            return b.rating - a.rating
        })
    }

    const searchBooks = document.querySelector(".books")

    if (searchBooks) searchBooks.remove()

    if (librayBook.length == 0) {
        const books = document.createElement("div")
        books.className = "books"

        const noBooks = document.createElement("h2")
        noBooks.className = "nobooks"
        noBooks.innerHTML = "Книг немає!"
        books.append(noBooks)
        root.append(books)

    } else {

        const books = document.createElement("div")
        books.className = "books"

        root.append(books)

        for (let i = 0; i < librayBook.length; i++) {

            if (i % 4 == 0) {
                var divBooks = document.createElement("div")
                divBooks.className = `divbook${i}`
            }
        }
    }
}

```

```

const book = document.createElement("div")

divBooks.append(book)
books.append(divBooks)

book.className = `book`
book.setAttribute("id", libraryBook[i].id)

book.innerHTML = `
<h3>${libraryBook[i].name}</h3><h3>${libraryBook[i].author}</h3>`

const bookImg = document.createElement("img")

if (libraryBook[i].img) bookImg.setAttribute("src", libraryBook[i].img)
else bookImg.setAttribute("src", "img/noIMGBook.jpg")

book.append(bookImg)

const rating = document.createElement("p")
rating.className = "rating"
rating.innerHTML = `Рейтинг: ${libraryBook[i].rating}`
book.append(rating)

const year = document.createElement("p")
year.innerHTML = libraryBook[i].year

book.append(year)
if (libraryBook[i].pdf) {
  const buttonRead = document.createElement("a")
  buttonRead.innerHTML = "Читати"
  buttonRead.setAttribute("target", `_blank`)
  buttonRead.setAttribute("href", `${libraryBook[i].pdf}`)
  book.append(buttonRead)
}
else {
  const buttonRead = document.createElement("h3")
  buttonRead.innerHTML = "Книга відсутня!"
  book.append(buttonRead)
}

const details = document.createElement("details")
const summary = document.createElement("summary")
summary.innerHTML = "Знайти схожі книжки по"

details.append(summary)
const button = document.createElement("p")
button.className = "searchBooks"
button.innerHTML = "предмету"
button.setAttribute("onclick", `options(libraryBook, "book", "${libraryBook[i].lessonTitle}",
""))

details.append(button)

const button1 = document.createElement("p")
button1.className = "searchBooks"
button1.innerHTML = "класу"
button1.setAttribute("onclick", `options(libraryBook, "book", "", "${libraryBook[i].class}`)`)
details.append(button1)

const buttonTopic = document.createElement("p")

```

```

        buttonTopic.className = "searchBooks"
        buttonTopic.innerHTML = "теми"
        buttonTopic.setAttribute("onclick", `topic(librayBook, "${librayBook[i].topic}")`)
        details.append(buttonTopic)

        book.append(details)

        const button2 = document.createElement("p")
        button2.className = "searchBooks"
        button2.innerHTML = "Додати в улюблене"
        button2.setAttribute("onclick", `addBook(librayBook, ${librayBook[i].id})`)
        book.append(button2)

    }
}
}

// Блок сортировки по выбору предмета и книги
function options(librayBook, type, lessons, classB, topic) {
    const searchMenuDiv = document.querySelector(".menu")

    var l = lessons
    var c = classB
    var t = topic

    let allLesson, allClass, noLessonValue
    var lessonChoice, classChoice, newLibrayBooks, noLesson, topicChoice

    if (type == "button") {
        allLesson = document.querySelectorAll(".lesson")
        allClass = document.querySelectorAll(".class")
        noLesson = document.querySelectorAll(".lessonNo")

        for (let i = 0; i < allLesson.length; i++) {
            if (allLesson[i].checked) lessonChoice = allLesson[i].value
        }

        for (let i = 0; i < allClass.length; i++) {
            if (allClass[i].checked) classChoice = allClass[i].value
        }

        for (let i = 0; i < noLesson.length; i++) {
            if (noLesson[i].checked) {
                noLessonValue = noLesson[i].value
            }
        }
    }
    else if (type == "book") {
        lessonChoice = l
        classChoice = c
        topicChoice = t
    }

    if (noLessonValue == lessonChoice) {
        const searchMesseg = document.querySelector(".messeng")
        if (searchMesseg) searchMesseg.remove()
        else {
            const messeng = document.createElement("div")
            messeng.className = "messeng"

```

```

messeng.style.background = "red"
messeng.innerHTML = "Ви обрали однакові предмети"
root.append(messeng)
menu(lessonName)

setTimeout(() => {
    const searchMesseg = document.querySelector(".messeng")
    searchMesseg.remove()

}, 5000);
}

}
else {
    if (noLessonValue) {
        var newlibrayBook = librayBook.filter((item) => item.lessonTitle !== noLessonValue);
        if (lessonChoice) {
            if (classChoice) {
                let arr = newlibrayBook.filter(elem => elem.lessonTitle === lessonChoice);
                arr = arr.filter(elem => elem.class === classChoice);
                books(arr, "rating")
            }
            else {
                newLibrayBooks = newlibrayBook.filter(elem => elem.lessonTitle ===
lessonChoice);
                books(newLibrayBooks, "rating")
            }
        }
        else {
            newLibrayBooks = newlibrayBook.filter(elem => elem.class === classChoice);
            books(newLibrayBooks, "rating")
        }
        if (type === "button") searchMenuDiv.remove()
    }
    else {
        if (lessonChoice) {
            if (classChoice) {
                let arr = librayBook.filter(elem => elem.lessonTitle === lessonChoice);
                arr = arr.filter(elem => elem.class === classChoice);
                books(arr, "rating")
            }
            else {
                newLibrayBooks = librayBook.filter(elem => elem.lessonTitle ===
lessonChoice);
                books(newLibrayBooks, "rating")
            }
        }
        else {
            newLibrayBooks = librayBook.filter(elem => elem.class === classChoice);
            books(newLibrayBooks, "rating")
        }
        if (type === "button") searchMenuDiv.remove()
    }
}
}

function topic(librayBook, topic) {

const newlibrayBook = librayBook

let arr = newlibrayBook.filter(elem => elem.topic === topic);

```

```

books(arr, "rating")

}
// Блок сортировки по классу юзера
function optionsUser(librayBook, user) {

console.log(librayBook)
console.log(user)

const searchMenuDiv = document.querySelector(".menu")
const newLibrayBooksUser = librayBook.filter(elem => elem.class == user[userId].class);
books(newLibrayBooksUser, "rating")
searchMenuDiv.remove()
}
// Блок для регистрации
function loginDiv(user) {

const windowLoginAndPassword = document.createElement("div")
windowLoginAndPassword.className = "windowLoginAndPassword"

const inputLogin = document.createElement("input")
inputLogin.setAttribute("placeholder", "Логін")
inputLogin.className = "login"

const inputPassword = document.createElement("input")
inputPassword.setAttribute("placeholder", "Пароль")
inputPassword.setAttribute("type", "password")
inputPassword.className = "password"

const button = document.createElement("button")
button.innerHTML = "Увійти"
button.setAttribute("onclick", "login(user)")

windowLoginAndPassword.append(inputLogin)
windowLoginAndPassword.append(inputPassword)
windowLoginAndPassword.append(button)
root.append(windowLoginAndPassword)
}
//Блок проверки логина и пароля
function login(user) {
const login = document.querySelector(".login").value
const password = document.querySelector(".password").value

for (let i = 0; i < user.length; i++) {
    if (user[i].login == login && user[i].password == password) {
        const window = document.querySelector(".windowLoginAndPassword")
        userId = i
        window.remove()
        header(user[i])
        buttonFunction()
        books(librayBook)
    }
}

}
//Добавить книги в любимые
function addBook(librayBook, id) {
for (let i = 0; i < librayBook.length; i++) {

    if (librayBook[i].id == id) myFavoriteBooks[myFavoriteBooks.length] = librayBook[i]

```

```

}
myFavoriteBooks = myFavoriteBooks.reduce((o, i) => {
  if (!o.find(v => v.id == i.id)) {
    o.push(i);
  }
  return o;
}, []);
}

//Добавление жанров доставая с книги
function genre() {
for (let i = 0; i < myFavoriteBooks.length; i++) {
  myFavoriteGenre[myFavoriteGenre.length] = { lessonTitle: myFavoriteBooks[i].lessonTitle }
  myFavoriteGenre = myFavoriteGenre.reduce((o, i) => {
    if (!o.find(v => v.lessonTitle == i.lessonTitle)) {
      o.push(i);
    }
  }
  return o;
}, []);
}
giveBooksForGenre(librayBook)
}

//создаем массив по моим выбранным книгам и их жанрам
var newBooksForGenre = []
var arrBook = []
function giveBooksForGenre(librayBook) {

for (let i = 0; i < myFavoriteGenre.length; i++) {
  newBooksForGenre[i] = librayBook.filter(elem => elem.lessonTitle ==
myFavoriteGenre[i].lessonTitle);

  arrBook.push(...newBooksForGenre[i])
}

arrBook = arrBook.reduce((o, i) => {
  if (!o.find(v => v.id == i.id)) {
    o.push(i);
  }
  return o;
}, []);

books(arrBook, "rating")

const searchMesseg = document.querySelector(".messeng2")
if (searchMesseg) searchMesseg.remove()
else {
  const messeng = document.createElement("div")
  messeng.className = "messeng2"
  messeng.innerHTML = "Ваші улюбленні жанри:"

for (let i = 0; i < myFavoriteGenre.length; i++) {
  const genreList = document.createElement("p")
  genreList.innerHTML = myFavoriteGenre[i].lessonTitle
  messeng.append(genreList)
}

root.append(messeng)

```

```
        setTimeout(() => {
            const searchMesseg = document.querySelector(".messeng2")
            searchMesseg.remove()
        }, 10000);
    }
}
loginDiv()
// buttonFunction()
// books(librayBook)
```

Програмний код books.css

```
.books {  
  display: flex;  
  flex-direction: column;  
}  
  
.book {  
  border: 1px solid black;  
  
  display: flex;  
  justify-content: space-around;  
  
  flex-direction: column;  
  align-items: center;  
  
  margin: 15px;  
  margin-left: 20px;  
  margin-bottom: 60px;  
  
  padding: 20px;  
  width: 230px;  
  
  min-height: 460px;  
  border-radius: 20px;  
  box-shadow: 2px 2px 10px black;  
}  
  
.divbook0,  
.divbook4,  
.divbook8,  
.divbook12,  
.divbook16,  
.divbook20,  
.divbook24,  
.divbook28,  
.divbook32,  
.divbook36,  
.divbook40,  
.divbook44,  
.divbook48,  
.divbook52,  
.divbook56,  
.divbook60,  
.divbook64,  
.divbook68,  
.divbook72,  
.divbook76,  
.divbook80,  
.divbook84,  
.divbook88,  
.divbook92,  
.divbook96,  
.divbook100,  
.divbook104,  
.divbook108,  
.divbook112,  
.divbook116,  
.divbook120,
```

```
.divbook124 {
  display: flex;
  flex-direction: row;
  margin-left: 40px;
}

.books img {
  width: 200px;
  height: 300px;
}

.books a {
  text-decoration: none;
  color: black;
  border: 1px solid black;

  padding: 10px 20px;

  border-radius: 20px;
  cursor: pointer;
  margin-top: 20px;
}

.books a:hover {
  background: #fdb93a;
}

.books a:active {
  background: #ffeece;
}

.nobooks {
  background: none;
  text-align: center;
  color: red;
}

details {
  user-select: none;
  cursor: pointer;
  margin-top: 20px;
}

.searchBooks {
  margin: 10px;
  user-select: none;
  cursor: pointer;
  border-bottom: 1px solid black;
}

.searchBooks:hover {
  color: blue;
}

h3 {
  margin: 20px 0;
}

.rating {
  margin: 20px 0px;
}
```

Програмний код header.css

```
.header {  
  display: flex;  
  justify-content: space-between;  
  
  align-items: center;  
  
  background: #f7d28e;  
  
  padding: 20px 50px;  
  box-shadow: 0px 4px 20px black;  
  
}  
  
.logo {  
  cursor: pointer;  
  user-select: none;  
}
```

Програмний код main.css

```
@import "header.css";
@import "books.css";
@import "search.css";

* {
  margin: 0;
  padding: 0;

  font-family: Trebuchet MS, sans-serif;
}

.root {
  position: relative;
}

button {
  cursor: pointer;
}

button:hover {
  background: #fdb93a;
}

button:active {
  background: #ffeece;
}

.windowLoginAndPassword {
  margin-left: 40vw;
  margin-top: 20vh;
  width: 400px;
  padding: 30px;
  border: 2px solid black;
  border-radius: 30px;
  display: flex;
  flex-direction: column;

  justify-content: center;
  background: rgb(209, 209, 209);
}

.windowLoginAndPassword input {
  margin-bottom: 30px;
  padding: 10px;
  border-radius: 30px;
}

.windowLoginAndPassword button {
  padding: 10px;
  border-radius: 30px;
}

.messeng {
  padding: 30px;
  font-size: 20px;
  position: absolute;
```

```
top: -40px;

border-radius: 0 0px 30px 30px;
border: 2px solid black;
left: 30%;

}

.messeng2 {
padding: 30px;
font-size: 20px;
position: absolute;
top: -20px;
max-width: 500px;
border-radius: 0 0px 30px 30px;
border: 2px solid black;
right: 20%;
z-index: 2;
background: #838383;
color: white;
display: flex;
flex-direction: column;

}

.messeng2 p {
margin-left: 10px;

}
```

Програмний код search.css

```
.search {  
  
    display: flex;  
    justify-content: center;  
    margin: 30px;  
}  
  
.search input {  
    padding: 10px 30px;  
    border-radius: 30px;  
    width: 30%;  
    min-width: 200px;  
}  
  
.search button {  
    margin-left: 40px;  
    padding: 10px 20px;  
    font-size: 15px;  
    border-radius: 30px;  
}  
  
.search button:last-child {  
    margin-left: 50%;  
    padding: 10px 20px;  
    font-size: 15px;  
    border-radius: 30px;  
}  
  
.menu {  
    background: white;  
    position: absolute;  
    top: 200px;  
    right: 40px;  
  
    width: 500px;  
    padding: 10px 30px;  
  
    border: 1px solid black;  
    border-radius: 30px;  
  
    z-index: 1;  
  
    display: flex;  
    flex-direction: column;  
  
    background: #eeeeee;  
}  
  
.menu h2 {  
    margin-top: 10px;  
}  
  
.divLessonName,  
.divClassName {  
    margin-top: 10px;
```

```
}  
  
.divLessonName label,  
.divClassName label {  
  margin-left: 10px;  
}  
  
.searchMenu {  
  margin-top: 20px;  
  border-radius: 30px;  
  padding-top: 10px;  
  padding-bottom: 10px;  
}  
  
.recommendationMenu {  
  background: rgb(248, 147, 147);  
  margin-top: 10px;  
  border-radius: 30px;  
  padding-top: 10px;  
  padding-bottom: 10px;  
}
```