

на тему Створення інтелектуальної системи рекомендацій
(назва кваліфікаційної роботи згідно наказу ОНТУ)
для відвідувачів інтернет-магазинів на прикладі
магазину з продажу художньої літератури

4 курсу 342а групи

(посада, прізвище та ініціали)

доц. Спаський І.Д.
(посада, прізвище та ініціали)

Завідувач кафедри КФМН _____ Юлія ФЕДЧЕНКО
(назва кафедри) (підпис) (Ім'я ПРИЗВИЩЕ)

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Інститут Навчально-науковий інститут комп'ютерної інженерії,
автоматизації, робототехніки та програмування
ім. П.М. Платонова

Кафедра комп'ютерних та фізико-математичних наук

Ступінь вищої освіти Бакалавр

Спеціальність 122 «Комп'ютерні науки»

Освітня програма Інформаційні технології проектування

ЗАТВЕРДЖУЮ

Зав. кафедри КФМН

Юлія ФЕДЧЕНКО

« 02 » лютого 2026 року

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Чередніченко Віталія Сергійовича

1. Тема роботи Створення інтелектуальної системи рекомендацій для відвідувачів
інтернет-магазинів на прикладі магазину з продажу художньої літератури

Затверджена наказом університету від « 18 » грудня 2025 р., наказ № 710-03

2 Термін здачі здобувачем закінченої роботи 8 червня 2026 р.

3. Вихідні дані роботи

Спроектувати та розробити сайт інтернет-магазину для книг та втілити
рекомендаційну систему, яка на основі потреб користувача буде надавати
рекомендовані книги

4. Перелік питань, які потрібно розробити

1. Аналіз предметної області. 2. Аналіз існуючих типів рекомендаційних систем.

3. Методи та технології проєкту, обґрунтування вибору засобів реалізації. 4. Етапи реалізації
програмного продукту з тестуванням результату. 5. Техніко-економічне обґрунтування.

6. Охорона праці

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

Презентація Power Point 13 слайдів

6. Консультанти по роботі, із зазначенням розділів роботи, що стосуються їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Техніко-економічна частина	Спаський І.Д., доцент		
Охорона праці	Корнієнко Ю. К., доцент		

7. Дата видачі завдання 18.12.2025

Керівник _____ Сергій КОТЛИК

Завдання прийняв до виконання _____ Віталій ЧЕРЕДНІЧЕНКО

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	Дослідження предметної області	28.02.2026	
2.	Аналіз існуючих аналогів	28.02.2026	
3.	Обґрунтування вибору засобів реалізації	28.02.2026	
4.	Моделювання та проєктування системи	01.04.2026	
5.	Реалізація програмного продукту	01.05.2026	
6.	Підготовка техніко-економічної частини	20.05.2026	
7.	Підготовка розділу охорони праці	25.05.2026	
8.	Оформлення пояснювальної записки	1.06.2026	
9.	Оформлення графічної частини та лістингу	1.06.2026	

Здобувач - дипломник _____ Віталій ЧЕРЕДНІЧЕНКО

Керівник роботи _____ Сергій КОТЛИК

Несу відповідальність за ідентичність електронного та друкованого варіантів кваліфікаційної роботи, даю згоду на обробку персональних даних та не заперечую проти розміщення кваліфікаційної роботи на офіційних web-ресурсах ОНТУ.

Підтверджую, що в кваліфікаційній роботі відсутні порушення норм академічної доброчесності.

Здобувач-дипломник Віталій ЧЕРЕДНІЧЕНКО _____

АНОТАЦІЯ

Розглянено предметну область рекомендаційних систем з аналізом їх корисності для комерційних платформ. Виконано аналіз колаборативної фільтрації, контентної фільтрації та гібридних методів. Обрано контентну фільтрацію для втілення в програмний продукт, оскільки вона здатна формувати рекомендації для користувача не вимагаючи наявності великої кількості інших користувачів.

Обґрунтовано та описано методи та інструментарій, які були використані для реалізації системи. Під час проєктування обрано клієнт-серверну архітектуру для реалізації та REST API. Використано мову JavaScript для реалізації функціоналу сторінок та для взаємодії серверної частини з клієнтською, що забезпечується за допомогою Node.js. Побудовано каркас сторінок та надано стиль через використання HTML та CSS. Обрано PostgreSQL як систему керування базами даних для організації таблиць.

Спроектовано та організовано архітектуру магазину. Сформовано вимоги до можливостей системи та виконано проєктування таблиць у СУБД зі встановленням взаємозв'язків. Окремо виділено увагу до таблиці взаємодії, до якої вносяться дані дій користувача в інтернет-магазині для формування пропозицій.

Виконано реалізацію частин програмного продукту, тобто авторизацію, реєстрацію, перегляд сторінки книги, кошик, придбання книги, каталог, пошук товарів та профіль з відображенням історії придбань, або панелі керування адміністратора, що залежить від встановленої ролі.

Протестовано працездатність системи через послідовне виконання реєстрації та авторизації нового запису у БД та взаємодії з книгами. Рекомендаційна система формує список пропозицій книг з ознаками, схожими до тих, що подобаються користувачу. Виконано техніко-економічне обґрунтування розробки та розглянуто питання охорони праці, які забезпечують безпечні умови під час виконання роботи.

Ключові слова: рекомендаційна система, інтернет-магазин, контентна фільтрація, електронна комерція, база даних, REST API, персоналізація.

ABSTRACT

The subject area of recommendation systems is considered with an analysis of their usefulness for commercial platforms. An analysis of collaborative filtering, content filtering and hybrid methods is performed. Content filtering is chosen for implementation in a software product, because it is able to form recommendations for the user without requiring the presence of a large number of other users.

The methods and tools used to implement the system are justified and described. During the design, a client-server architecture was chosen for implementation and REST API. JavaScript was used to implement page functionality and for the interaction of the server part with the client, which is provided using Node.js. The page framework was built and style was provided using HTML and CSS. PostgreSQL was chosen as the database management system for organizing tables.

The store architecture was designed and organized. Requirements for the system capabilities were formed and tables in the DBMS were designed with the establishment of relationships. Special attention was paid to the design of the interaction table, which contains data on user actions in the online store for building offers.

The implementation of parts of the software product was carried out, i.e. authorization, registration, viewing the book page, shopping cart, purchasing a book, product search catalog and profile with display of purchase history, or the administrator control panel, depending on the assigned role.

The system's performance was tested by sequentially performing registration and authorization of a new record in the database and interaction with books. The recommendation system generates a list of book suggestions with features similar to those that the user likes.

A feasibility study of the development was performed and occupational health and safety issues were considered, which are responsible for ensuring safe conditions during work.

Keywords: recommendation system, online store, content filtering, e-commerce, database, REST API, personalization.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ВИМІРЮВАННЯ, СКОРОЧЕНЬ ТА ТЕРМІНІВ	8
ВСТУП.....	9
1. АНАЛІЗ ГАЛУЗІ ТА ОБҐРУНТУВАННЯ ОБРАНОЇ СИСТЕМИ	12
1.1 Опис рекомендаційних систем та їх історії.....	12
1.2 Огляд реальних сервісів, які використовують рекомендаційні системи.....	13
1.3 Опис та порівняння типів рекомендаційних систем	14
1.3.1 Контентна фільтрація.....	14
1.3.2. Колаборативна фільтрація.....	15
1.3.3 Гібридні методи.....	16
1.4 Основні проблеми рекомендаційних систем.....	16
1.5 Обґрунтування вибору типу рекомендаційної системи для реалізації	17
2. ОПИС ОСНОВНИХ МЕТОДІВ ТА ТЕХНОЛОГІЙ ПРОЄКТУ З ОБґРУНТУВАННЯМ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ	18
2.1 Вибір підходу та стилю.....	18
2.2 Вибір середовища розробки	20
2.3 Вибір СУБД	23
2.4 Використання технологій.....	25
3. ПРОЄКТУВАННЯ СИСТЕМИ.....	29
3.1 Визначення вимог до системи. Інформаційна модель системи	29

					КРБ.КФМН.1.710-03.2.6				
Змн.	Арк.	№ докум.	Підпис	Дата	Створення інтелектуальної системи рекомендацій для відвідувачів інтернет-магазинів на прикладі магазину з продажу художньої літератури	Лім.	Арк.	Аркушів	
Розробив		Чередніченко В.С.							
Перевірів		Котлик С.В.					6	115	
						ОНТУ, 122, 342 гр.			
Н Контр.		Котлик С.В.							
Затвердив		Федченко Ю.С.							

3.2	Проектування бази даних	31
4.	ОПИС ЕТАПІВ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ПРОДУКТУ	35
4.1	Архітектура та файлова структура програмного комплексу	35
4.2.	Програмна реалізація серверної логіки та взаємодії з БД	36
4.3.	Реалізація інтелектуального модуля формування рекомендацій	37
4.4.	Програмна розробка клієнтського інтерфейсу	38
4.5.	Тестування працездатності та рекомендацій	40
4.5.1	Процес реєстрації та авторизації користувачів	40
4.5.2	Тестування взаємодії користувача з функціоналом магазину	41
4.5.3	Тестування рекомендаційної системи	42
4.5.4	Тестування панелі керування адміністратора	44
5.	ТЕХНІКО-ЕКОНОМІЧНА ЧАСТИНА	46
6.	ОХОРОНА ПРАЦІ	62
	ЗАГАЛЬНІ ВИСНОВКИ	66
	ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	68
	ДОДАТОК А Презентація	70
	ДОДАТОК Б Лістинг	75
	ДОДАТОК В Публікації	80

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ВИМІРЮВАННЯ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

БД – база даних

СУБД – система управління базами даних

ЕОМ – електронна обчислювальна машина

HTTP – Hypertext Transfer Protocol

JSON – JavaScript Object Notation

JWT – JSON Web Token

HTML – HyperText Markup Language

CSS – Cascading Style Sheets

					КРБ.КФМН.1.710-03.2.6	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми. Сучасний Інтернет, незважаючи на відносно короткий період існування порівняно з іншими галузями, за цей час став важливою частиною ділового та суспільного життя. Це легко простежити через розвиток комп'ютерних технологій та появу великої кількості цифрових сервісів. Онлайн-магазини, сервіси бронювання, цифрові торгові платформи та служби доставки дозволяють компаніям реалізовувати товари і надавати послуги незалежно від місця перебування користувача. Одночасно зі зростанням кількості товарів і послуг виникла потреба в алгоритмах, які допомагають користувачам швидше знаходити необхідну інформацію та обирати продукцію відповідно до власних інтересів.

Одним з таких алгоритмів є рекомендаційні системи. Вони аналізують інформацію про взаємодію користувачів із товарами та формують персональні пропозиції. Завдяки цьому підвищується зручність використання торгових платформ, а також ефективність взаємодії між користувачем і каталогом товарів. У зв'язку з активним використанням рекомендаційних систем у сфері електронної комерції тема їх аналізу та програмної реалізації є актуальною. Саме розробці та впровадженню рекомендаційної системи для книжного інтернет-магазину присвячена ця робота.

Мета і задачі розробки. Мета цієї роботи – поетапне проєктування та програмна реалізація книжного інтернет-магазину з реалізацією рекомендаційної системи, яка формує персональні пропозиції на основі взаємодії користувача з книгами.

Досягнення поставленої мети реалізується через покрокове виконання наступних задач:

- проаналізувати існуючі типи рекомендаційних систем та принципи їх роботи;
- виконати порівняння основних підходів до формування рекомендацій;
- обґрунтувати вибір рекомендаційної системи для реалізації в роботі;

					КРБ.КФМН.1.710-03.2.6	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

- спроектувати структуру бази даних книжного інтернет-магазину;
- реалізувати клієнтську та серверну частини системи;
- реалізувати механізм взаємодії користувача з каталогом книг;
- реалізувати систему формування рекомендацій;
- виконати тестування працездатності створеного рішення.

Об'єктом дослідження є процес функціонування книжного інтернет-магазину та взаємодія користувача з його каталогом. У межах дослідження розглядаються особливості клієнт-серверної архітектури, зберігання та обробки даних, а також процес отримання інформації про дії користувача для подальшого формування рекомендацій. Паралельно розглядатимуться їх властивості та взаємозв'язок.

Предметом дослідження є рекомендаційні системи та методи їх реалізації у середовищі книжного інтернет-магазину. У дослідженні розглядаються їх сучасні типи, а саме колаборативна фільтрація, гібридні методи та контентна фільтрація. Саме шляхом визначення їх переваг та недоліків вибирається найбільш придатний метод.

Методи розробки. Для досягнення поставленої мети використовуються методи аналізу та порівняння рекомендаційних систем, методи проектування баз даних, принципи клієнт-серверної взаємодії, а також засоби розробки веб-технологій для створення інтернет-магазину та реалізації рекомендаційної системи.

Практичне значення одержаних результатів. Практичне значення роботи полягає у створенні працездатного книжного інтернет-магазину з підтримкою персоналізованих рекомендацій. Реалізована система дозволяє автоматизувати процес формування пропозицій для користувача на основі його взаємодії з книгами, що сприяє підвищенню зручності використання магазину та спрощує пошук товарів, які можуть зацікавити користувача.

Отримані результати можуть бути використані як основа для подальшого розвитку рекомендаційних механізмів та розширення функціональності інтернет-магазину.

					КРБ.КФМН.1.710-03.2.6	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

Апробація результатів роботи. Основні результати роботи були представлені у тезах до дипломного проєктування та використані під час розробки книжного інтернет-магазину з рекомендаційною системою.

					КРБ.КФМН.1.710-03.2.6	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

1. АНАЛІЗ ГАЛУЗІ ТА ОБҐРУНТУВАННЯ ОБРАНОЇ СИСТЕМИ

1.1 Опис рекомендаційних систем та їх історії

Для початку визначимо що являє собою рекомендаційна система. Інтелектуальна рекомендаційна система – ряд структурованих комп'ютерних алгоритмів, задача яких складається у отриманні даних з електронних платформ, які надаються користувачем при взаємодії з товарами, виконувати їх оцінку та на основі результату сформувати персональні пропозиції та рекомендації тих товарів, що співпадають своїми характеристиками, наданими клієнтами. Такі алгоритми підходять для формування пропозицій різного спектру товарів. Це може бути гра, в яку користувач захоче пограти, або книга, яку захочеться прочитати чи фільм, який можна подивитися.

Рекомендаційні системи є результатом довгого та стрімкого розвитку цифрових технологій. А саме вона охоплює галузі збору, аналізу інформації та розвитку штучного інтелекту. Завданням цих галузей полягало в тому, щоб допомогти краще орієнтуватися у великому масиві даних та ефективніше шукати інформацію чи товар, що відповідає потребам користувача.

Перші концепції таких алгоритмів почали зароджуватися в 1970-1980 роках у час активного розвитку пошукових систем та фільтрацій контенту. Але як окрему технічну галузь рекомендаційні системи набули популярності у 1990-х роках. Першою такою системою прийнято вважати Tapestry, яка була розроблена в 1992 році дослідниками Хероу PARC. Ця система використовує ручну фільтрацію та дозволяла користувачам ділитися між собою оцінками до документів, що своєю чергою стало фундаментом до побудови колаборативної фільтрації.

В середині 90-х стали більш розвиненими сучасні аналоги рекомендаційних систем, серед яких були GroupLens і MovieLens, що були розроблені в Міннесотському університеті. На той час вони вже були здатні в автоматичному порядку аналізувати дані, отримані від клієнтів, та вважаються одними з перших прикладів використання колаборативної фільтрації у

					КРБ.КФМН.1.710-03.2.6	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

сучасному сенсі. Часом прориву у практичному використанні подібного роду алгоритмів почався у періоди кінця 1990-х та початку 2000-х паралельно з розвитком галузі електронної комерції. Серед одних з перших компаній, хто втілила їх стала компанія Amazon. Це стало важливим чинником в розвитку, втіленні та використанні адаптивних інтелектуальних рекомендаційних систем. Зараз вони активно використовуються інтернет-магазинах, стрімінгових сервісах, музичних платформах і т. д.

1.2 Огляд реальних сервісів, які використовують рекомендаційні системи

Оскільки сучасний ринок перенасичений різноманітними електронними комерційними платформами, це дає можливість для їх перегляду. Розглянемо декілька прикладів платформ, в яких активно використовуються подібні адаптивні рекомендації.

YouTube

YouTube – найпопулярніша платформа для розміщення відео-контенту. YouTube активно застосовує систему аналізу потреб, на основі яких формується каталог відео, що відповідають уподобанням користувача. Розглядаються не лише жанри, а такі показники як підписка, історія перегляду, швидкість прокручування, коментарі та інше. Наявність стількох даних дозволяють швидко та ефективно підлаштовуватися до потреб своєї аудиторії та підбирати цікаий контент.

Spotify

Spotify – стрімінговий сервіс, що містить в собі цифровий аудіо-контент, такі як музика, подкаст, відео та інші. На програмному рівні активно використовуються алгоритми для формування персональних плейлістів контенту. Spotify аналізує прослуховування, частоту повторення композицій, жанри та інші показники, адаптуючись тим самим до аудиторії.

					КРБ.КФМН.1.710-03.2.6	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

Facebook та Instagram

Facebook та Instagram — відомі соціальні мережі публічного доступу з можливістю створення, публікації та обміну контентом різного формату. Рекомендаційні механізми цих платформ відображаються через вміст контентної стрічки, публікації та профілі. Усе це підлаштовується до персональних потреб та інформації про діяльність.

Amazon

Amazon — глобальний комерційний сервіс для інтернет покупок. Amazon один з перших, хто почав втілювати інтелектуальні рекомендаційні системи. Вона використовується для вивчення покупок, перегляду товарів, оцінок та інших дій, отриманих при користуванні. У результаті формуються персональні пропозиції, які відображаються в розділах «Схожі товари» чи «Інші також купили».

Netflix

Netflix — платформа для розміщення та перегляду фільмів, серіалів та анімаційних проєктів. Для Netflix рекомендаційна система відіграє активну роль в визначенні контенту, який пропонується аудиторії. Аналізуються такі чинники, як історія переглядів, виставлена оцінка, час, витрачений на взаємодію з контентом, та поведінка інших клієнтів, які мають схожі інтереси. В результаті користувач отримує контент, що відповідає його інтересам.

1.3 Опис та порівняння типів рекомендаційних систем

Тепер розглянемо різні типи рекомендаційних систем та їхні особливості. А саме огляд проводитиметься над колаборативною фільтрацією, контентною фільтрацією та гібридними методами.

1.3.1 Контентна фільтрація.

Контентна фільтрація — один з основних методів побудови рекомендаційної системи. Цей алгоритм базується на отриманні інформації щодо

					КРБ.КФМН.1.710-03.2.6	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

дій користувача в системі та їх ідентифікації. Також розглядаються характеристики товарів, з якими відбувалася взаємодія. Остаточним результатом є виведення товарів, схожих на ті, якими раніше цікавились.

Об'єктам веб-ресурсу надаються спеціальні атрибути, до яких можна віднести жанри, мову, формат, дату видання або інші. Система джерела товарів збирає ці дані та на їх основі формує профіль, де відображає рекомендації побудовані на описаних вище ознаках з певними вагами. Так коли відображаються нові книги, то система порівнює їх з персональними профілями та визначає ступінь відповідності, за яким будуються пропозиції. Серед переваг контентної фільтрації слід зазначити:

- побудова рекомендацій на основі вподобань;
- незалежність роботи системи від даних інших користувачів;
- стабільне функціонування навіть при невеликій аудиторії;
- можливість аналізу рекомендацій(тобто причину інтересу користувачів до певного товару.

1.3.2. Колаборативна фільтрація.

Колаборативна фільтрація є способом організації рекомендаційної системи, робота якої заснована на контакті великої кількості її учасників з системою, тому на відміну від контентної фільтрації функціонує через колективний досвід підлаштовуючи пропозиції під уподобання груп, що мають схожі інтереси. Існує два способи реалізації колаборативної фільтрації, а саме орієнтований на користувачів та орієнтований на об'єкти. Основні плюси колаборативної фільтрації:

- здатність знаходити неочевидні зв'язки між об'єктами сервісу;
- більша різноманітність при створенні рекомендацій;
- відсутність необхідності в детальному описі характеристик товарів;
- врахування реального досвіду та інтересів клієнтів;
- задовільна працездатність у разі наявності великої кількості даних.

					КРБ.КФМН.1.710-03.2.6	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

1.3.3 Гібридні методи.

Ідея гібридних методів відрізняється від раніше розглянутих систем поєднанням одночасно декількох методів для покращення ефективності рекомендаційних систем. Дуже часто мова йде про з'єднання колаборативної та контентної фільтрації. Але все не обмежується лише ними двома та цілком дозволяє використовувати інші підходи, наприклад машинного навчання, аналіз поведінки і т. д.

Однак незважаючи на загальні плюси, гібридні методи досить складні у реалізації. Їх використання є вимогливим до технічних потреб розробника і вимагає ретельного налаштування та оптимізації алгоритмів. Головні плюси гібридного методу:

- поєднання сильних сторін різних підходів (контентного та колаборативного);
- підвищена точність і релевантність рекомендацій;
- більш різноманітні та збалансовані рекомендації;
- краща адаптація до різних типів даних і сценаріїв використання.

1.4 Основні проблеми рекомендаційних систем

Хоча було перелічено багато плюсів, але існують також проблеми та недоліки, на які важливо звернути увагу. Серед них важливою є «холодний старт», проблема, що з'являється при недостатній кількості мінімально накопиченої інформації для коректної роботи. Ці дані можуть стосуватися як користувача, так і товару.

Окрім холодного старту серйозним мінусом є надто вузький спектр пропозицій, які рекомендаційна система може зробити. Це призводить до руйнування різноманітності продукції.

					КРБ.КФМН.1.710-03.2.6	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

Слід додатково зазначити труднощі з роботою при великих обсягах даних, які значно можуть вплинути на продуктивність у тому випадку, якщо ще не виконано грамотну оптимізацію.

1.5 Обґрунтування вибору типу рекомендаційної системи для реалізації

Після переліку усіх трьох типів було вирішено обрати контентну фільтрацію. Обґрунтування у виборі полягає в тому, що на відміну від тієї ж колаборативної фільтрації є можливість будувати список схожих товарів за невеликою кількістю користувачів, що надала недостатньо інформації. Таким чином запобігається проблема "холодного старту".

Слід звернути увагу на архітектуру цього проєкту. Система передбачає каталог, заповнений продуктами, які матимуть певні атрибути. Відповідна побудова сайту є сприятливою, оскільки її структура сайту побудована відповідно до принципів роботи системи та використовує таку структуру для власної ефективності.

Контентна фільтрація також підходить для проєкту за свою прозорість. Тобто надається можливість бачити рекомендації та розуміти чому йому рекомендовано той чи інший товар, що покращує досвід користувача та підвищує його лояльність і довіру до системи. Наступний плюс – невибагливість в технічному плані, завдяки чому відсутня потреба строго налаштування, оптимізації та надмірної потужності обчислювальних ресурсів. Хоча може показатися що деякі мінуси, такі як обмеження відкриття більш різноманітних товарів й відповідно створення однотипних рекомендацій заважатимуть, але це не є досить критичним.

					КРБ.КФМН.1.710-03.2.6	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

2. ОПИС ОСНОВНИХ МЕТОДІВ ТА ТЕХНОЛОГІЙ ПРОЄКТУ З ОБГРУНТУВАННЯМ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ

Перед початком проєктування перш за все необхідно проаналізувати увесь набір інструментарію, архітектурних стилів, систем керування базами даних та технологіями. Важливість цього вибору полягає в тому, що результат вплине на підсумкову ефективність, зручність, розширюваність та безпеку середовища каталогу.

Обрані засоби реалізації мають забезпечувати стабільну роботу інтернет-магазину, підтримку взаємодії між клієнтською й серверною частинами, зручне зберігання інформації про користувачів і товари, та можливість реалізації інтелектуальної системи рекомендацій. Саме тому таким чином вибір здійснюється з урахуванням функціональних вимог, особливостей предметної області та потреб сучасної розробки.

2.1 Вибір підходу та стилю

Почнемо з розгляду стилю та підходу, який є початковим фундаментальним кроком, бо визначає структуру усього програмного продукту, принципи обміну даними та майбутньою можливістю розширення. Для цього розглянемо декілька відомих архітектур, з описом плюсів та мінусів. У сучасних проєктах найчастіше використовуються монолітна, клієнт-серверна та мікросервісна архітектури, саме серед них і виберемо потрібну.

Монолітна архітектура є підходом, за яким усі функції описуються у в рамках одного проєкту. Такий вибір корисний тим, що спрощує процес початкової розробки на стартовому етапі та відносно мала вимогливість до налаштування інфраструктури. Але ж незважаючи на спрощення – при збільшенні кількості та обсягу функцій при оновленні ускладнюється сама підтримка розроблюваного сервісу, а кожне нове внесення змін може погіршити ситуацію, уповільнивши роботу всієї системи. Додатково зростає обсяг споживаних ресурсів необхідних для життєвого циклу праці.

					КРБ.КФМН.1.710-03.2.6	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

Наступним варіантом є мікросервісна архітектура, яка виконує поділ програми на відокремлені незалежні сервіси, де для кожного такого сервісу передбачено власне виконання окремого завдання, з забезпеченням взаємодії з іншими сервісами через спеціальний мережевий інтерфейс. Мікросервісна архітектура корисна тим, що забезпечує високу масштабованість, незалежність окремих компонентів та можливість використання різних технологій у різних частинах системи. Але у описаного рішення є одна проблема. Незважаючи на гарантовану масштабованість, для невеликих та середніх робіт використання мікросервісів часто не є раціональним через складність налаштування, адміністрування та одночасної підтримки великої кількості сервісів.

Тепер перейдемо до клієнт-серверної архітектури. Клієнт-серверна архітектура – засіб організації архітектури, за якою система розподіляється на окремі взаємодіючі частини – клієнтську та серверну.

Клієнтська частина відповідає за взаємодією користувача з системою, тобто за візуальний інтерфейс, надсилання запитів до сервера на обробку та за виведення тих самих оброблених результатів запиту. Усі процедури, здійснювані з боку клієнтів, виконуються з відповідного клієнтського пристрою. Клієнтська частина в цій роботі реалізується через використання технологій HTML, CSS та JavaScript і запускається у веб-браузері користувача.

Серверна частина отримує клієнтські запити та виконує їх обробку, надаючи результат у відповідь до запиту, реалізує бізнес-логіку системи, забезпечує коректну роботу з базою даних та організує авторизацію облікових записів. Таке розподілення надає незалежність інтерфейсу від серверної логіки, дозволяючи тим чином ефективно модернізувати систему у майбутньому.

Серед трьох підходів було вирішено обрати клієнт-серверну архітектуру, що обумовлено можливістю централізованого збереження інформації про користувачів, книги, замовлення та результати рекомендацій літератури. Окрім цього слід зазначити, що ця архітектура забезпечує підвищенню рівня безпеки збереження даних, оскільки основна програмна логіка та база даних ізольовані

					КРБ.КФМН.1.710-03.2.6	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

від чужих очей та залишаються недоступними для прямого втручання з боку клієнтської частини.

Тепер визначимось зі стилем взаємодії між частинами вмісту системи. Найкраще підходить стиль REST(Representational State Transfer), що активно застосовується у сучасних галузях розробки. REST заснований на використанні стандартного протоколу HTTP, низці спеціальних правил та постановок взаємодії між клієнтом та сервером. Використання REST API є оптимальним саме для інтернет-магазину, бо більшість операцій системи пов'язані з створенням, отриманням, оновленням та видаленням даних. Каталог книг, кошик, замовлення, профілі користувачів, профілі адміністраторів та рекомендації представляються у вигляді окремих ресурсів. REST дозволяє працювати з ними за допомогою стандартних HTTP-запитів, що спрощує взаємодію між клієнтською та серверною частинами системи. До основних таких запитів належать GET, POST, PUT та DELETE. Запит GET надсилається у тому випадку, коли потрібно отримати певні дані із сервера. У протилежній ситуації коли вже важливо створити новий запит для надсилання на сервер, то для цього використовується POST. Коли потрібно оновити інформацію, то оптимальним запитом є PUT(або PATCH) та своєю чергою при потребі видалення інформації використовується DELETE.

Процедура передачі даних у межах роботи здійснена у форматі JSON. Цей формат є досить поширеним шляхом обміну для чітко структурованих даних через їх збереження у вигляді пар «ключ-значення».

В результаті коли ми обрали архітектурний підхід та стиль відштовхуючись від їх властивостей, перейдемо до наступного пункту.

2.2 Вибір середовища розробки

Важливим етапом реалізації будь-якої програмної частини є вибір середовища розробки. Від обраного програмного забезпечення буде залежати зручність процесу написання коду, швидкість виконання типових операцій,

					КРБ.КФМН.1.710-03.2.6	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

можливість налагодження та ефективність роботи розробника взагалі. Сучасний ринок програмного забезпечення пропонує значну кількість інтегрованих середовищ розробки та редакторів, кожен з яких має власні переваги та особливості. Серед найбільш популярних інструментів для веб-розробки можна виділити Visual Studio, WebStorm, Sublime Text та Visual Studio Code, тому виконаємо їх порівняння в таблиці.

Таблиця 2.1 – Плюси та мінуси середовищ розробки

Назва середовища	Плюси	Мінуси
Visual Studio	Містить розвинені засоби налагодження, підтримує велику кількість мов програмування та інтеграцію з різноманітними сервісами.	Для невеликих програмних продуктів це середовище може виявитися надмірно складним та вимогливим до апаратних ресурсів.
WebStorm	Спеціалізується на веб-розробці та забезпечує глибоку інтеграцію з JavaScript, Node.js та сучасними веб-технологіями.	Використання цього середовища потребує придбання ліцензії, що може бути недоцільним для навчальних або індивідуальних робіт.
Sublime Text	При розробці надає високу швидкодію та простоту для використання.	Функціональність редактора значно залежить від додаткових модулів і поступається можливостям сучасних інтегрованих

		середовищ розробки.
Visual Studio Code	Поєднує простоту у використанні з широкими функціональними можливостями, підтримує роботу з великою кількістю мов програмування, підтримка системи автоматичного доповнення коду, можливість завантаження та налаштування великої кількості функціональних плагінів, має вбудовану систему керування версіями Git.	Споживає високий обсяг ресурсів, залежний від сторонніх плагінів, складний у налаштуванні для масштабних проєктів у порівнянні з повноцінними IDE

Серед розглянутих середовищ більш усього підходить до використання Visual Studio Code. Однією з основних причин його вибору є гнучка інтегрованість з великим різноманіттям технологій, що дозволяє нам використовувати JavaScript, HTML, CSS та SQL, завдяки цьому вся розробка клієнтської та серверної частин може виконуватись у межах одного середовища.

Крім того, Visual Studio Code містить вбудований термінал, що дозволяє виконувати команди Node.js, працювати з менеджером пакетів npm та запускати серверну частину без необхідності використання сторонніх програм.

Ще однією ключовою перевагою є підтримка системи контролю версій Git, завдяки чому можливо зберігати проміжні результати роботи, відстежувати зміни у коді та за необхідністю повертатися до минулих версій, тобто до комітів.

Окремої уваги заслуговує система розширень. За допомогою додаткових модулів можна розширювати функціональність редактора, інтегрувати

					КРБ.КФМН.1.710-03.2.6	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

підтримку PostgreSQL, покращувати роботу з JavaScript та автоматизувати окремі етапи розробки. Це робить середовище універсальним та зручним інструментом для створення сучасних програмних застосунків.

Таким чином, вибір Visual Studio Code обумовлений його безкоштовністю, широкими функціональними можливостями, підтримкою сучасних веб-технологій та зручністю використання під час розробки книжкового інтернет-магазину.

2.3 Вибір СУБД

Тепер вирішимо питання збереження даних, оскільки це є досить важливою частиною для теми роботи. Для праці книжного інтернет-магазину необхідно організоване та надійне зберігання інформації про користувачів, книги, жанри, замовлення, вміст кошиків та зафіксовані дії на сайті для роботи рекомендаційної системи. Саме тому розглянемо відповідні системи керування базами даних, серед яких були обрані на розгляд SQLite, MySQL та PostgreSQL.

SQLite являє собою легку вбудовану базу даних, що не потребує окремого серверного процесу, добре підходить для невеликих локальних робіт, однак має технічні обмеження при одночасній роботі великої кількості користувачів, що робить її менш придатна для подальшого масштабування.

MySQL також є досить популярною серверною СУБД та використовується для створення проєктів. Проте MySQL уступає PostgreSQL, що надає ширші можливості роботи зі складними запитами, більш розвинуту підтримку цілісності даних та додаткові механізми розширення функціональності. Тому в результаті було обрано PostgreSQL.

PostgreSQL – реляційна система управління БД, з відкритим вихідним кодом. Ця система є досить поширеною в галузях розробки, що передбачає створення робіт різного рівня складності. Цей вибір обґрунтовано тим, що забезпечує надійне зберігання інформації, підтримує складні структури даних та надає широкі можливості для реалізації взаємозв'язків між компонентами елементів архітектури.

					КРБ.КФМН.1.710-03.2.6	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

Оскільки особливістю книжкового інтернет-магазину є наявність значної кількості взаємопов'язаних об'єктів, то втілені ті самі об'єкти будуть користувачі, книги, жанри, кошики, замовлення та історія взаємодії клієнтів із товарами. Для коректної праці системи необхідно забезпечити цілісність таких зв'язків і виключити можливість появи некоректних записів. PostgreSQL містить механізми первинних та зовнішніх ключів, що дозволяють контролювати взаємозв'язки між таблицями та підтримувати узгодженість даних.

Важливою перевагою PostgreSQL є підтримка нормалізованої структури бази даних, завдяки цьому інформація зберігається без надлишкового дублювання, зменшуючи обсяг пам'яті, необхідної для зберігання даних, а також спрощуючи їх оновлення та супровід. В цьому проєкті нормалізація використовується для організації окремих таблиць користувачів, книг, замовлень, жанрів та взаємодій, що забезпечує логічну структуру бази даних.

Ще однією важливою характеристикою PostgreSQL – підтримка індексування. Індеси дозволяють значно прискорити пошук інформації в таблицях, що особливо важливо для інтернет-магазину, оскільки у майбутньому окрім виведення каталогу буде розроблятися пошукувач книг за ознакою назви та жанру. В таких ситуаціях система повинна швидко знаходити необхідні записи навіть за умови збільшення кількості товарів у каталозі. Використання індексів дозволяє підвищити швидкодію таких операцій та покращити загальну продуктивність веб-застосунку.

Також PostgreSQL підтримує виконання складних SQL-запитів з використанням об'єднання таблиць, вкладених запитів, агрегатних функцій та сортування результатів, що є особливо корисним для реалізації рекомендаційної системи, оскільки її робота передбачає аналіз накопиченої інформації про вподобання користувачів та пошук книг, які можуть бути для них найбільш цікавими. Наявність розвинених засобів роботи з даними дозволяє виконувати такі операції безпосередньо на рівні бази даних.

Окрім того, ця СУБД надає високий рівень безпеки збереження інформації через підтримку механізму розмежування прав доступу, що дозволяє визначати

					КРБ.КФМН.1.710-03.2.6	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

дозволені операції для різних категорій користувачів. Таким чином можливо обмежувати доступ до службових даних та забезпечувати захист інформації від несанкціонованих змін.

Важливим фактором під час вибору PostgreSQL стала її здатність ефективно працювати зі значними обсягами інформації та можливість подальшого масштабування системи. Наприклад у випадку потреби у розширенні функціональності інтернет-магазину або збільшення кількості користувачів структура бази даних може бути доповнена новими таблицями та зв'язками без необхідності кардинальної перебудови вже існуючої системи.

Таким чином, обрана система керування базами даних відповідає всім потребам проєкту, що необхідні для початку проєктування та реалізації рекомендаційної системи, бо додатково має широкі можливості для інтеграції різноманітних технологій та систем, що так потрібно для серверної частини. Тепер коли визначено архітектуру, середовище розробки та систему управління базами даних, залишилось обрати технології, які відіграватимуть важливу роль у реалізації проєкту.

2.4 Використанні технологій

Для реалізації книжкового інтернет-магазину важливо використати сучасні веб-технології для забезпечення розробки клієнтської та серверної частин системи. Поєднання обраних технологій дозволяє реалізувати повноцінний веб-застосунок із підтримкою авторизації користувачів, керування каталогом книг, обробки замовлень та формування персоналізованих рекомендацій.

Основу клієнтської частини становить мова розмітки HTML(Hypertext Markup Language), яка використовується для побудови структури веб-сторінок та розміщення елементів інтерфейсу. За допомогою HTML загалом реалізовано сторінки каталогу книг, авторизації, реєстрації, кошика, профілю користувача та адміністративної панелі. Крім того, HTML використовується для створення форм введення даних, навігаційних елементів, контейнерів відображення книг та інших компонентів інтерфейсу.

					КРБ.КФМН.1.710-03.2.6	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

Після розміщення основних елементів сторінок йде черга оформлення їх зовнішнього вигляду, для чого використовується технологія CSS(Cascading Style Sheets). Каскадні таблиці стилів дозволяють визначати кольорову схему сайту, розташування елементів, відступи, розміри блоків та інші параметри оформлення, завдяки чому забезпечується формування загального стилю усіх сторінок, що позитивно впливає на зручність використання системи.

Наступною важливою технологічною складовою клієнтської частини є JavaScript. Ця мова програмування використовується для реалізації інтерактивності веб-застосунку та забезпечення взаємодії між користувачем і системою. Це можна побачити через реалізацію обробки подій інтерфейсу, завантаження даних з серверу, відображення результатів пошуку, керування кошиком та формування рекомендацій на основі отриманої інформації.

Щодо засобу надсилання HTTP-запитів з клієнтської частини до серверної, то тут використовується JavaScript-інтерфейс – Fetch API. Цей механізм дозволяє асинхронно виконувати обмін даними між клієнтом і сервером без необхідності перезавантаження сторінки інтернет-магазину, що забезпечує швидку реакцію інтерфейсу на дії користувача

Серверна частина програмного продукту реалізована за допомогою програмної платформи Node.js, оскільки дозволяє виконувати JavaScript-код на стороні сервера поза браузером, завдяки чому підсумковий результат підтримує функціонал усіх частин магазину.

Для реалізації серверної частини були розглянуті декілька популярних технологій, серед яких Node.js, PHP та Python.

PHP є одним з найпоширеніших засобів створення веб-сайтів та інтернет-магазинів. Його перевагою є велика кількість готових рішень та тривалий час використання у веб-розробці. Але для цієї роботи його використання було визнано недоцільним, оскільки клієнтська частина вже реалізується через JavaScript, а застосування Node.js дозволяє одночасно використовувати одну мову програмування на клієнтській частині та на серверній частині.

					КРБ.КФМН.1.710-03.2.6	Арк.
						26
Змн.	Арк.	№ докум.	Підпис	Дата		

Python також є популярним інструментом для праці з сервером завдяки фреймворкам Django та Flask. Однак для реалізації інтернет-магазину його переваги не є критичними, а використання додаткової мови програмування ускладнило б структуру проєкту.

З огляду на це було обрано Node.js, оскільки він забезпечує єдине середовище розробки, підтримує асинхронну обробку запитів та добре інтегрується з іншими використаними технологіями проєкту.

Своєю чергою налаштування серверної логіки виконується з використанням фреймворку Express.js. Express.js значно спрощує створення веб-сервера, реалізацію REST API та створення маршрутів для роботи з книгами, користувачами, кошиком, замовленнями та рекомендаційною системою. Фреймворк також забезпечує підтримку middleware-компонентів, що перевіряють права доступу, аналізують HTTP-запити, обробляють помилки та реалізують механізми безпеки.

Наступною використаною технологією є JWT (JSON Web Token). Вона застосовується для реалізації механізму авторизації користувачів. Після успішного входу до системи сервер формує токен доступу, який містить інформацію про користувача. Далі цей токен використовується для підтвердження особи користувача під час виконання захищених операцій.

Таким чином, обраний набір технологій повністю відповідає поставленим вимогам до роботи. Користувач буде взаємодіяти з HTML-сторінками, оформленими за допомогою CSS. JavaScript забезпечуватиме обробку подій та надсилання запитів до сервера. Серверна частина має реалізовуватися на Node.js та Express.js, і буде виконувати необхідну бізнес-логіку та взаємодіє з базою даних PostgreSQL. Отримані результати повертатимуться клієнту у форматі JSON та відображаються в інтерфейсі веб-застосунку.

Обраний набір технологій забезпечує стабільну працю та створює основу для майбутньої реалізації та розвитку функціональних можливостей сайту з поступовим розширенням. Після вибору архітектурного підходу, інструментів

					КРБ.КФМН.1.710-03.2.6	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

розробки та технологій можна перейти до етапу проектування системи, де вже буде йти реалізація безпосередньо самого інтернет-магазину.

Для формування персоналізованих рекомендацій у роботі вирішено використовувати метод контентної фільтрації. Цей підхід був обраний через особливості книжкової продукції, за якою кожна книга має визначений набір характеристик для порівняння схожості. На відміну від колаборативної фільтрації, контентний підхід не потребує великої стартової кількості користувачів та накопиченої статистики, що дозволяє рекомендаціям формуватися навіть для нових користувачів після виконання перших взаємодій із каталогом.

Для побудови рекомендаційної системи використовуються дані про книги та дії користувача. До характеристик книг належать назва, жанр, автор, формат, текстовий опис, ціна та оцінка. Під час аналізу вподобань користувача враховуються історія переглядів книг, додавання товарів до кошика та виконані покупки. На основі накопиченої інформації система визначає книги зі схожими характеристиками та формує пропозиції.

					КРБ.КФМН.1.710-03.2.6	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

3. ПРОЄКТУВАННЯ СИСТЕМИ

3.1 Визначення вимог до системи. Інформаційна модель системи

Цей розділ присвячений проєктуванню книжкового інтернет-магазину перед його програмною реалізацією. На першому етапі проєктування визначимо концептуальні вимоги до системи. Одним з важливих елементів інтернет-магазину є наявність літературного каталогу, який стабільно відображає усі книги, що були внесені в базу даних. Окрім стандартних процедур повинно надаватися можливість пошуку книг, що виконується як за назвою, так і за жанром на окремій сторінці.

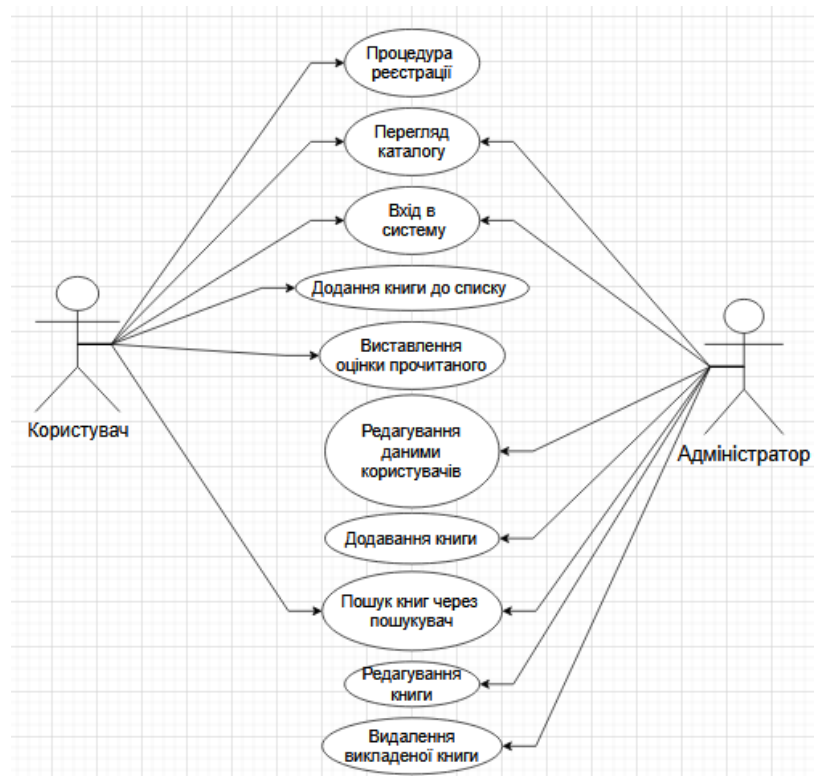


Рисунок 3.1 – Діаграма варіантів

Далі йде реєстрація та авторизація користувачів з розподілом ролей. На сайті можливо перебувати лише в одному з трьох станів – неавторизованому, клієнтському та адміністраторському. Між цими станами можна перемикатися через вихід або вхід в обліковий запис в розділі “Логін”. У неавторизованому

стані неможливо додавати книги у кошик та заходити у профіль, єдина доступна дія – перегляд каталогу. Якщо запис відсутній, то залишається зареєструватися лише як клієнт.

Авторизований користувач (клієнт) своєю чергою може перевіряти свій профіль та працювати з кошиком. Профіль клієнта містить перелік товарів, що були куплені у каталозі. Функції адміністратора відрізняються від функцій клієнта. По-перше адміністратора неможливо зареєструвати через сайт, це робиться лише через панель керування адміністратора чи напряду через СУБД. По-друге адміністратор не може взаємодіяти з кошиком та книгами так, як клієнт. Всі його спроби зробити це будуть автоматично проігноровані. По-третє профіль адміністратора містить спеціальну панель керування, що дозволяє виконувати операції додавання та видалення користувачів чи книг не залишаючи магазин та без безпосереднього доступу до БД.

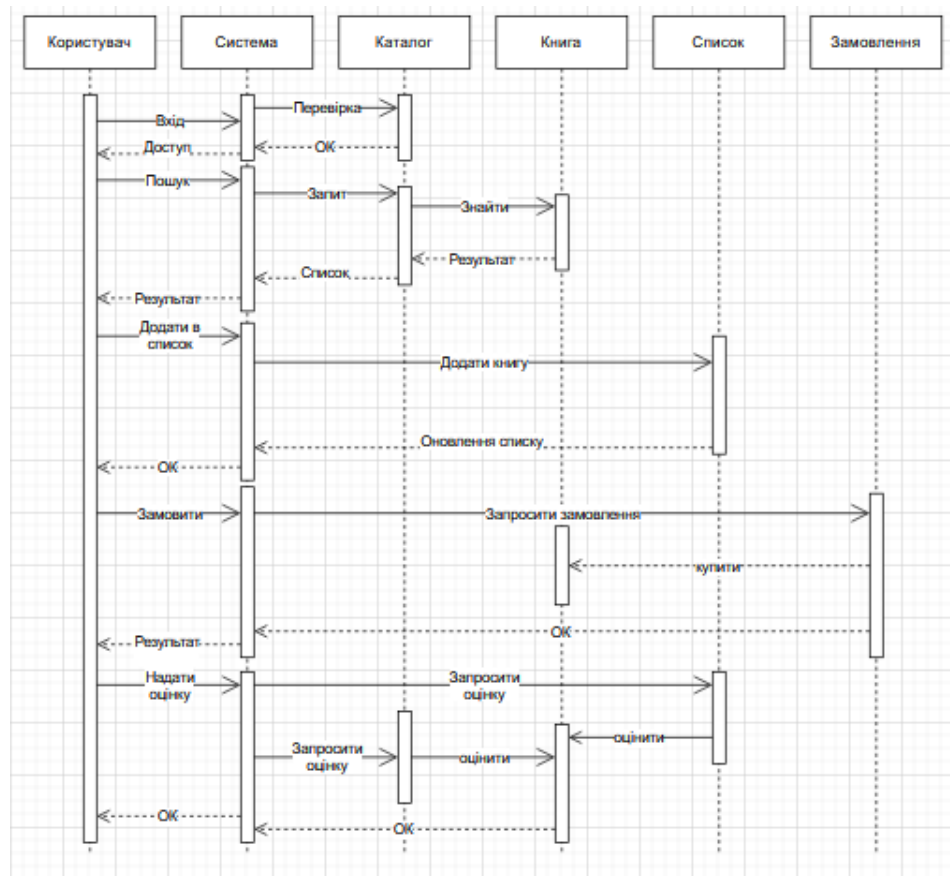


Рисунок 3.2 – Діаграма послідовності

Тепер перейдемо до самої ключового елементу, якому присвячено диплом – рекомендаційної системи. Як говорилося раніше, користувач може переглядати сторінки книг, додавати в кошик та купувати. Усі ці дії при виконанні записуються у базу даних з різними оцінками у системі ваг. Найслабша дія за таким оцінюванням – перегляд книги. Середньою вважається додавання у кошик. А найбільш пріоритетна дія – придбання книги. Таким чином розміщення даних дій у такому пріоритеті дозволяє формувати більш релевантні рекомендації. Для прикладу якщо переглянути сторінку книги жанру антиутопія та після цього додати у кошик книгу жанру фентезі, то за ієрархією пріоритетів ваг перш за все в рекомендаціях буде показуватися фентезі. Саме таким чином повинна працювати система.

3.2 Проєктування бази даних

Перейдемо до формування БД, що як вказувалося раніше в обґрунтуванні технологій, буде виконуватися в системі управління базами даних PostgreSQL.

Таблиця 3.1 – Спроектовані таблиці та поля

Назва таблиці	Назва поля
Користувач(user)	ID, ім'я, пошта, пароль, роль
Книги(books)	ID, назва, автор, жанр, опис, формат файлу, оцінка, ціна
Кошик(carts)	ID, ID користувача
Елементи кошика(cart_items)	ID, ID списку, ID книги, кількість
Замовлення(orders)	ID, ID користувача, дата, статус
Вміст замовлення(order_items)	ID, ID замовлення, ID книги, кількість, ціна
Жанри(genres)	ім'я

Зв'язок книг та жанрів(book_genres)	ID книги, ID жанру
Таблиця взаємодій(user_book_interactions)	ID користувача, ID книги, дія, створено в

Почнемо з таблиці users. Ця таблиця є однією з основних. Її роль полягає в тому, щоб зберігати інформацію про користувачів. До цих даних належать нікнейм, пароль, пошта та роль. Пошта та пароль є універсальними елементами таблиці, що дозволяють входити у власний акаунт та взаємодіяти з магазином. Роль, своєю чергою, визначає, які можливості та права користувач може мати або не може. Така концепція може мати лише одно з двох значень – «user» та «admin».

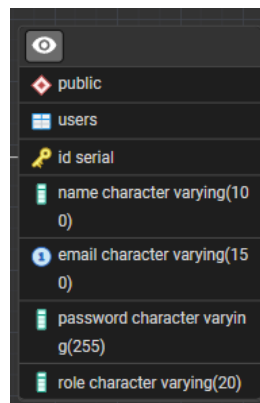


Рисунок 3.3 – Таблиця users

Якщо в ролі буде вказано «user», то користувач системи буде мати можливість взаємодії з книгами, сторінкою кошика та профілю. У випадку встановлення «admin» йде надання прав адміністратора з можливістю користуватися панелю керування, розташованій у профілі, але не може працювати з кошиком та придбання книг як клієнт. У випадку якщо реєстрація або вхід не було виконано, то користувач знаходиться у неавторизованому стані.

Наступним йде таблиця books.

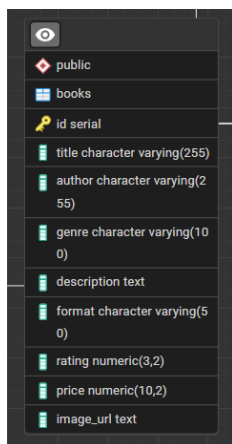


Рисунок 3.4 – Таблиця books

Таблиця books містить інформацію про книги, яка використатиметься для каталогу та рекомендаційної системи, тобто назва книги, автор, жанр, опис, формат, оцінка, ціна та адреса та посилання на обкладинку. Основним, на що треба звернути увагу, – жанри. Жанри потрібні для рекомендаційної системи як джерело даних на основі яких потрібно будувати схожу літературу. Реалізовано вони у таблиці genres, що має список усіх доступних жанрів книг. Оскільки одна книга може належати одночасно різної тематики, було створено додаткову таблицю book_genres, яка реалізує зв'язок «багато до багатьох» між книгами та жанрами.

Для запису та фіксації даних дій користувача використовується таблиця user_book_interactions. Поле дії(action) збирає інформацію про тип виконаної процедури для ідентифікації його пріоритету за системою ваг. Так у таблиці присутнє поле created_at, що при виконанні дії автоматично вписує в себе її дату та час.

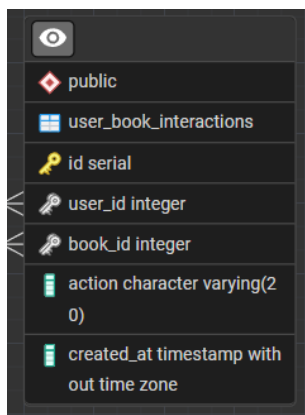


Рисунок 3.5 – Таблиця user_book_interactions

Таблиці orders та order_items потрібні для реалізації оформлення замовлень продуктів. У orders вписано загальну інформацію про замовлення, яке користувач виконав на сайті, а таблиця order_items є переліком усіх книг у складі певного замовлення. Останім залишилось реалізувати кошик. Для цього було створено carts та cart_items. Суть carts полягає в тому, що вона зберігає інформацію про кошик користувача при доданні нового товару, а cart_items зберігає самі книги, , що були додані до нього, та їх кількість.

4. ОПИС ЕТАПІВ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ПРОДУКТУ

4.1 Архітектура та файлова структура програмного комплексу

Для зручності організації було вирішено поділити каталог на дві частини: серверну частину та клієнтську. Почнемо з серверної частини. Вона являє собою архітектуру, організовану за принципами стилю REST API, що дозволяє реалізувати зв'язок та взаємодію бази даних з клієнтською частиною магазину. Серверна частина відповідає за обробку бізнес-логіки, автентифікацію та за виконання певних операцій над об'єктами системи. Реалізація цих процедур здійснена за допомогою використання файлів JavaScript, що, своєю чергою, забезпечується Node.js.

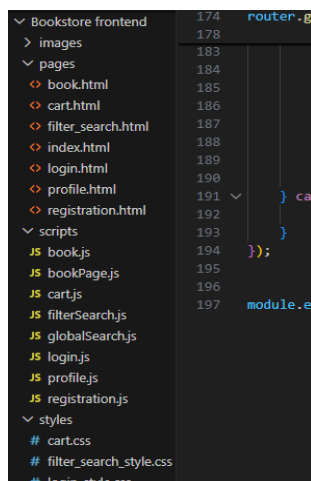


Рисунок 4.1 – Каталог файлів

Клієнтська частина містить візуальну та програмну частини проекту з якими взаємодіятимуть клієнти. Ця розділ реалізована в окремих файлах через використання мов HTML та CSS, за допомогою яких були реалізовані каркас та стилі сторінок інтернет-магазину. Додатково використано мову JavaScript для реалізації певного функціоналу окремих та загальних сторінок. Основними компонентами клієнтської частини є каталог для відображення книг, панель навігації, система загального пошуку, пошуку з фільтрами, сторінка реєстрації даних користувача в систему, сторінка входу до облікового запису та персональний користувацький профіль.

					КРБ.КФМН.1.710-03.2.6	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

4.2. Програмна реалізація серверної логіки та взаємодії з БД

Спочатку позначимо як працює автнефікація. Це здійснюється через використання стандарту JWT(JSON Web Token), що забезпечує безпечний обмін інформацією між клієнтом та сервером у вигляді JSON. Після успішного входу система надає користувачу токен доступу, який буде збережено на стороні клієнта та передається у вигляді HTTP-заголовку для кожного запиту.

Почнемо з файлів, що відповідають за серверну частину. Головним є `server.js`, який відповідає за ініціалізацію сервера, використання `middleware` та впровадження маршрутизації API-запитів. Для підключення до самої бази даних використовується спеціальний модуль підключення, який реалізований у файлі `index.js`, розміщеному у папці `db`. Далі слід приділити увагу наступним каталогам серверної частини:

- `routes` – каталог з модулями маршрутизації HTTP-запитів для доступу роботи з авторизацією, реєстрацією, клієнтами, кошиком, книгами та замовленнями тих самих книг;
- `middleware` – містить декілька модулів для перевірки JWT-токенів та контроль прав користувачів в системі;
- `public` – папка для збереження файлів обкладинок книг. Система звертається до них через посилання, яке вона бере через таблицю `books` в базі даних з колонки `image_url`.

Пряма взаємодія серверної частини з базою даних виконується через `sql`-запити, що надсилаються до СУБД та звертається до таблиць. Також система підтримує рольове розділення прав між користувачами, що реалізовано у вигляді різних механік взаємодії з магазином клієнту та адміністратора. Тепер опишемо файли, відповідаючі за роботу з серверною логікою, що реалізовані на стороні клієнтської частини.

login.js

Цей файл відповідає за авторизацію користувача на у свій акаунт. `login.js` надсилає POST-запит до `/api/auth/login` та отримує JWT-токен, який зберігається у `localStorage`.

					КРБ.КФМН.1.710-03.2.6	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

registration.js

Файл реєстрації реалізує виконання процедури додавання нового запису в список особистих записів з виконанням автоматичного входу у новостворений акаунт. Внесення даних виконується через надсилання запиту до маршруту `/api/auth/register`.

cart.js

Задача `cart.js` полягає у взаємодії з API кошика, що стосується з отримання вмісту кошика, видалення обраної книги з кошика та оформлення покупки доданої у кошик книги.

profile.js

Профіль описує персональну сторінку облікового запису, дозволяє тим відображати історію придбань чи надавати доступ до панелі керування.

4.3. Реалізація інтелектуального модуля формування рекомендацій

Темою наступного підрозділу є рекомендаційний модуль, що відповідає за персоналізацію діяльності користувача з сайтом та створення індивідуальних рекомендацій книг, що відповідають вподобанням. Надання запропонованої літератури здійснюється через асинхронну взаємодію з серверним API та відображається на сторінках у спеціальних HTML-контейнерах `#recommendations_container` та `#similar_container`. У `#recommendations_container` виводиться сам список рекомендацій, що побудований на основі зафіксованого вмісту таблиці `user_book_interactions`, а `#similar_container` розміщується на сторінці для перегляду певної книги та містить список літератури, що відповідає за жанрами з тією самою відкритою книгою.

Також додатково важливими файлами модулю, є `book.js`, `globalSearch.js` та `filterSearch.js`.

```
div.innerHTML = `
```

```
<div class="recommendation_book_block">
```

```

```

					КРБ.КФМН.1.710-03.2.6	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        <h4>${book.title}</h4>
        <p>score: ${book.score.toFixed(2)}</p>
        <div class="genres">
            ${genresHTML}
        </div>
    </div>
    `;

```

Файл book.js відповідає за відображення сторінок книг, працює з виведенням схожої літератури та також за взаємодію користувачів з рекомендаційною системою, а наступні файли globalSearch.js та filterSearch.js відрізняються тим, що вже реалізують механіку пошуку. Тобто пошук поділений на два типи – глобальний та за фільтрами. Глобальний пошук доступний на панелі навігації сайту й може бути виконаний знаходячись на всіх сторінках магазину. Своєю чергою пошук за фільтрами можна виконати лише на спеціальній окремій сторінці при веденні назви книги та обранні чекбоксу потрібного жанру.

4.4. Програмна розробка клієнтського інтерфейсу

Клієнтський інтерфейс розбитий на окремі сторінки, що відрізняються зовнішнім виглядом та відповідають за виконання різних операцій

index.html

Головна сторінка має каталог книг та розділ для рекомендацій, в які буде виводитися сформовані рекомендації. Для отримання списку книг використовується скрипт book.js, що надає список усіх книг, які завантажуються у #books_container для каталогу чи у контейнер для пропозицій. Усі книги, що виводяться мають власну візуальну стилізацію. Слід вказати про наявність числової оцінки релевантності, що мається у кожного екземпляру книги та плюсується у разі співпадіння з жанрами літератури, що сподобалась клієнтам.

					КРБ.КФМН.1.710-03.2.6	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

login.html

Задача авторизації – надані користувачеві можливості увійти у існуючий в базі даних персональний акаунт, який виконується скриптом login.js. Сторінка мкладається з полів для введення пошти та власного паролю. У випадку якщо до акаунту потрібно отримати доступ адміністратору, то процедура аналогічна, але вже потрібно поставити галочку у чекбоксі “Зайти як адмін”.

registration.html

За створення та додання нового запису відповідний файл registration.html, який також через власний JavaScript впроваджує виконання операції на сайті. Ця реєстрація дозволена лише для користувача без будь-якої можливості реєстрації адміністратора через цю сторінку.

profile.html

profile.html надає доступ, будь це клієнт або адміністратор, до власної сторінки електронного запису. Клієнт коли буде заходити на сторінку профілю побаче список придбаних книг. Для адміністратора ж буде виведено інтерфейс панелі керування. Праця перахованих можливостей залежить від того, чи була виконана авторизація.

cart.html

У сторінці cart.html втілені утілено графічну частину сторінки кошика, яка відображає усі книжки, які були туди додані. Книги, що виводяться на цій сторінці, можна купувати, відправляючи їх таким чином в список куплених заказів за допомогою скрипту cart.js.

filterSearch.html

Для можливості пошуку за фільтром у прокті виділена окрема сторінка, на якій пристуня форма для введення назви літератури та нижє є чекбокси з усіма жанрами книг. Така сторінка корисна у випадку якщо користувач не має бажання листати увесь каталог для пошуку книги потрібного жанру.

					КРБ.КФМН.1.710-03.2.6	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

4.5. Тестування працездатності та рекомендацій

4.5.1 Процес реєстрації та авторизації користувачів

Почнемо тестування реєстрації. Для цього використаємо окрему сторінку для реєстрації клієнта та панелі керування адміністратора. На сторінці реєстрації введемо дані клієнта, якого назвемо test1, а в панелі керування внесемо дані для створення запису користувача test2.

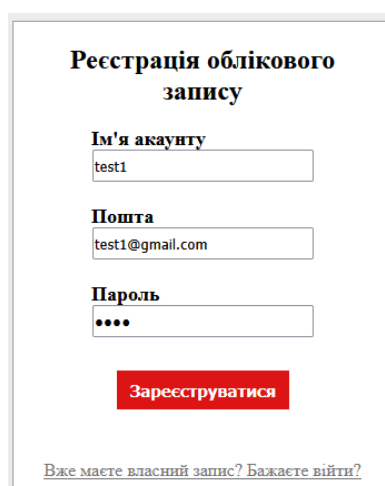


Рисунок 4.2 – Внесення даних для реєстрації

Якщо після виконання реєстрації спробувати зайти в СУБД та перевірити вміст таблиці users, то ми можемо побачити що серед всіх користувачів з'явилися нові, а саме ті, яких було додано. Лише тоді, коли новий запис створено та занесено клієнт може увійти в систему. Після автентифікації розділ з рекомендаціями буде залишатися порожнім, бо у базі даних відсутня будь-яка інформація дій в магазині. При відправці даних через форму реєстрації роль створеного користувача автоматично виставляє значення “клієнт”, але не в адміністративній панелі. Панель вже може обирати яку роль надати. Якщо з деяких причин панель не може створити адміністраторський акаунт, то цю процедуру можна виконати безпосередньо через використання СУБД.

test1	test1@gmail.com	\$2b\$10\$WgJLqJXcDUcoAEkpp3sot.DbpvaeoY41gXNa9d4p8xsMornBZI...	user
test2	test2@gmail.com	\$2b\$10\$DQc5R53IsIJYwKi/VgYOV.93w6xuq5gj.HqjwCCrvgaNOWew8h...	user

Рисунок 4.3 – Результат реєстрації

					КРБ.КФМН.1.710-03.2.6	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

Тепер спробуємо увійти в запис test1. Для цього ми відкриваємо сторінку “Логін” через панель навігації. Виконавши перехід до сторінки вводимо дані своєї пошти та паролю, які були вказані на формі реєстрації. Для тієї ситуації, коли акаунт, в який здійснюється вхід, є адміністраторським нижче використовується плажка з надписом «Зайти як адмін». При схваленні входу системою, з’являється можливість працювати з магазином без блокування операцій з книгами та переходу до сторінок кошика чи профілю.

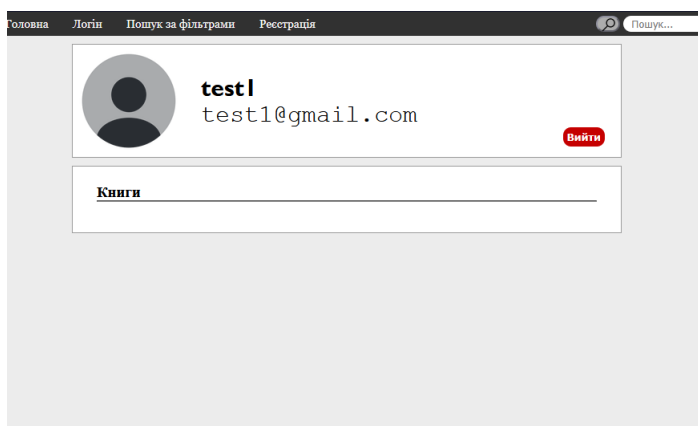


Рисунок 4.4 – Вхід у запис

Вхід у запис здійснений та закінчується пересиланням до сторінки профілю. Цей профіль поки не містить будь якої купленої книги, але протягом тестування далі тут з’являться книги, при умові того що акаунт користувацький, бо у іншому випадку буде неможливо покупати книги. Перейдемо до тестування з вмістом магазину.

4.5.2 Тестування взаємодії користувача з функціоналом магазину

Перше, що користувач побаче – список усієї літератури, що виводиться з БД. Кожна книга містить хоча б один певний жанр, що також записано у БД. Не обов’язково обмежуватися лише одним жанром, вони здатні перетинатися. До основних відносяться фентезі, антиутопія, наукова фантастика, детективи, та трилери. Уся література виводиться з власною інформацією. Можливо побачити обкладинку, назву, ім’я автора, ціну, оцінку, текстове описання та формат.

					КРБ.КФМН.1.710-03.2.6	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

Спробуємо натиснути на будь-яку книжку з каталогу. Це відкриє нам її персональну сторінку, на якій окрім характеристик присутній список літератури, який схожий за жанрами до книги, що ми відкрили.

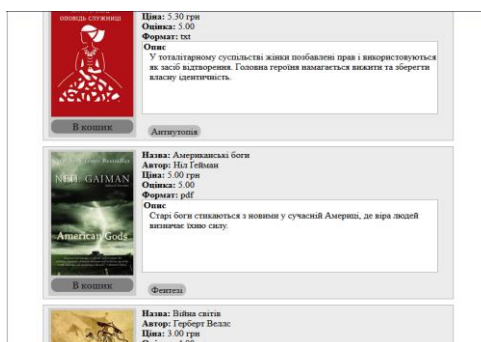


Рисунок 4.5 – Книги

Тепер додамо книгу у кошик. Це можна зробити або через каталог, або через сторінку книги. В результаті обраний товар буде збережено та розміщено на окремій сторінці, що відображає вміст кошика. Якщо натискаємо на кнопку видалити, то система видаляє книгу з кошика користувача та більше не виводить його на відповідній сторінці. У випадку якщо користувач вирішив придбати товар, то цей самий товар видаляється з кошика та переходить в список замовлень, що можна продивитись на сторінці профілю. Після придбання неможливо видалити замовлену літературу з вашого профілю. Виконання перелічених дій з кошиком та купівлею фіксується системою та заноситься у БД, відразу формуючи рекомендації, що тепер будуть завжди відображатися для профілю, в якому ви працювали з товарами.

4.5.3 Тестування рекомендаційної системи

У минулому підпункті ми виконали перегляд, додавання в кошик та придбання книжного товару. Як вже було вказано, система сформувала рекомендовану літературу. Саме для коректного та грамотного розрібру оглянемо усі рекомендації поетапно після відкриття сторінки книг, додання у кошик та їх покупки.

					КРБ.КФМН.1.710-03.2.6	Арк.
						42
ЗМН.	Арк.	№ докум.	Підпис	Дата		



Рисунок 4.6 – Рекомендації після перегляду книги

Ці пропозиції відповідають схожості характеристик переглянеої книги, у нашому випадку це була фентезі, тому відповідно бачимо книги жанра фентезі у розділі запропонованого. Можемо бачити, як оцінка релевантності score, надала одиницю до всіх книг. Тепер спробуємо додати до кошика щось іншого жанру для зміни пріоритетної тематики, наприклад нехай це буде антиутопія. Листваємо нижче, обираємо що завгодно, що є антиутопією та натискаємо на кнопку “В кошик”. Після минулих виконаних оновлюємо головну сторінку магазину.



Рисунок 4.7 – Рекомендації після додавання до кошика

В оновленому рядку бачемо, що вміст рекомендованих книг оновився та тепер замість книжок жанров фентезі ми основному можемо бачити антиутопії. Можна простежити, що у оцінки релевантності вже більше значення, що є результатом праці з кошиком. Слід ще помітити, що найбільшу оцінку виставлено для літератури, яка поєднує загальні жанри, якими користувач цікавився. Це можна простежити через те, що найбільшу оцінку отримала “Лабіринт Фавна”, бо містить в собі одночасно фентезі та антиутопію. Тепер

					КРБ.КФМН.1.710-03.2.6	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

простежимо за тим, як система відреагує на придбання. Для цього заходимо на сторінку кошика та натискаємо кнопку придбання у тому товарі, що сюди нещодавно занесли(антиутопія).

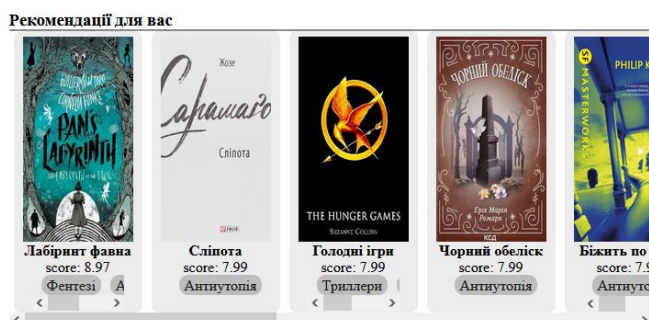


Рисунок 4.8 – Рекомендації після придбання

Ну й відповідно оновивши сторінку бачемо, що оцінка релевантності зросла у найвище число, що привело до видання пропозицій, які пріоритетно базуються на антиутопічних книгах. Також не ігнорується фактор взаємодії з фентезі у минулому, що при купівлі книги так само як і при додаванні до кошика будує рекомендації при можливості з минулими уподобаннями.

4.5.4 Тестування панелі керування адміністратора

Наступним та останнім залишилося перевірити функціонал адміністратора. Оскільки тепер нам не потрібен акаунт користувача, то виходимо з нього та виконуємо автентифікацію у запис адміністратора. Для тестування спробуємо виконати видалення створених нами записів користувачів через введення даних та через вибір режиму видалення користувачів. Видаєння або додання можливе лише при веденні даних назви користувача, пошти, паролю та ролі. Якщо потрібно когось видалити, то на відміну від додання адміністратор повинен обов'язково ввести його ID, що ми й відповідно виконуємо.

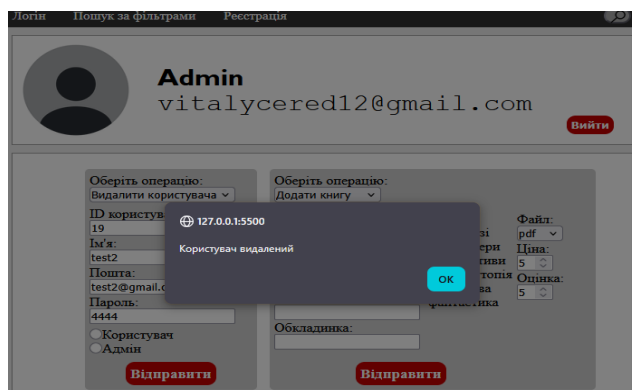


Рисунок 4.9 – Видалення користувача

Коли усі поля були заповнені, натискаємо на кнопку “Відправити”. У разі якщо клієнт за вказаними даними присутній у БД, то виводиться повідомлення про успішне видалення, чи про відсутність такого користувача. Відповідно якщо подивитися вміст БД, то його більше не буде там бачно. Тепер виконаємо тестування додання нової книги. На відміну від полів користувача тут окрім звичайних текстових полів пристуня обкладинка. За причиною того, що книги у цій роботі не є реальними, а лише елементами даних з інформаційними характеристиками, обкладинка є jpg зображенням до якого отримується доступ крізь посилання до файлу картинки у каталозі. Тому перед заповненням полів шукаємо jpg обкладинку, нехай це буде Робін Гуд. Підготувавши обкладинку обов’язково запам’ятовуємо її ім’я файлу, інакше при розбіжності нічого не виведеться. Далі заповнюємо поля та натискаємо “Відправити”. Нова книга додана

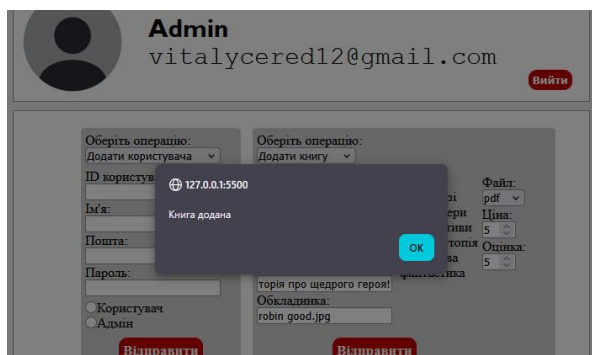


Рисунок 4.10 – Додавання книги

					КРБ.КФМН.1.710-03.2.6	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

5. ТЕХНІКО-ЕКОНОМІЧНА ЧАСТИНА

5.1 Організаційно-економічне обґрунтування розробки

Основою проєкту є інтернет-магазин, в якому втілена працююча функціональна рекомендаційна система, яка надає користувачам рекомендації, створені на основі їх уподобань. В результаті користувач повинен отримати власні рекомендації книг, що формуються за його взаємодією з користувачем. Описана взаємодія являє собою перегляд сторінок книг, додавання їх у кошик та придбання. Ці дії мають різні пріоритети, що фіксуються системою та дозволяє адаптуватися до поведінки користувача в магазині та підлаштуватися.

Таблиця 5.1 – Порівняльна характеристика методів рекомендаційних систем

Назва методу	Переваги	Недоліки
Контентна фільтрація	Не вимагає великої кількості користувачів для початку роботи системи; дозволяє формувати рекомендації на основі жанру, автора, опису та інших характеристик. Добре підходить для книжкового інтернет-магазину, де кожний товар має жанри та інші ознаки.	Може обмежувати користувача товарами, схожими на ті, з якими він вже взаємодіяв. Якість рекомендацій залежить від повноти опису книг. Система гірше виявляє нові вподобання користувача та також можливе повторення однотипних рекомендацій
Колаборативна фільтрація	Враховує поведінку інших користувачів; може знаходити неочевидні зв'язки між товарами; добре працює у великих інтернет-магазинах з	Має проблему “холодного старту” для нових користувачів і нових товарів. Вимагає значного обсягу даних про взаємодії

	великою кількістю користувачів та історичних даних.	користувачів. Складніша в реалізації для невеликого або нового магазину.
Гібридний метод	Поєднує переваги контентної та колаборативної фільтрації. Може забезпечувати дуже високу точність рекомендацій. Частково зменшує проблему холодного старту. Дас змогу враховувати як властивості товару, так і поведінку користувачів.	Вимагає складнішої архітектури програмного продукту та більшого обсягу даних. Складніше тестується та налаштовується. Збільшує трудомісткість розробки.

Таблиця 5.2 – Аналіз конкурентів книжкового інтернет-магазину

Компанія та сервіси	Переваги	Недоліки
Yakaboo	Один із найвідоміших книжкових онлайн магазинів України. Широкий асортимент книг різних жанрів і мов. Сформована клієнтська база. Наявність акцій, добірок та системи пошуку.	Високий рівень конкуренції всередині каталогу. Рекомендації можуть бути менш гнучкими. Складніше виділити нові або маловідомі книги серед великого каталогу.
Книгарня «Є»	Відомий книжковий бренд. Поєднання онлайн та офлайн продаж. Орієнтованість на українську книжкову продукцію. Довіра	Значна частина конкурентної переваги пов'язана з мережею бібліотек. Онлайн-рекомендації можуть поступатися

	покупців.	спеціалізованій персоналізованій системі.
Book24	Широкий каталог книг різних жанрів. Наявність новинок, та знижок. Зручні способи оплати та доставки. Фокус на книжковій продукції.	Менше впізнавання порівняно з більшими книжковими платформами. Рекомендаційний функціонал може бути обмежений.
Наш Формат	Напір на україномовній літературі. Якісний каталог різноманітних видань. Сформована цільова аудиторія	Тематика звужує охоплення ринку. Менша різноманітність для користувачів, які шукають максимально різні жанри.
Vivat	Відоме українське видавництво та інтернет-магазин. Наявність художньої літератури, нон-фікшн, коміксів, манги, мерчу та електронних книг. Активне рух власної продукції	Основна перевага пов'язана з продукцією конкретного видавництва.
Видавництво Старого Лева	Має стійкий бренд. Якісна художня, дитяча та підліткова література. Наявність паперових та електронних книг.	Каталог переважно пов'язаний із власними виданнями. Менше можливостей для порівняння книг різних

					КРБ.КФМН.1.710-03.2.6	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

		видавництв у межах одного каталогу.
Лабораторія	Сучасне позиціонування. Наявність україномовних книг, нон-фікшн, професійних і художніх книг. Акцент на змістовності та якості каталогу	Вузький асортимент порівняно з великими маркетплейсами. Менша масовість бренду. Обмеження в охопленні різних читацьких сегментів.
Rozetka, категорія “Книги”	Велика аудиторія платформи. Зручна логістика. Довіра до маркетплейсу. Можливість купувати книги разом з іншими товарами.	Книги лише є однією з багатьох категорій. Відсутня вузька спеціалізація саме на книжкових рекомендаціях. Досвід користувача менш орієнтован на інтереси читача.
Клуб Сімейного Дозвілля	Відоме видавництво та книжковий магазин. Регулярні акції. Орієнтація на масового читача. Наявність популярної художньої літератури.	Обмеженість каталогу порівняно з іншими великими книжковими платформами. Більший акцент на власних виданнях і масових жанрах.
Prom.ua, категорія “Книги”	Велика кількість продавців. Широкий вибір товарів. Конкуренція цін. Можливість швидко	Якість описів і сервісу залежить від окремого продавця. Рекомендації для читачів менш виражені.

	знайти книги від різних постачальників.	
--	---	--

Особливість цього проєкту дозволяє ефективно працювати з великою кількістю клієнтів, що обумовлено особливістю контентної фільтрації, яка формує рекомендації персонально та їй не потрібно багата кількість користувачів для того, щоб пропозиції почали працювати. В результаті з точки зору практичного значення розроблений інтернет-магазин має значення як фундамент для повноцінного магазину, можливості якого можна розширювати та покращити.

Задача цього розділу полягає в наданні оцінки трудомісткості, собівартості його створення, ціну та економічний ефект.

5.2 Планування робіт і визначення трудомісткості розробки

Для грамотного надання оцінки трудомісткості нам буде потрібно створити таблицю, в якій буде описано етапи роботи над проєктом, вписати в нього затрачений час та використати його для розрахунку.

Таблиця 5.3 – Етапи виконання роботи

Назва та номер етапу	Зміст	Трудомісткість, год
1. Аналіз предметної області	Аналіз рекомендаційних систем, електронної комерції та аналогічних прикладів	17
2. Формування вимог	Визначення функцій користувача, адміністратора, каталогу, кошика та рекомендацій	14
3. Проєктування	Побудова структури	43

					КРБ.КФМН.1.710-03.2.6	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

архітектури та БД	системи, визначення таблиць і зв'язків	
4. Розробка серверної частини	Реалізація API для книг, користувачів, кошика, замовлень та авторизації	39
5. Розробка клієнтської частини	Обмеження прав доступу, додавання та видалення книг і користувачів	46
6. Розробка рекомендаційної системи	Формування профілю користувача та підбір книг за жанровими ознаками	33
5. Реалізація кошика, профілю та покупок	Обробка додавання товарів, історії купівель і профілю користувача	42
6. Тестування та налагодження	Перевірка основних сценаріїв роботи системи, усунення помилок	23
7. Підготовка документації	Опис програмного продукту, техніко-економічна частина, оформлення матеріалів	58

5.2.1 Мережевий графік виконання робіт

Для визначення послідовності виконання робіт та надання загальної оцінки тривалості розробки було сформовано мережевий графік проєкту. Мережевий графік дозволяє відобразити взаємозв'язки між окремими етапами розробки програмного продукту та визначити послідовність їх виконання. Також такий підхід дає можливість оцінити критичні етапи, від яких залежить загальна тривалість виконання роботи.

					КРБ.КФМН.1.710-03.2.6	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

Тривалість кожного етапу визначалась на основі трудомісткості робіт та їх складності. Послідовність виконання робіт наведена в таблиці 5.4.

Таблиця 5.4 – Графік виконання робіт

№ коду роботи	Найменування роботи	Тривалість (дні)
0-1	Аналіз предметної області	2
1-2	Формування вимог	2
2-3	Проектування архітектури та БД	5
3-4	Розробка серверної частини	5
4-5	Розробка клієнтської частини	6
5-6	Розробка рекомендаційної системи	4
6-7	Реалізація кошика, профілю та покупок	5
7-8	Тестування та налагодження	3
8-9	Підготовка документації	7

В результаті на отриманих побудовано мережевий графік з візуалізацією зв'язків та шляху.

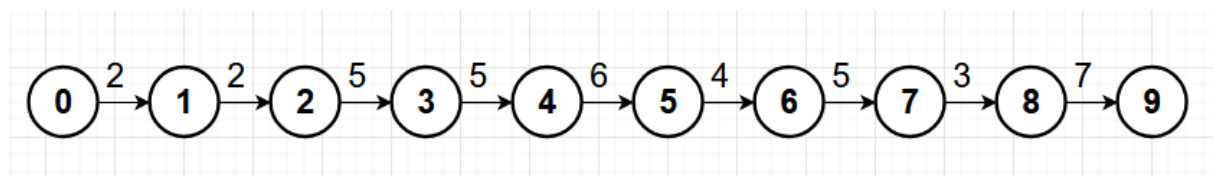


Рисунок 5.1 – Мережевий графік

5.2.2 Визначення трудомісткості розробки програмного продукту (ПП)

Отримана трудомісткість загально буде дорівнювати 315:

$$T_{\Sigma} = 315(\text{год})$$

Тепер визначимо тривалість розробки в людино-днях:

$$T_{\text{дн}} = \frac{315}{8} = 39,38(\text{людино-днів}),$$

де 8 годин – тривалість робочого дня.

Тепер порахуємо з урахуванням коефіцієнта переведення в календарні дні:

					КРБ.КФМН.1.710-03.2.6	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

$$T_{\text{кал}} = \frac{315}{8 \cdot 0,73} = 53,9(\text{календарних днів})$$

В результаті ми вираховували, що кількість днів, витрачених на проектування, розробку та оформлення, буде приблизно дорівнювати 54 календарних днів.

5.3 Розрахунок собівартості та ціни розробки

Наступне, що нам потрібно визначити – собівартість. Для розрахунку на потрібно урахувати декілька змінних, а саме умовна заробітна плата для розробника, витрати на керівництво та консультацію, додаткова заробітна плата, відрахування на соціальні заходи, витрати на використання обчислювальної техніки, матеріальні та накладні витрати.

Для розрахунку з місячним окладу використаємо умовний оклад у 30000 грн. Також використаємо середню кількість робочих днів у місяці та отриману кількість людино-днів:

$$C_{\text{зп}} = \frac{30000 \cdot 39,38}{22} = 53700(\text{грн}),$$

де 22 – середню кількість робочих днів у місяці;

39,38 – кількість людино-днів.

Витрати на керівництво та консультацію приймаються в розмірі 10% від основної заробітної плати розробника:

$$C_{\text{зп}} = 53700 \cdot 0,1 = 5370(\text{грн}).$$

Загальна основна заробітна плата:

$$C_{\text{зо}} = 53700 + 5370 = 59070(\text{грн}).$$

Додаткова заробітна плата приймається у розмірі 10 %:

$$C_{\text{зд}} = 59070 \cdot 0,1 = 5907(\text{грн}).$$

Відрахування на соціальні заходи приймаються у розмірі 22 %:

$$C_{\text{сс}} = (59070 + 5907) \cdot 0,22 = 14295(\text{грн}).$$

Накладні витрати приймаються у розмірі 50 % від основної заробітної плати:

$$C_{\text{н}} = 59070 \cdot 0,5 = 29535(\text{грн}).$$

					КРБ.КФМН.1.710-03.2.6	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

Тепер наведемо матеріальні витрати. Зробимо це через таблицю.

Таблиця 5.5 – Матеріальні витрати

Найменування витрат	Кількість	Ціна за одиницю, грн	Вартість, грн
Папір для друку документації	1 пачка	180	180
Друк і брошурування матеріалів	1 комплект	500	500
USB-накопичувач	1 шт.	250	250
Канцелярські матеріали	1 комплект	300	300
Транспортно-заготівельні витрати, 10%	-	-	123

Усього матеріальних виплат за загальним числом буде 1353 грн. Витрати на використання обчислювальної техніки визначаються з урахуванням часу роботи з ЕОМ. Для розрахунку прийнято 340 годин машинного часу, вартість 1 години – 8 грн, коефіцієнт використання ЕОМ – 1,08, коефіцієнт використання СУБД – 1,1:

$$C_{\text{ЕОМ}} = 340 \cdot 1,08 \cdot 8 \cdot 1,1 = 3231(\text{грн}).$$

Таблиця 5.6 – Собівартість розробки програмного продукту

Стаття витрат	Сума, грн	Питома вага, %
Матеріальні витрати	1353	1,19
Витрати на	3231	2,85

використання ЕОМ		
Основна заробітна плата	59070	52,1
Додаткова заробітна плата	5907	5,21
Відрахування на соціальні заходи	14295	12,6
Накладні витрати	29535	26,04

Отже, кошторисна собівартість розробки становить:

$$C = 1353 + 3231 + 59070 + 5907 + 14295 + 29535 = 113391(\text{грн}).$$

Оскільки програмний продукт може бути відчужений від безпосереднього розробника та використаний як готове рішення для інтернет-магазину, ціну розробки доцільно визначати з урахуванням коефіцієнта дослідного зразка та нормативного прибутку.

Коефіцієнт обліку витрат на виготовлення дослідного зразка:

$$K = 1,1$$

Норматив рентабельності:

$$P_n = 25\%$$

Нормативний прибуток:

$$P_p = \frac{(C - C_m) \cdot P_n}{100} = \frac{(113391 - 1353) \cdot 25}{100} = 28010(\text{грн})$$

Ціна програмного продукту:

$$Ц = K \cdot C + P_p$$

$$Ц = 1,1 \cdot 113391 + 28010 = 152740(\text{грн})$$

					КРБ.КФМН.1.710-03.2.6	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

В результаті розрахунку встановлюємо, що орієнтована ціна розробки буде дорівнювати 152740 грн.

5.4 Розрахунок капітальних і поточних витрат

Капітальні витрати пов'язані з підготовкою програмного продукту до впровадження та забезпеченням мінімальної технічної бази для його експлуатації. У цьому проєкті не передбачається будівництво або реконструкція приміщень, тому відповідна стаття витрат дорівнює нулю.

Таблиця 5.7 – Розрахунок капітальних витрат

Найменування витрат	Позначення	Сума, грн
Довиробничі витрати, пов'язані з розробкою ПП	К _п	152740
Комп'ютерне обладнання для адміністрування системи	К _{ко}	28000
Допоміжне обладнання, 10% від вартості комп'ютерного обладнання	К _{во}	2800
Будівництво або реконструкція приміщень	К _с	0
Сумарні капітальні виплати	К ₂	183540

Загальні капітальні витрати знаходяться через формулу:

$$K_2 = K_n + K_{ко} + K_{во} + K_c + K$$

Знаходимо суму:

$$K_2 = 152740 + 28000 + 2800 + 0 = 183540(\text{грн})$$

Поточні витрати після впровадження системи включають витрати на технічну підтримку, хостинг, домен, амортизацію обладнання, електроенергію, ремонт і допоміжні матеріали.

Таблиця 5.8 – Розрахунок поточних витрат після впровадження

Стаття витрат	Сума, грн/рік
Технічна підтримка та адміністрування системи	32400
Хостинг і доменне ім'я	7800
Амортизація комп'ютерного обладнання	16800
Витрати на електроенергію	1800
Ремонт обладнання	1680
Допоміжні матеріали	560
Сумарні поточні витрати після впровадження	61040

Для порівняння прийнято, що до впровадження системи витрати на ручне адміністрування каталогу, добір товарних пропозицій і підтримку користувачів становили 91200 грн на рік. Після впровадження програмного продукту ці витрати зменшуються завдяки автоматизації пошуку, каталогізації, кошика, обробки взаємодій користувача та формування рекомендацій.

					КРБ.КФМН.1.710-03.2.6	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

Річна економія на поточних витратах:

$$C_1 - C_2 = 91200 - 61040 = 30160(\text{грн}),$$

де C_1 – поточні витрати до впровадження;

C_2 – поточні витрати після впровадження.

5.5 Оцінка економічної ефективності впровадження

Економічна ефективність інтернет магазину визначається не лише прямим скороченням витрат, а й приростом прибутку від використання рекомендаційної системи. Для книжного інтернет-магазину персоналізовані рекомендації можуть впливати на кількість переглянутих товарів, середню глибину взаємодії з каталогом і ймовірність додавання книги до кошика.

Для розрахунку прийнято, що річний обсяг продажів інтернет-магазину до впровадження рекомендаційної системи становить 1800000 грн. Очікуване зростання обсягу продажів після впровадження рекомендаційних систем приймається на рівні 10%. При маржинальності продажів 30% приріст прибутку становитиме:

$$\Delta\P = 1800000 \cdot 0,10 \cdot 0,30 = 54000 (\text{грн})$$

Річний економічний результат від впровадження визначається як сума економії поточних витрат і приросту прибутку:

$$E_z = (C_1 - C_2) + \Delta\P = 30160 + 54000 = 84160(\text{грн})$$

Очікуваний економічний ефект з урахуванням нормативного коефіцієнта ефективності одноразових витрат:

$$E_o = E_z - E_n \cdot K_n$$

де $E_n = 0,25$;

					КРБ.КФМН.1.710-03.2.6	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

$$K_{\Pi} = 183540 \text{ грн.}$$

$$E_o = 84160 - 0,25 \cdot 183540 = 38275 (\text{грн})$$

Коефіцієнт економічної ефективності одноразових витрат:

$$E = \frac{E_o}{K_{\Pi}} = \frac{84160}{183540} = 0,459$$

Оскільки отримане значення $E=0,459$ перевищує нормативний коефіцієнт $E_n=0,25$, проєкт можна вважати економічно доцільним.

Строк окупності одноразових витрат:

$$T = \frac{1}{E} = \frac{1}{0,551} = 2,18 (\text{року})$$

Таблиця 5.9 – Основні техніко-економічні показники проєкту

Показник	Одиниця виміру	Значення
Загальна трудомісткість розробки	год	315
Тривалість розробки	календарні дні	53,9
Кошторисна собівартість розробки	грн	113391
Орієнтовна ціна програмного продукту	грн	152740
Капітальні витрати	грн	183540
Поточні витрати після впровадження	грн/рік	61040
Річний економічний результат	грн/рік	84160

Очікуваний економічний ефект	грн	38275
Коефіцієнт економічної ефективності	-	0,459
Строк окупності	роки	2,18

5.6 Бізнес-план впровадження програмного продукту

Для початку визначимо, що цей продукт становить собою інтернет-магазин з впровадженою інтелектуальною системою рекомендацій, який здатний на основі взаємодії користувача системи сформувати пропозиції, сході за характеристиками до того, чим цікавився клієнт.

Цільовою аудиторією книжного інтернет магазину є підлітки, студенти та старші, які цікавляться літературою та люблять читати художню літературу. Наявність рекомендаційної системи дозволяє утримувати та залучати клієнтів через адаптацію до їх інтересів. Також окрім рекомендаційної системи сайт оснащений зручним функціоналом для взаємодії з літературою, тобто інструменти пошуку, списки схожих книг та інше. Основними джерелами доходу такого інтернет-магазину можуть бути:

- продаж паперових книг;
- продаж електронних книг;
- реалізація тематичних добірок і комплектів;
- продаж акційних пропозицій;
- просування новинок або книг певних видавництв;
- повторні покупки постійних користувачів.

Слід зазначити, що ключовими витратами після впровадження інтернет-магазину є технічна підтримка, адміністрація системи, оплата хостингу з отриманням доменного ім'я, забезпечення електроенергії, огляд за технікою та закупа допоміжних матеріалів. Згідно з проведеними розрахунками, поточні витрати після впровадження становлять 61040 грн на рік, тоді як витрати до впровадження системи прийнято на рівні 91200 грн на рік. В результаті очікувана річна економія поточних витрат становить 30160 грн.

Також ураховується можливий приріст прибутку через використання рекомендаційної системи. Приймаємо у розрахунок, що річний обсяг продажів інтернет-магазину до впровадження системи буде становити 1800000 грн та очікує зростання продажів у 10%, а маржинальність продажів буде очікувати 30%. Через такі умови приріст прибутку буде дорівнювати :

$$\Delta\P = 1800000 \cdot 0,10 \cdot 0,30 = 54000(\text{грн})$$

Загальний річний економічний результат визначається як сума економії поточних витрат і приросту прибутку:

$$E_{\Gamma} = 30160 + 54000 = 84160(\text{грн})$$

Капітальні витрати на впровадження програмного продукту становлять 183540 грн. За умови отримання річного економічного результату в розмірі 84160 грн строк окупності дорівнює:

$$T = 183540 / 84160 = 2,18 \text{ року.}$$

В результаті ми визначили, що впровадження книжкового інтернет-магазину з рекомендаційною системою є економічно доцільним. Проєкту вимагає початкових затрат на розробку та забезпечення технічного арсеналу, але при подальшій праці дозволяє зменшити поточні витрати, забезпечити навання адміністраторів та отримати прибуток.

					КРБ.КФМН.1.710-03.2.6	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

6. ОХОРОНА ПРАЦІ

Небезпечні фактори, що існують на робочих місцях, це ті, які становлять загрозу для життя чи здоров'я людини. Прикладом небезпечного фактору є змінний електричний струм напруги 220 В. Комп'ютер, як і будь-який електричний прилад, особливо при його неправильному підключенні, може бути джерелом ураження користувача ПК електричним струмом [1].

Шкідливі фактори, що існують на робочих місцях, це ті фактори які здатні викликати професійні захворювання, або взагалі недуги. Шкідливими факторами при роботі з персональним комп'ютером є неіонізуюче випромінювання промислової частоти, збільшене нервово-емоційне навантаження на оператора, збільшення навантаження на органи зору та дрібні стереостатичні рухи кінцівок [1].

До шкідливих виробничих факторів відносяться:

- наявність шуму та вібрації;
- м'яке рентгенівське випромінювання;
- електромагнітне випромінювання;
- ультрафіолетове і інфрачервоне випромінювання;
- електростатичне поле між екраном і оператором;
- відсутність або недолік природного світла
- пряма та відбита блискіть;
- наявність пилу, озону, оксидів азоту й аероіонізації;
- показники мікроклімату: температура повітря T , відносна вологість W ,

швидкість руху повітря V .

Ці фактори можуть викликати у працівника певні розлади здоров'я, зокрема підвищення артеріального тиску, кон'юктивіти, тендовагініти та інші захворювання.

За ступенем пожежної небезпеки це приміщення із візуальними дисплейними терміналами (ВДТ) електронно-обчислювальних машин (ЕОМ) відноситься до категорії В [2]. Це приміщення, в яких є горючі гази (ГГ),

					КРБ.КФМН.1.710-03.2.6	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

легкозаймисті, горючі і важкогорючі рідини, а також речовини та матеріали, які здатні при взаємодії з водою, киснем повітря або один з одним вибухати і горіти або тільки горіти; горючий пил і волокна, тверді горючі та важкогорючі речовини і матеріали, за умови, що приміщення, в яких вони знаходяться (обертаються), не відносяться до категорій А, Б і питома пожежна навантага для твердих і рідких легкозаймистих та горючих речовин на окремих ділянках площею не менше 10 м² кожна перевищує 180 МДж/м [2].

Розміщення робочих місць ВДТ ЕОМ в підвальних і цокольних поверхах забороняється [3]. Робоче місце знаходиться у комп'ютерному класі. Приміщення знаходиться на першому поверсі 3-поверхової будівлі. Приміщення з ВДТ має природне і штучне освітлення. Природне освітлення здійснюється через світлові отвори, орієнтовані переважно на північ, північний захід. На робочих місцях з ВДТ КПО (коефіцієнт природного освітлення) складає понад 1.5%. Приміщення з ВДТ обладнане системами опалювання, вентиляції і кондиціонування повітря. Віконні отвори обладнані регульованими пристосуваннями: це жалюзі, фіранки і зовнішні козирки [3].

Задане приміщення розміром:

- довжина - $A = 12$ м;
- ширина - $B = 15$ м;
- висота - $H = 3,2$ м.

Площа приміщень, в яких розташовують персональні комп'ютери (ПК), визначають згідно з діючими нормативними документами з розрахунку на одне робоче місце: площа – не менше 6,0 м², об'єм – не менше 20,0 м³, з урахуванням максимальної кількості осіб, що одночасно працюють в зміні. У нашому випадку площа – 180 м², а об'єм – 576 м³. З цього виходить, що максимальна кількість робочих місць рівна двадцять вісім.

Лінія електромережі для ЕОМ виконана як окрема групова трьохпровідна мережа, шляхом прокладки фазового, нульового робочого, нульового захисного проводів.

					КРБ.КФМН.1.710-03.2.6	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

Нульовий захисний провід використовується тільки для занулення електроприймача. Використання нульового робочого проводу в якості нульового захисного заборонено. Штепсельні з'єднання й розетки мають спеціальні контакти для підключення нульового захисного проводу.

ЕОМ не підключено до звичайної двохпровідної мережі, у тому числі з використанням перехідних пристосувань. Експлуатація кабелів і проводів з ушкодженою ізоляцією, саморобних подовжувачів не проводиться.

Згідно з [4], всі приміщення за ступенем небезпеки ураження електричним струмом діляться на три класи:

1. Приміщення без підвищеної небезпеки.
2. Приміщення з підвищеною небезпекою.
3. Особливо небезпечні приміщення.

Приміщення з ПК відноситься до приміщення з підвищеною небезпекою, тому що існує можливість одночасного дотику людини до будь яких з'єднання з землею, металоконструкціями будинків, технологічними апаратами, механізмами і т.п., з одного боку, і до металевих корпусів електрообладнання з іншого.

Згідно з правилами улаштування електроустановок (ПУЕ), все електрообладнання ділиться на два класи [4, 5]:

- з напругою до 1000 В;
- з напругою понад 1000 В.

Електрообладнання приміщень з ПК відноситься до першого класу, з напругою до 1000 В.

ЕОМ, периферійні пристрої та інше обладнання (апарати управління світильники і тому подібне), електропроводи і кабелі по виконанню і ступеню захисту повинні відповідати класу зони по ПУЕ [5].

Лінія електромережі для живлення ЕОМ, периферійних пристроїв ЕОМ і устаткування для обслуговування, ремонту і наладки ЕОМ виконана як окрема групова трьохдротова мережа.

					КРБ.КФМН.1.710-03.2.6	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

Приміщення, в якому розташовані ПК, має першу ступінь вогнестійкості згідно [6].

1 ступінь вогнестійкості – це будинки з несучими та огорожувальними конструкціями з природних матеріалів або штучного каменю, бетону або залізобетону із застосуванням листових і плиткових негорючих матеріалів.

Як засоби пожежогасіння на цьому об'єкті застосовуються вуглекислотні вогнегасники, призначені для гасіння спалахів установок напругою до 1000 В.

Приміщення категорії В з площею 180 м² має 4 вуглекислотні вогнегасника с заряд вогнегасної речовини 6 кг.

Приміщення з ПК має загальне рівномірне освітлення. При роботі з документами загальне рівномірне доповнене місцевим. Джерелами світла для загального рівномірного освітлення є газорозрядні лампи низького тиску, а для місцевого дозволяється використовувати лампи розжарювання.

При роботі за дисплеєм освітленість визначається мінімальним об'єктом розрізнення шириною лінії рукописного або друкарського тексту, який читає користувач з листа. Освітлення на робочому місці становить 425 Лк [7].

					КРБ.КФМН.1.710-03.2.6	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗАГАЛЬНІ ВИСНОВКИ

В рамках дипломної роботи було розглянуто особливості використання рекомендаційних систем у сфері електронної комерції та проаналізовано основні підходи до їх побудови. Було проведено порівняння існуючих типів рекомендаційних систем, їх переваг та недоліків з урахуванням особливостей книжкового інтернет-магазину.

На основі проведеного аналізу для реалізації проєкту було обрано метод контентної фільтрації. Вибір цього підходу обґрунтований тим, що дозволяє формувати індивідуальні пропозиції на основі характеристик книг та історії взаємодії користувача з товарами. Контентна фільтрація є оптимальним для літератури та не потребує великої кількості користувачів для початку роботи.

У процесі проєктування БД було розроблено структуру бази даних у СУБД PostgreSQL, що забезпечує зберігання інформації про користувачів, книги, жанри, кошики, замовлення та взаємодії користувачів із товарами. Спроектowana структура дозволяє підтримувати цілісність даних та забезпечує можливість подальшого розширення системи.

Практичним результатом роботи стала реалізація книжкового інтернет-магазину на основі клієнт-серверної архітектури. Було створено функціонал реєстрації та авторизації користувачів, перегляду каталогу книг, пошуку та фільтрації товарів, роботи з кошиком, оформлення замовлень з відображенням у профілі, а також адміністративна панель для керування.

Окремо реалізовано модуль рекомендацій, який аналізує історію переглядів, додавання книг до кошика та інші взаємодії користувача з системою. На основі отриманих даних формується показник релевантності та список пропонованої літератури, що краще відповідають інтересам користувача.

Для перевірки працездатності було проведено тестування основних функцій системи. Результати перевірки підтвердили коректність роботи реєстрації, авторизації, взаємодії з каталогом книг, роботи кошика та формування рекомендацій.

					КРБ.КФМН.1.710-03.2.6	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

Отримані результати можуть бути використані як основа для подальшого розвитку системи. У майбутньому можливе розширення рекомендаційного модуля шляхом використання гібридних алгоритмів, впровадження оцінювання книг користувачами, покращення механізмів аналізу вподобань та розширення функціональних можливостей інтернет-магазину.

					КРБ.КФМН.1.710-03.2.6	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ricci F. , Rokach L. , Shapira B. Recommender Systems Handbook: академічний текст. New York, USA: Springer , 2022. 1008 с.
2. Falk K. Practical Recommender Systems. United States. 448 с.
3. Duckett J. HTML and CSS Design and Build Web Sites. United States: John Wiley & Sons, Inc.. 478 с.
4. Dackett J. Javascript and jQuery. United States: John Wiley & Sons, Inc. 641 с.
5. Casciaro M. , Mammino L. Node.js Design Patterns 3th edition. United Kingdom: Packt Publishing. 661 с.
6. Gruber M. Understanding SQL. United States: Sybex Inc. 462 с.
7. Hsu L. , Obe R. PostgreSQL: Up and Running 3rd edidtion. United States: O'Reilly Media. 314 с.
8. Інформація про JWT // URL: <https://www.jwt.io/introduction>
9. Документація Node.js // URL: <https://nodejs.org/docs/latest/api>
10. Інформація про Express.js // URL: <https://expressjs.com/en/5x/guide/routing>
11. Інформація про Fetch API // URL: https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API
12. Документація PostgreSQL // URL: <https://www.postgresql.org/docs>
13. Кталоґ інформації JavaScript // URL: <https://javascript.info>
14. Введення у REST API // URL: <https://restfulapi.net>
15. Введення в рекомендаційні системи // URL: <https://www.nvidia.com/en-us/glossary/recommendation-system>
16. Катренко Л. А., Катренко А. В. Охорона праці в галузі комп'ютинґу: підручник. Львів: Магнолія – 2006, 2012. 544 с.
17. ДСТУ Б В.1.1-36:2016 Визначення категорій приміщень, будинків, установок за вибухопожежною та пожежною небезпекою. – К.: МІНРЕГІОНБУД України, 2016. 66 с.
18. https://drive.google.com/file/d/0B6R-P_LwCRN-YTJHcGLOY2hsOFU/view?resourcekey=0-sr_df_wNDdEgc6Z4LWJLNQ

					КРБ.КФМН.1.710-03.2.6	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

19. https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=65419
20. ДСанПіН 3.3.2.007-98. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин. – Чинний з 10.12.1998. – К.: Держспоживстандарт України, 1998. 25 с.
<https://zakon.rada.gov.ua/rada/show/v0007282-98#Text>
21. https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=6007
22. НПАОП 40.00-1.32-01. (ДНАОП 0.00-1.32-01) Правила будови електроустановок. Електрообладнання спеціальних установок. Київ, 2001. 78 с.
23. https://dnaop.com/html/1692/doc-%D0%9D%D0%9F%D0%90%D0%9E%D0%9F_40.1-1.32-01
24. https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=47257
25. Правила улаштування електроустановок. К.: Міненерговугілля України, 2017. 617 с.
26. <https://mev.gov.ua/storinka/pravyla-ulashtuvannya-elektroustanovok>
27. https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=69769
28. ДБН В.1.1.7-2016. Пожежна безпека об'єктів будівництва. Загальні вимоги. Київ, 2017. 47 с.
29. https://e-onstruction.gov.ua/laws_detail/3080743763845318619?doc_type=2
30. <https://dbn.co.ua/load/normativy/dbn/1-1-0-88>
31. ДБН В.2.5-28:2018. Природне і штучне освітлення. Київ, 2018. 137 с.
32. https://dbn.co.ua/load/normativy/dbn/dbn_v_2_5_28/1-1-0-1188
33. https://e-construction.gov.ua/laws_detail/3074958732556240833?doc_type=2
34. https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=79885

					КРБ.КФМН.1.710-03.2.6	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК А Презентація

Створення інтелектуальної системи рекомендацій для відвідувачів інтернет-магазинів на прикладі магазину з продажу художньої літератури

Кваліфікаційна робота бакалаврського рівня вищої освіти

СПЕЦІАЛЬНІСТЬ: 123 «КОМП'ЮТЕРНІ НАУКИ»

Здобувач: Чередніченко Віталій · Керівник: Котлик С. В.
Одеський національний технологічний університет · Одеса – 2026

Основні проблеми



Великий обсяг літератури



Складність пошуку потрібних книг



Необхідність персоналізації контенту



Використання інтелектуальних рекомендацій

Актуальність теми

Сучасні інтернет-магазини містять велику кількість товарів, через що користувачам стає складніше знаходити продукцію, яка найбільше відповідає їхнім інтересам та потребам. Особливо актуальною дана проблема є для книжкових магазинів, де асортимент може складатися з тисяч видань різних жанрів та авторів. Саме тому впровадження рекомендаційних систем є важливим напрямком розвитку сучасної електронної комерції.

Задачі та цілі проекту

Основні задачі

• Аналіз рекомендаційних систем

• Вибір технологій розробки

• Проєктування бази даних

• Реалізація веб-додатку

• Реалізація рекомендаційного модуля

• Тестування системи

Мета, об'єкт, предмет та задачі роботи

Мета роботи

Розробка книжкового інтернет-магазину з інтелектуальною рекомендаційною системою на основі контентної фільтрації.

Об'єкт дослідження

Процес взаємодії користувача з книжковим інтернет-магазином.

Предмет дослідження

Методи та технології побудови рекомендаційних систем для електронної комерції.

					КРБ.КФМН.0.710-03.2.6	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

Аналіз рекомендаційних систем

У роботі було проведено аналіз основних підходів до побудови рекомендаційних систем.

Контентна фільтрація

Формує рекомендації на основі характеристик товарів та попередніх дій користувача.

Колаборативна фільтрація

Використовує інформацію про поведінку інших користувачів зі схожими інтересами.

Гібридні системи

Поєднують декілька методів для підвищення якості рекомендацій.

→ Після аналізу було обрано метод контентної фільтрації.

Обґрунтування вибору контентної фільтрації

Для реалізації рекомендаційного модуля було обрано метод контентної фільтрації.

Переваги обраного підходу

- Незалежність від великої кількості користувачів
- Відсутність критичної проблеми холодного старту
- Простота реалізації
- Можливість пояснення причин рекомендацій
- Відповідність структурі книжкового каталогу

Даний підхід добре працює для систем, де товари мають чітко визначені характеристики, такі як жанр, автор та рейтинг.



Технології реалізації проєкту

Для реалізації програмного комплексу було використано сучасний стек веб-технологій.

Клієнтська частина

- HTML
- CSS
- JavaScript

Серверна частина

- Node.js
- Express.js

Збереження даних

- PostgreSQL

Середовище розробки

- Visual Studio Code

Архітектура системи

Програмний продукт реалізовано відповідно до клієнт-сервєрної архітектури. Користувач працює з веб-інтерфейсом, який взаємодіє із сервером через REST API.



Сервєрна частина виконує: обробку HTTP-запитів, бізнес-логіку, авторизацію користувачів та взаємодіє з базою даних. База даних PostgreSQL забезпечує централізоване зберігання інформації про книги, користувачів та замовлення.

Функціональні можливості системи

Система підтримує роботу двох категорій користувачів.

Користувач

- Перегляд каталогу
- Пошук книг
- Робота з кошиком
- Особистий профіль
- Історія покупок
- Персональні рекомендації

Адміністратор

- Додавання книг
- Видалення книг
- Керування користувачами
- Адміністрування системи

Проектування бази даних

Для зберігання інформації використано реляційну базу даних PostgreSQL.

Основні таблиці

Структура БД забезпечує підтримку каталогу книг, кошика, замовлень та персоналізації системи.



Реалізація рекомендаційної системи

Рекомендаційний модуль побудований на основі аналізу взаємодії користувачів з книжками. Визначається такі дії:



Кожна дія має власну вагу та впливає на формування профілю інтересів користувача. На основі отриманих даних система визначає жанри, що цікавлять користувача, та формує список рекомендованої літератури.

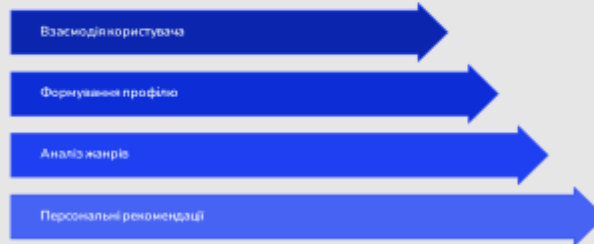


Схема алгоритму послідовності формування рекомендацій на основі обраних даних про взаємодію користувача з каталогом.

Реалізація серверної частини

Основні компоненти

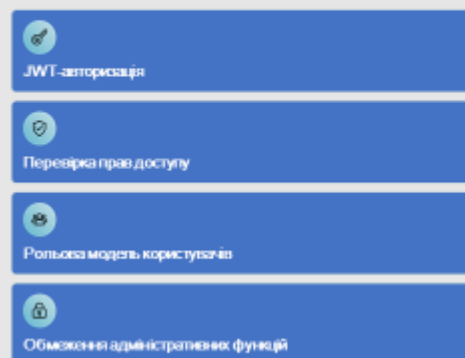


Функціональні можливості

- Авторизація та реєстрація
- Робота з каталогом
- Обробка кошика
- Оформлення замовлень
- Підтримка рекомендаційної системи

Авторизація та безпека

Для безпеки доступу використовується JWT (JSON Web Token). Після успішної авторизації користувач отримує токен доступу, який використовується для виконання захищених операцій.



Висновки

У процесі виконання кваліфікаційної роботи було створено повноцінний книжковий інтернет-магазин, та досліджено сучасні підходи до побудови рекомендаційних систем та розроблено книжковий інтернет-магазин із персоналізованими рекомендаціями.

Практична цінність

Створена система дозволить користувачам здійснювати пошук та придбання книг, а також отримувати рекомендації на основі власних інтересів та попередньої взаємодії з каталогом.

Технологічне застосування

Розроблений програмний продукт демонструє практичне застосування технологій веб-розробки та методів контентної фільтрації в галузі електронної комерції.

СПЕЦІАЛЬНІСТЬ 122

ОДЕСА – 2026

					КРБ.КФМН.0.710-03.2.6	Арк.
						74
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК Б Лістинг

app.js

```
const express = require("express");
const cors = require("cors");
const path = require("path");
const app = express();
app.use(cors());
app.use(express.json());
app.get("/", (req, res) => {
    res.send("Backend is working!");
});
const usersRoutes = require("./routes/users");
app.use("/api/users", usersRoutes);
const cartRoutes = require("./routes/cart");
app.use("/api/cart", cartRoutes);
const ordersRoutes = require("./routes/orders");
app.use("/api/orders", ordersRoutes);
const authRoutes = require("./routes/auth");
app.use("/api/auth", authRoutes);
const booksRoutes = require("./routes/books");
app.use("/api/books", booksRoutes);
app.use("/covers", express.static(path.join(__dirname, "../public/covers")));
const recRoutes = require("./routes/recommendations");
app.use("/api/recommendations", recRoutes);
app.use("/api/interactions", require("./routes/interactions"));
module.exports = app;
```

auth.js

```
const express = require("express");
const router = express.Router();
const pool = require("../db");
const bcrypt = require("bcrypt");
const jwt = require("jsonwebtoken");
const SECRET_KEY = "bookstore_2026";
router.post("/register", async (req, res) => {
    try {
        const { name, email, password, role } = req.body;
        const userCheck = await pool.query(
            "SELECT * FROM users WHERE email = $1",
            [email]
        );
        if (userCheck.rows.length > 0) {
```

					КРБ.КФМН.2.710-03.2.6	Арк.
						75
ЗМН.	Арк.	№ докум.	Підпис	Дата		

```

        return res.status(400).json({ error: "Користувач вже існує" });
    }
    const hashedPassword = await bcrypt.hash(password, 10);
    const result = await pool.query(
        `INSERT INTO users (name, email, password, role)
        VALUES ($1, $2, $3, $4)
        RETURNING id, name, email, role`,
        [name, email, hashedPassword, role || "user"]
    );
    res.json(result.rows[0]);
} catch (err) {
    res.status(500).json({ error: err.message });
}
});
module.exports = router;

```

books.js

```

const { authenticateToken, isAdmin } = require("../middleware/authMiddleware");
const express = require("express");
const router = express.Router();
const pool = require("../db");
router.get("/similar/:id", async (req, res) => {
    try {
        const { id } = req.params;
        const genres = await pool.query(
            `SELECT g.id
            FROM book_genres bg
            JOIN genres g ON g.id = bg.genre_id
            WHERE bg.book_id = $1
            `, [id]);
        const genreIds = genres.rows.map(g => g.id);
        const similar = await pool.query(
            `SELECT b.*, COUNT(*) as score
            FROM books b
            JOIN book_genres bg ON bg.book_id = b.id
            WHERE bg.genre_id = ANY($1) AND b.id != $2
            GROUP BY b.id
            ORDER BY score DESC
            LIMIT 5
            `, [genreIds, id]);
        res.json(similar.rows);
    } catch (err) {

```

					КРБ.КФМН.2.710-03.2.6	Арк.
						76
ЗМН.	Арк.	№ докум.	Підпис	Дата		

```

        res.status(500).json({ error: err.message });
    }
});
module.exports = router;

```

recommendationService.js

```

const pool = require("../db");
const { getUserProfile } = require("../profileService");
async function getRecommendations(user_id) {
    const purchased = await pool.query(`
        SELECT DISTINCT book_id
        FROM user_book_interactions
        WHERE user_id = $1 AND action = 'buy'
    `, [user_id]);
    const purchasedIds = purchased.rows.map(r => r.book_id);
    const profile = await getUserProfile(user_id);
    const books = await pool.query(`
        SELECT
            b.*,
            ARRAY_REMOVE(ARRAY_AGG(g.name), NULL) AS genres
        FROM books b
        LEFT JOIN book_genres bg ON bg.book_id = b.id
        LEFT JOIN genres g ON g.id = bg.genre_id
        GROUP BY b.id
    `);
    if (profile.length === 0) {
        return books.rows
            .sort((a, b) => b.rating - a.rating)
            .slice(0, 10);
    }
    const genreMap = {};
    profile.forEach(p => {
        genreMap[p.genre] = Number(p.weight);
    });
    const scored = books.rows
        .filter(book => !purchasedIds.includes(book.id))
        .map(book => {
            let score = 0;
            if (book.genres && Array.isArray(book.genres)) {
                book.genres.forEach(g => {
                    if (g) {
                        score += genreMap[g] || 0;
                    }
                });
            }
        });
    }

```

					КРБ.КФМН.2.710-03.2.6	Арк.
						77
ЗМН.	Арк.	№ докум.	Підпис	Дата		

```

    }
  });
}
return {
  ...book,
  score
};
});
return scored
  .filter(b => b.score > 0)
  .sort((a, b) => b.score - a.score)
  .slice(0, 10);
}
module.exports = { getRecommendations };

```

book.js

```

const recContainer = document.getElementById("recommendations_container");
function loadRecommendations() {
  const token = localStorage.getItem("token");
  if (!token) return;
  fetch("http://localhost:3000/api/recommendations", {
    headers: {
      "Authorization": "Bearer " + token
    }
  })
  .then(res => res.json())
  .then(data => {
    recContainer.innerHTML = "";
    data.forEach(book => {
      const div = document.createElement("div");
      const genresHTML = book.genres && book.genres.length
        ? book.genres.map(g => `<div class="book_genre">${g}</div>`).join("")
        : "";
      div.innerHTML = `
        <div class="recommendation_book_block">
          
          <h4>${book.title}</h4>
          <p>score: ${book.score.toFixed(2)}</p>
          <div class="genres">
            ${genresHTML}
          </div>

```

					КРБ.КФМН.2.710-03.2.6	Арк.
						78
ЗМН.	Арк.	№ докум.	Підпис	Дата		

```
        </div>
    };
    recContainer.appendChild(div);
    });
    });
}
loadRecommendations();
```

					КРБ.КФМН.2.710-03.2.6	Арк.
						79
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК В Публікації

Матеріали конференції «Стан, досягнення та перспективи інформаційних систем і технологій»

УДК 004.8:339.37

СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ РЕКОМЕНДАЦІЙ ДЛЯ ВІДВІДУВАЧІВ ІНТЕРНЕТ-МАГАЗИНІВ НА ПРИКЛАДІ МАГАЗИНУ З ПРОДАЖУ ХУДОЖНЬОЇ ЛІТЕРАТУРИ

КОТЛИК С.В., СОКОЛОВА О.П., ЧЕРЕДНІЧЕНКО В.С.
(sergknet@gmail.com)

Одеський національний технологічний університет

Анотація. У роботі розглядається задача створення інтелектуальної системи рекомендацій для відвідувачів інтернет-магазину художньої літератури. На основі матеріалів дослідження проаналізовано сучасні підходи до побудови рекомендаційних систем, зокрема методи колаборативної та контентної фільтрації, а також їх гібридні поєднання.

Запропоновано архітектуру рекомендаційної системи, яка базується на аналізі поведінкових характеристик користувачів та їх взаємодії з веб-ресурсом. Особливістю підходу є використання історії переглядів, пошукових запитів і покупок для формування персоналізованих рекомендацій у режимі, наближеному до реального часу. Результати дослідження підтверджують ефективність застосування інтелектуальних рекомендаційних систем у сфері електронної комерції, що дозволяє підвищити релевантність пропозицій, покращити користувацький досвід та збільшити ефективність роботи інтернет-магазину.

Вступ. Інтернет-магазин сьогодні — це вже не просто каталог товарів. Це середовище, яке повинно “розуміти” користувача. Бо якщо не розуміє — користувач іде туди, де розуміють. Особливо це актуально для ринку художньої літератури. Книгу складніше “продати”, ніж товар повсякденного використання. Людина приходить не просто за продуктом — вона шукає емоцію, історію, настрій. І якщо система не допомагає знайти “ту саму книгу”, вона фактично втрачає клієнта.

Тут і з’являється ключова ідея рекомендаційних систем: не чекати, поки користувач знайде, а запропонувати раніше. Сучасні технології дозволяють аналізувати поведінку користувача, передбачати його інтереси та формувати персоналізовані рекомендації. Але проблема в тому, що універсальних рішень не існує. Те, що працює для електроніки, може не працювати для літератури.

У цій роботі розглядається підхід до створення інтелектуальної рекомендаційної системи, адаптованої саме для інтернет-магазину художніх книг, де важливу роль відіграють не тільки дані, а й контекст.

Постановка задачі. Основною метою дослідження є розробка інтелектуальної системи рекомендацій, яка забезпечує персоналізований підбір книг для користувачів інтернет-магазину.

Для досягнення цієї мети необхідно вирішити ряд задач:

- проаналізувати існуючі підходи до побудови рекомендаційних систем;
- визначити найбільш ефективний метод або їх комбінацію для задачі підбору художньої літератури;
- розробити архітектуру системи, яка забезпечує обробку поведінкових даних користувачів;
- реалізувати механізм формування рекомендацій у режимі, наближеному до реального часу;
- оцінити ефективність запропонованого рішення.

Особливістю поставленої задачі є необхідність врахування не лише формальних параметрів (жанр, автор, рейтинг), але й неявних характеристик — інтересів, стилю читання, навіть “смаку” користувача.

171

					КРБ.КФМН.2.710-03.2.6	Арк.
						80
Змн.	Арк.	№ докум.	Підпис	Дата		

І якщо говорити чесно — це вже не просто задача програмування. Це задача розуміння людини через дані.

Методи і матеріали досліджень. Методичною основою дослідження є використання гібридного підходу до побудови рекомендаційних систем.

Колаборативна фільтрація дозволяє знаходити подібних користувачів і формувати рекомендації на основі їхньої поведінки. Якщо двоє читають однакові книги — є велика ймовірність, що їм сподобається ще щось однакове.

Контентна фільтрація, у свою чергу, аналізує характеристики самих книг: жанр, автора, ключові слова, опис. Вона дозволяє рекомендувати схожі об'єкти незалежно від інших користувачів.

У роботі ці підходи поєднуються. І це ключовий момент: кожен окремо - хороший, разом - значно кращі.

Для реалізації системи використано клієнт-серверну архітектуру. Вона включає:

- клієнтську частину (HTML, CSS, JavaScript), яка відповідає за взаємодію з користувачем;

- серверну частину (PHP), де реалізована логіка рекомендацій;

- базу даних (MySQL), яка зберігає інформацію про користувачів і товари.

Основними джерелами даних є:

- історія переглядів;

- пошукові запити;

- покупки користувачів;

- взаємодія з елементами інтерфейсу.

Обробка цих даних дозволяє сформувати поведінковий профіль користувача, на основі якого генеруються рекомендації.

Результати досліджень. У результаті проведеного дослідження було розроблено інтелектуальну рекомендаційну систему, інтегровану в структуру інтернет-магазину.

Система здатна формувати персоналізовані рекомендації з урахуванням індивідуальної поведінки користувача. Було встановлено, що використання гібридного підходу дозволяє значно підвищити точність рекомендацій у порівнянні з використанням окремих методів. Особливо ефективним виявилось використання поведінкових даних. Наприклад, навіть коротка історія переглядів дозволяє зробити більш точні припущення щодо інтересів користувача, ніж статичні параметри профілю.

Розроблена система працює у режимі, близькому до реального часу, що дозволяє адаптувати рекомендації залежно від дій користувача під час сесії. Іншими словами — система “вчиться” прямо під час роботи.

Практичне тестування показало, що використання рекомендацій:

- скорочує час пошуку товару;

- підвищує ймовірність покупки;

- покращує загальний користувацький досвід.

Висновки. У ході дослідження було підтверджено ефективність використання інтелектуальних рекомендаційних систем у сфері інтернет-торгівлі, зокрема для продажу художньої літератури. Обґрунтовано доцільність застосування гібридного підходу, який поєднує колаборативну та контентну фільтрацію. Такий підхід дозволяє компенсувати недоліки кожного з методів і підвищити якість рекомендацій.

Розроблена система демонструє здатність адаптуватися до поведінки користувача та формувати релевантні пропозиції у динамічному середовищі. Практична значущість роботи полягає у можливості її використання в реальних комерційних проєктах. Запропонований підхід може бути масштабований і адаптований до інших сфер електронної комерції.