

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ВИПУСКНА РОБОТА

на тему:

**«Інформаційна система для книжкового магазину
з використанням сучасних веб-технологій»**

Завідувач

випускаючої кафедри

Довбиш А.С.

Керівник роботи

Проценко О.Б.

Студентка гр. ІН–73-9

Рубцова А.А.

СУМИ 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютерних наук

Затверджую _____

Зав. кафедрою Довбиш А.С.

“ _____ ” _____ 2021 р.

Завдання

до випускної роботи

Студентки четвертого курсу, групи ІН-73-9 спеціальності “Комп’ютерні науки” денної форми навчання Рубцової Анастасії Андріївни.

Тема: “Інформаційна система для книжкового магазину з використанням сучасних веб-технологій”

Затверджена наказом по СумДУ

№ _____ від _____ 2021 р.

Зміст пояснювальної записки: 1) аналітичний огляд методів побудови інтернет-магазину; 2) постановка завдання й формування завдань дослідження; 3) огляд і опис засобів для розробки; 4) розробка інтернет-магазину книг «Домашня бібліотека»; 5) аналіз результатів.

Дата видачі завдання “ _____ ” _____ 2021 р.

Керівник випускної роботи _____ Проценко О.Б.

Завдання прийняв до виконання _____ Рубцова А.А.

РЕФЕРАТ

Записка: 70 стор., 33 рис., 4 додатка, 7 джерел.

Об'єкт дослідження— Інформаційна система для книжкового магазину з використанням сучасних веб-технологій

Мета роботи — розробити інтернет-магазин з використанням сучасних веб-технологій, а також шаблону проектування MVC. За допомогою системи управління базами даних MySQL спроектувати базу даних проекту для зручного збереження інформації. За рахунок використання технології AJAX зробити інтернет-магазин більш динамічним та швидким. Засобами PHP реалізувати адміністративну панель для можливості інформаційного наповнення інтернет-ресурсу. За допомогою використання фреймворку Bootstrap забезпечити адаптивність сайту.

Результати — виконано вибір методів вирішення поставленої задачі; застосовано шаблон проектування MVC, на основі якого реалізовано книжковий інтернет-магазин з використанням сучасних веб-технологій.

ІНТЕРНЕТ-МАГАЗИН, КНИЖКИ, ФРЕЙМВОРК, СУБД, PHP, MVC,
MYSQL, HTML, CSS, JAVASCRIPT, AJAX

Зміст

ВСТУП	5
1 ІНФОРМАЦІЙНИЙ ОГЛЯД.....	7
1.1 Дослідження предметної галузі.....	7
1.2 Актуальність виконання проекту	8
1.3 Аналіз аналогів	10
1.4 Постановка задачі.....	16
2 ВИБІР МЕТОДІВ ВИРІШЕННЯ ЗАДАЧІ.....	17
2.1 Вибір методів розроблення	17
2.2 Вибір засобів програмування.....	19
3 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	23
3.1 Проектування структури інтернет-ресурсу.....	23
3.2 Програмна реалізація.....	36
ВИСНОВКИ	56
СПИСОК ЛІТЕРАТУРИ.....	57
ДОДАТКИ.....	59
Додаток А.....	59
Додаток Б	60
Додаток В	67
Додаток Г	69

ВСТУП

На сьогодні ведення бізнесу в Інтернеті є дуже популярною справою, що швидко розвивається та має величезний потенціал. Великі та малі підприємства бажають слідувати сучасним тенденціям та використовувати новітні розробки, саме тому з'являється необхідність тісної співпраці зі сферою інформаційних технологій.

Сайт є найсучаснішим і найефективнішим каналом комунікації з потенційним клієнтом. Тільки можливості інтернет-ресурсу дозволяють повною мірою використовувати всі способи донесення інформаційного повідомлення до користувача.

На сайті можна розмістити значно більше інформації ніж, наприклад, в рекламному блоці чи звичайній газеті, саме тому з'являється необхідність в систематизації та управлінні великими потоками даних. Для кращого сприйняття інформація може бути представлена у різному вигляді: текст, аудіо, відео.

Доступність є головним критерієм для розвитку бізнесу в сучасних умовах. Оскільки мережа Інтернет у наш час набула широкої популярності, вона є єдиним джерелом, де зосереджений більший відсоток різноманітної інформації, який доступний кожному.

Однією з важливих складових успіху підприємства є наявність веб-сторінки, що презентує бізнес у всесвітній мережі Інтернет. У наш час існує велика різноманітність сайтів. В залежності від послуг або продукції, які надає підприємство, виділяють наступні типи інтернет-ресурсів: сайт-портал, корпоративний сайт, інтернет-магазин, сайт-каталог, блог, сайт-візитка, – усі вони відрізняються між собою структурою, дизайном, а також інформацією, яка знаходиться на сторінках даного веб-сайту.

Шукаючи відповідь на запитання, більша частина людей звертається за допомогою до мережі Інтернет. Основною метою такого пошуку є

знаходження великої кількості інформації про компанію або кваліфікаційного робітника, який спеціалізується на розв'язанні питань клієнтів.

Для того, щоб зацікавити користувачів та зробити процес сприйняття інформації більш ефективним інтернет-ресурс повинен бути наділений наступними особливостями:

- стислий зміст і лаконічна презентація компанії та її послуг;
- простий інтерфейс;
- адаптивність сайту під різні пристрої;
- висока швидкість завантаження сайту;
- стабільність та надійність роботи ресурсу.

1 ІНФОРМАЦІЙНИЙ ОГЛЯД

Інтернет-ресурси є невід'ємною частиною нашого життя. Завдяки своїй зручності у процесі використання, а також доступності у будь-який момент часу, сайти набули широкої популярності. Під час навчального процесу вони є одним з основних джерел отримання інформації для студентів та викладачів, саме тому робітники вищих навчальних закладів почали активно використовувати їх у своїй трудовій діяльності. Проте веб-сайти застосовуються не лише у навчальному процесі. Вони є актуальними для величезних корпорацій та власників маленького бізнесу. Оскільки в наш час ведення бізнесу в Інтернеті стає більш поширеним, зростає попит на інтернет-ресурси, які б стали надійним підґрунтям для співпраці між клієнтом та компанією, що надає необхідні послуги. Саме для вирішення даних питань були створені інтернет-магазини, що є чудовим рішенням для тих, хто хоче заробляти за допомогою мережі Інтернет.

1.1 Дослідження предметної галузі

Інтернет-магазин книг «Домашня бібліотека» призначений для реалізації книжкової продукції через Інтернет. Наразі подібні веб-ресурси користуються великою популярністю серед людей, які люблять читати, прагнуть до самореалізації або бажають поглибити свої знання з цікавих для них тем. Саме тому першочерговим завданням є створення книжкового інтернет-магазину з можливістю зручного ознайомлення з асортиментом продукції та подальшим швидким оформленням замовлення. Реалізація поставленої задачі автоматизує наявні процеси та забезпечить: структуроване збереження даних, їх опрацювання та пошук. Використання такого програмного продукту надасть можливість співпрацювати з адміністрацією інтернет-магазину у зручний для користувача спосіб.

Для реалізації поставленого завдання необхідно чітко визначити структуру бази даних, де буде зберігатися уся необхідна інформація, щодо

наявних в магазині книг. Важливим є вирішення зовнішнього вигляду інтернет-ресурсу, його структури та системи навігації.

1.2 Актуальність виконання проекту

Розвиток всесвітньої мережі Інтернет зумовлює створення великої кількості веб-сайтів, завдання яких полягає у задоволенні потреб користувачів. Одним з найважливіших завдань є представлення даних таким чином, щоб час на їх сприйняття був мінімальним, а продуктивність та ефективність – максимальною.

Наразі неможливо уявити успішного бізнесу, товари або послуги якого не представлені в Інтернеті. Кожен підприємець прагне до автоматизації усіх його наявних процесів, саме на цьому етапі з'являється необхідність у кваліфікованих робітниках, які спеціалізуються на створенні сучасних та прибуткових інтернет-ресурсів.

Інтернет-магазин є електронним ресурсом, сайтом з певним каталогом за допомогою якого відбувається прямий продаж товарів чи послуг кінцевому користувачу. Основною задачею даного виду інтернет-ресурсу є швидке та ефективно обслуговування клієнтів. Перелік і ціна товарів, їх детальний опис та наявність на складі є основними елементами бажаного для користувача інтернет-ресурсу.

Основною відмінністю інтернет-магазину від звичайного магазину є відсутність потреби наступних фізичних засобів: торгівельного залу, вітрини, продавців, касирів та охоронців. Створення інтернет-магазину дозволяє реалізувати дану інфраструктуру програмно [4].

Реалізація інтернет-магазину «Домашня бібліотека», що пропонує якісні товари та вигідні умови для співпраці між компанією і потенційним клієнтом, надає можливість автоматизувати процес роботи адміністраторів магазину. Цього можна досягти шляхом відмови від застарілих методів ведення бізнесу на користь сучасним та зручним. На сьогодні робота книжного інтернет-магазину потребує впровадження новітніх технологій, що допоможуть

вдосконалити наявні процеси шляхом їх автоматизації. Найкращим варіантом розв'язання такої проблеми є створення інтернет-магазину, застосування якого підвищить рівень продажів, а також створить привабливий вигляд компанії в очах клієнтів. Переваги такого інтернет-ресурсу:

- Економія часу. Зникає необхідність у відвідуванні магазину, пошуку робітників, що можуть надати необхідну споживачеві консультацію стосовно того чи іншого товару. В інтернет-магазині вся інформація зберігається централізовано на сайті, саме тому пошук займає декілька хвилин, а оформлення замовлення може бути виконано в один клік.

- Великий вибір асортименту. Оскільки звичайні магазини є прив'язаними до конкретного приміщення, їх асортимент товарів та послуг є обмеженим. У свою чергу інтернет-магазини надають можливість замовлення товару з іншого регіону або навіть країни, що значно розширює асортимент продукції.

- Доступність. Для того, щоб здійснити покупку необхідно мати лише комп'ютер чи телефон з доступом до Інтернету. Замовлення можна оформити у будь-який зручний для клієнта час, оскільки інтернет-магазин працює цілодобово.

- Низькі ціни. Інтернет-магазин звільняє його власника від таких затрат, як оплата праці продавцям та консультантам, оренда приміщення. Результатом цього є порівняно низькі ціни на товар. Оскільки клієнт не обмежений одним інтернет-магазином, він має можливість порівняти ціни на декількох веб-сайтах та оформити замовлення в найбільш бажаному для нього інтернет-магазині.

- Відгуки клієнтів. Головною метою кожного клієнта є вибір та подальше придбання якісного товару. Перед тим, як оформити замовлення користувач бажає ознайомитися з детальним описом товару, переглянути наявні фото та дізнатися відгуки клієнтів, що вже придбали дану продукцію.

Подібна інформація може вберегти потенційного покупця від шахраїв чи недобросовісної компанії, що має на меті продати неякісний товар [2].

В інтернет-магазині, що є орієнтованим на роботу з великою кількістю клієнтів, обов'язковим є визначення функціональних можливостей, головною метою яких є забезпечення зручної роботи сервісу з потенційним покупцем. Найбільш вагомими є наступні:

- можливість інформування клієнта шляхом його ознайомлення з актуальним переліком книжок, наявних в магазині, їх ціною та детальним описом;
- можливість зареєструватися в особистому кабінеті та переглядати перелік книжок, які були купленими, цікавими для користувача чи знаходяться у «Кошику» для подальшого замовлення бажаного товару;
- ознайомлення з переліком книжок, які були запропоновані користувачу на основі його нещодавніх покупок;
- наявність контактної інформації адміністраторів інтернет-магазину для проведення онлайн-консультацій;
- можливість застосування загального пошуку або пошуку в категоріях книжок.

1.3 Аналіз аналогів

У нашому повсякденному житті роль газет, журналів та телебачення поступово зменшується, оскільки ці джерела інформації стають дедалі менш ефективними. Найкращим та найсучаснішим ресурсом отримання даних з будь-яких питань є мережа Інтернет, що об'єднує велику кількість веб-сайтів, створюючи тим самим головний інформаційний осередок. Необхідність реалізації інтернет-ресурсу є першочерговою для людей, які займаються продажем товарів або надають послуги. Оскільки кожна організація прагне до автоматизації усіх її наявних процесів, з'явилась необхідність у створенні інтернет-магазину книг «Домашня бібліотека» з можливістю зручного замовлення та продажу.

Створення веб-сайту доволі трудомісткий та тривалий процес, який відбувається в декілька етапів, після проходження яких ідея замовника перетворюється в реальний програмний продукт, що справно працює. Основними етапами є такі:

- визначення теми та головної мети створення сайту, проведення необхідних досліджень на обрану тему, пошук потенційних конкурентів;
- складання технічного завдання, у якому чітко встановлені всі задачі замовника;
- проектування сайту – визначення архітектури сайту та логічної структури сторінок інтернет-ресурсу;
- розробка макета дизайну сайту;
- HTML–CSS верстка – вирішення адаптивності та кросбраузерності веб-сторінок;
- реалізація функціоналу, розміщеного на сайті;
- заповнення інтернет-ресурсу необхідною інформацією;
- тестування програмного продукту, виявлення і виправлення знайдених помилок;
- публікація сайту в Інтернеті;
- адміністрування, модифікація та підтримка інтернет-ресурсу.

Етапи розробки проекту повинні виконуватися послідовно, з метою уникнення несподіваних та небажаних наслідків.

Одним з найважливіших завдань на початку створення програмного продукту є вирішення типу сайту та описання головних вимог щодо його функціональної частини. Інтернет-магазин – один з найбільш інформативних інтернет-ресурсів, на якому зазвичай представлено асортимент магазину, детальна інформація щодо кожного товару, а також основні відомості про компанію: назва, рід діяльності, послуги та контактні дані. Інформаційна функція веб-ресурсу полягає в тому, щоб надати користувачу якомога повнішу інформацію про товари [1]. Головними особливостями даного інтернет-

ресурсу є його проста структура, зрозуміла система керування, зручна для користувача система навігації по магазину, а також мінімальна кількість дій, необхідний для здійснення покупки. Інтернет-магазин має забезпечувати клієнта значно більшим обсягом корисної інформації, необхідної для прийняття рішення про покупку. Для максимальної ефективності на веб-сторінках повинні знаходитись наступні розділи:

- інформація про компанію: історія створення, етапи розвитку, основні досягнення;
- відомості про магазин, оплату і доставлення товару, гарантії, повернення, контактні дані;
- каталог товарів інтернет-магазину;
- кошик покупця, особистий кабінет, а також сторінка оформлення замовлення.

Інтернет-магазин стане найкращим варіантом для представників малого та середнього бізнесу, які мають на меті його розширити та зробити більш прибутковим. Для невеликих компаній, які знаходяться на початкових етапах свого розвитку даний інтернет-ресурс є чудовим варіантом знаходження потенційних покупців, а також зручним осередком для їх подальшої співпраці.

У процесі дослідження предметної галузі було вирішено, що найкращим варіантом для реалізації поставленого завдання стане інтернет-магазин, оскільки:

- реалізація веб-вузла можлива у коротші терміни, порівняно, наприклад, з корпоративним сайтом;
- представлення необхідної користувачу інформації у зручному для нього вигляді;
- докладний опис кожного товару чи послуги, що складають правильне та чітке враження у клієнта;
- добре продумана структура сайту, просте меню та зручна навігація;

- правильно оформлені веб-сторінки підтримують імідж компанії, викликаючи довіру у потенційних покупців;
- інтернет-магазин – чудовий старт для нового бізнесу.

Структура інтернет-магазину повинна бути максимально простою, що досягається шляхом створення невеликої кількості розділів та розробкою зручної навігації між веб-сторінками. Для прикладу, обрано відомий електронний магазин книг «Yakaboo» (Рис. 1.1).

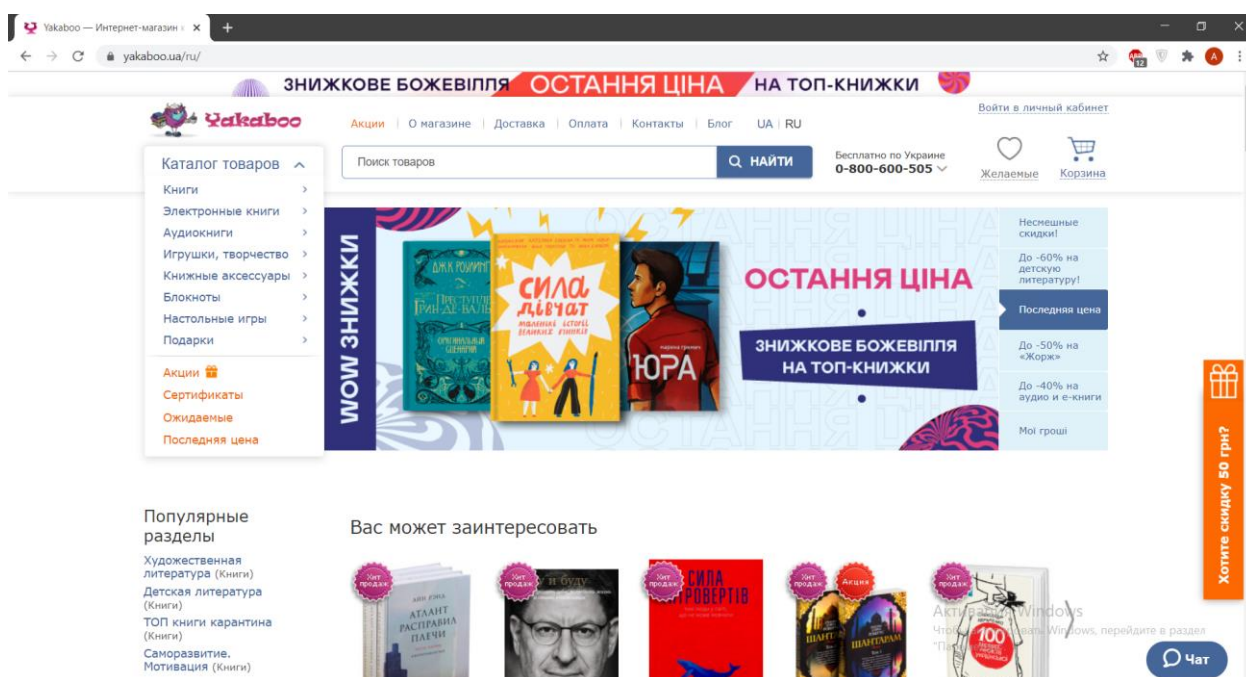


Рисунок 1.1 – Приклад інтернет-магазину книг «Yakaboo»

Основними розділами даного інтернет-ресурсу є:

- сторінка «Про магазин» виконує найважливішу функцію – знайомить користувача з напрямком діяльності, у якому працює компанія. Даний розділ містить основні відомості про інтернет-магазин, його історію створення, досягнення, а також описання продукції, на якій він спеціалізується;
- «Акції» – розділ, де користувач може ознайомитися з переліком акційних товарів, а також переглянути вміст таких категорій, як : «Усі», «Книги», «Електронні книги», «Аудіокниги».

– блог – онлайн журнал, інтернет-щоденник, основним наповненням якого є записи, що постійно додаються та відображаються у хронологічній послідовності. За допомогою такого інтернет-ресурсу можна розповісти про себе, поділитися цікавими ідеями та думками, знайти своїх читачів і підтримувати з ними контакт. Блог є простим у створенні та легким у процесі експлуатації, саме тому даний інтернет-ресурс стає все більш популярним. Для того, щоб користувачі мали змогу отримати більшу кількість інформації з питань, які їх цікавлять, зручним стає створення такого інтернет-ресурсу, де є не тільки можливість замовити товар або послугу, але й ознайомитися з компанією, яка їх надає, прочитати відгуки експертів та інших користувачів, а також поставити власні питання. Саме тому з'являється необхідність в об'єднанні декількох інтернет-ресурсів в один. Звичайною практикою є додавання розділу «Блог», наприклад, до інтернет-магазину, що сприяє більшій ефективності останнього. Головною перевагою даного методу є більший рівень інформативності користувача, завдяки даним, які розміщені у блозі.

– розділ «Оплата» містить перелік варіантів оплати інтернет-замовлення їх детальний опис;

– сторінка «Доставка» допомагає клієнту ознайомитися з можливими способами доставлення інтернет-замовлення. Містить перелік компаній, які надають послуги доставлення товару та з якими співпрацює інтернет-магазин;

– контакти – основний модуль інтернет-магазину, який містить форму зворотного зв'язку, інформацію про місце, де знаходиться компанія, а також можливі посилання на соціальні мережі чи інші інтернет-ресурси.

Для зручного управління інформаційним наповненням інтернет-магазину необхідним є створення адміністративного блоку. За допомогою адміністративної панелі, наявної в інтернет-магазині, адміністратори веб-

ресурсу зможуть додавати нові товари, редагувати та видаляти старі, а також додавати нові сторінки і змінювати існуючі.

Найбільш цікавим елементом для клієнта є каталог товарів електронного магазину, який найчастіше представляє собою розгалужену структуру даних з великою кількістю розділів та підрозділів. Іноді користувач має погане уявлення про те, у якому розділі цікавий для нього товар знаходиться та чи наявний він у магазині взагалі. Саме для вирішення даних питань якісно розроблений інтернет-ресурс повинен забезпечувати зручний пошук, головною задачею якого є зменшення кількості переходів між сторінками каталогу для доступу до необхідної інформації. Для прикладу обрано книжковий інтернет-магазин «book24» (Рис. 1.2), у якому пошук товару можна здійснити за наступними умовами: ім'я чи прізвище автора, назва книги, а також ISBN (міжнародний стандартний номер книги).

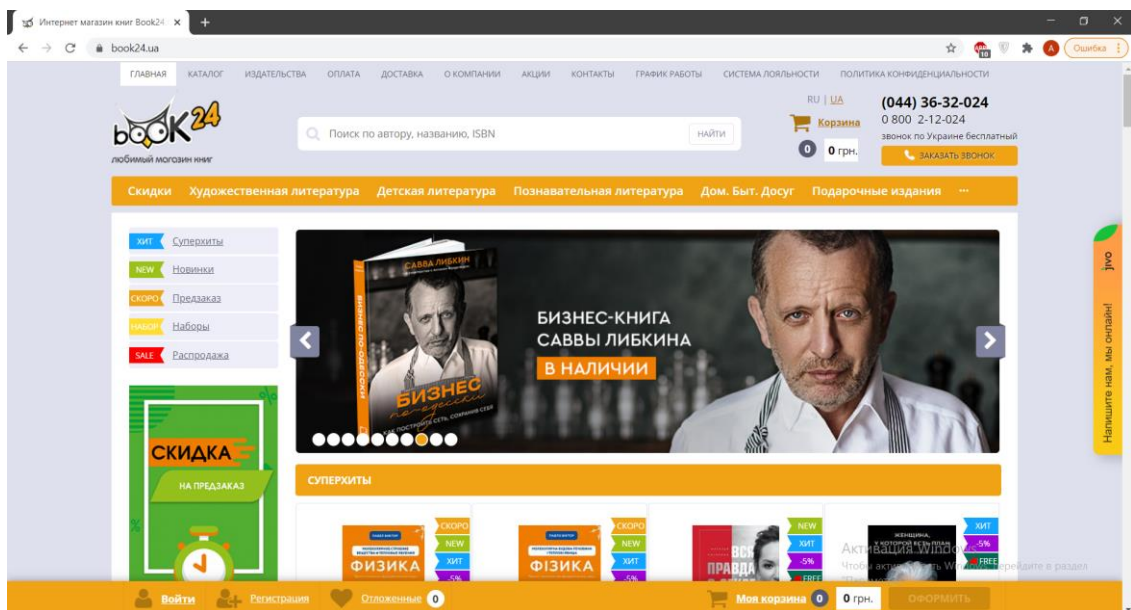


Рисунок 1.2 – Приклад інтернет-магазину книг «book24»

Особливістю реалізації пошуку в Інтернеті є те, що відбувається вибірка усіх записів, які задовольняють умовам пошуку. У випадку великої кількості вибірки даних виведення результатів пошуку відбувається посторінково для того, щоб користувачам не доводилося довго чекати завантаження усієї

вибірки. Результат може включати в себе сотні, тисячі або навіть десятки тисяч записів. Як правило, відвідувачі сайту не переглядають усі доступні сторінки вибірки, обмежуючись двома чи трьома. Саме тому даний механізм пошуку працює повільно та неефективно. Проте він надає можливість здійснити вибірку однакових товарів від різних постачальників, порівняти їх між собою та обрати оптимальний варіант.

1.4 Постановка задачі

У даній роботі необхідно розробити книжковий інтернет-магазин, що матиме наступний функціонал:

- реєстрацію та авторизацію користувачів у системі;
- перегляд наявних у магазині товарів;
- можливість переглянути товар та додати його до кошику;
- пошук необхідної книжки за її номером, назвою чи автором;
- особистий кабінет користувача, з можливістю редагування персональної інформації, а також перегляду нещодавніх покупок;
- можливість зручного оформлення замовлень;
- перегляд записів блогу;
- можливість інформаційного наповнення інтернет-ресурсу за рахунок використання адміністративної панелі, що доступна адміністраторам сайту.

Результатом роботи повинен стати інтернет-магазин, що має бути зрозумілим та простим у користуванні для потенційного клієнта. Усе інформаційне наповнення інтернет-ресурсу має зберігатись у базі даних. Сайт повинен бути безпечним та захищеним від SQL-ін'єкцій. Інтернет-магазин має бути швидким та зручним, що може бути досягнуто завдяки використанню сучасних веб-технологій.

2 ВИБІР МЕТОДІВ ВИРІШЕННЯ ЗАДАЧІ

2.1 Вибір методів розроблення

Сучасне матеріальне виробництво та інші сфери діяльності все більше потребують інформаційного обслуговування, обробки величезної кількості даних. Більша частина процесів: пошуку, зберігання, опрацювання та передавання інформації потребують повної автоматизації. Саме тому експерти з області інформаційних технологій кожного дня працюють над скороченням часу, витраченого на роботу з даними, та створенням максимально зручної для користувача інформаційної системи.

У наш час попит на створення та оптимізацію інтернет-ресурсів стрімко збільшується. В залежності від часу, який відведено на реалізацію програмного продукту, а також коштів, що наявні в бюджеті замовника існує декілька способів створення веб-сайту:

- створення сайту з використанням готової CMS – системи;
- використання фреймворків;
- написання власного коду.

Створення сайту з використанням готової CMS – системи. CMS (від англ. – Content Management System) – це система управління контентом сайту, яка включає програмне забезпечення для роботи з вмістом веб-сторінок (редагування тексту, додавання фотографій, відео та мультимедійних файлів, створення нових розділів, зміна дизайну інтернет-ресурсу). Іноді CMS називають движком сайту, що керує усіма процесами. Зазвичай системи керування контентом складаються з двох частин: back-office, що відповідає за основний функціонал сайту, і front-office – забезпечує взаємодію з користувачем. Необхідність створення CMS обумовлюється збільшенням попиту на використання складніших за структурою сайтів, а також інтернет-ресурсів, дизайн яких не схожий на інші. Веб-сайти, інформація на яких систематично оновлюється, стали цікавішими та зручнішими для користувачів. Їх адміністрування та технічна підтримка зазвичай здійснюється

спеціалістами та фахівцями. Системи управління контентом надають можливість не долучати до роботи кваліфікованих робітників, оскільки усі необхідні програмні коди вже реалізовані та автоматично оновлюються. Це дозволило зосередитися на змісті інформації, її щоденному оновленні.

Переваги, що надають системи управління контентом:

- більша частина CMS безкоштовні та доступні кожному;
- мінімальна кількість часу, витраченого на створення сайту, що обумовлена зрозумілим та зручним інтерфейсом;
- можливість розширення функціоналу, за допомогою модулів та плагінів;
- наявність великої кількості інструментів для роботи;
- безпека та гнучкість використання;
- висока якість розробки програмних кодів, тому ймовірність виявлення помилок є доволі незначною;
- автоматичні оновлення.

Завдяки використанню CMS у процесі розробки інтернет-магазину клієнти матимуть змогу обирати товари, класти їх до кошику, оформлювати замовлення. Зі зворотного боку через інтерфейс адміністратора власники сайту матимуть можливість управляти товарами, їх описами, цінами, бачити замовлення клієнтів і дані, що були внесені при оформленні замовлення [2].

Отже, на сьогоднішній день системи керування контентом дозволяють з легкістю та без особливих умінь управляти сайтом, коректувати його, додавати чи редагувати інформацію, створювати індивідуальні шаблони та використовувати вже наявні.

PHP є найбільш популярною у світі серверною скриптовою мовою програмування, що пройшла довгий шлях розвитку від невеликих, вбудованих в код HTML-сторінок до сучасної мови, на якій розробляється більша частина динамічних сайтів. Складність та функціональність сучасних сайтів невпинно зростає, а тому з'являється необхідність в автоматизації процесу створення

інтернет-ресурсу, оскільки написання усього необхідного коду з нуля вимагає величезної кількості часу. Для того, щоб облегшити програмістам процес розробки сайтів та веб-додатків почали створювати фреймворки [5]. Фреймворк – це набір інструментів, що використовуються для швидкого створення сайтів та веб-додатків шляхом використання вбудованих шаблонів. Існує величезна кількість фреймворків з різноманітними інструментами, реалізовані для різнопланових задач та написані на різних мовах програмування. Основними видами є такі: для серверної частини додатку, наприклад, робота з базами даних та для клієнтської – створення дизайну веб-сторінок та їх оформлення.

Головними перевагами фреймворків є:

- гнучкість та широкі можливості;
- їх використання не накладає ніяких обмежень на розробника, навпаки – надає йому можливість створювати та використовувати необхідний функціонал;
- вища швидкість роботи;
- проекти, реалізовані на основі фреймворків легко розширюються та модернізуються.

Усі фреймворки містять лише базові компоненти, інші додаткові функції створюються у процесі розробки й на власний розсуд користувача. Для того, щоб використовувати фреймворки у процесі розробки сайту необхідно розуміти логіку бізнес-процесів, які будуть реалізовані.

Процес написання власного інтернет-ресурсу вимагає ще більших витрат часу, а також професійних знань та навичок, які необхідні у процесі створення веб-сайтів. Подібний спосіб гарантує реалізацію унікального ресурсу, але потребує багато сил та необхідність залучення до процесу декількох розробників. Створення власного сайту передбачає вивчення основних мов програмування та ознайомлення з новітніми технологіями

розробки інтернет-ресурсів. Можливість закріпити усе вивчене на практиці допомагає людям набути досвіду та стати справжніми професіоналами.

2.2 Вибір засобів програмування

WordPress – сучасна система управління контентом, яка спочатку призначалася для обслуговування блогів. Але сьогодні ця CMS пропонує можливість створювати сайти будь-якого типу і будь-якої складності [6]. WordPress найкраще підійде для створення сайтів з невеликою кількістю контенту, наприклад, сайт-візитка чи лендінг. Дана CMS пропонує велику кількість безкоштовних плагінів та шаблонів, завантаження та налаштування яких є доволі простим процесом. Розробники WordPress надають їй якісну підтримку, що передбачає щомісячне виправлення дрібних помилок та розширення наявного функціонала. Проте дана система управління контентом не передбачена для роботи з масштабними проектами, оскільки, через зростання кількості інформації, яка розміщена в базі даних, знижується швидкість завантаження веб-сторінок.

Laravel – це безкоштовний PHP фреймворк з відкритим вихідним кодом, створений для розробки додатків по архітектурному шаблону MVC. Структура коду Laravel framework відповідає популярному шаблону проектування MVC, тобто в ньому можна виділити моделі (models), представлення (views) та контролери (controllers). Даний шаблон проектування зарекомендував себе як перевірене часом рішення ефективної структури додатку, першочергово, у web, що дозволяє відділити логіку додатку від його візуальної частини. MVC надає можливість зробити код більш читабельним, в процес розробки – зручним, розмежовуючи роботу frontend- та backend-розробників. Завдяки менеджеру пакетів Composer, фреймворк Laravel дозволяє легко встановлювати і підключати різноманітні компоненти для подальшого використання у веб-додатку. Основними перевагами фреймворку Laravel є:

- зрозуміла документація;

- Laravel містить зручний механізм обробки помилок та виключень;
- фреймворк включає в себе вбудовані механізми автентифікації та авторизації користувачів, які легко налаштувати під власні вимоги;
- синтаксис API простий і зрозумілий; не містить складних конструкцій, натомість – добре продумані та чіткі назви функцій [7].

Наразі даний фреймворк підтримує та удосконалює велика команда спеціалістів, що слідує основним ІТ-тенденціям та впроваджує їх в наявний проект.

Існуючими недоліками фреймворку Laravel є такі:

- достатньо мала кількість статей, прикладів коду та офіційної документації на російській чи українській мові;
- порушена зворотна сумісність між версіями фреймворку;
- нелогічне розміщення каталогів і файлів [7].

Оскільки фреймворк Laravel є простим в освоєнні, а також добре зарекомендував себе у реалізації невеликих проектів він став найбільш популярним серед інших фреймворків (Рис. 1.3).

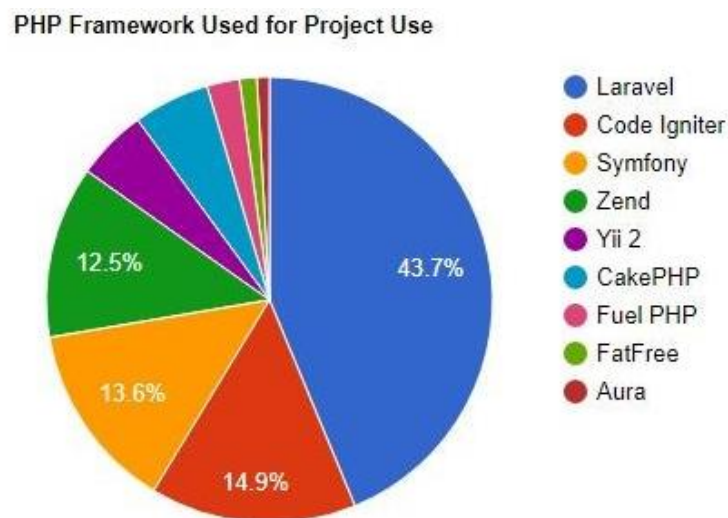


Рисунок 1.3 – Найбільш популярні фреймворки 2018 року [8]

MySQL представляє собою систему управління реляційними базами даних [3]. На сьогодні вона є однією з найпопулярніших систем управління базами даних (Рис. 1.4).

Rank			DBMS	Database Model	Score		
Jan 2020	Dec 2019	Jan 2019			Jan 2020	Dec 2019	Jan 2019
1.	1.	1.	Oracle	Relational, Multi-model	1346.68	+0.29	+77.85
2.	2.	2.	MySQL	Relational, Multi-model	1274.65	-1.01	+120.39
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1098.55	+2.35	+58.29
4.	4.	4.	PostgreSQL	Relational, Multi-model	507.19	+3.82	+41.08
5.	5.	5.	MongoDB	Document, Multi-model	426.97	+5.85	+39.78
6.	6.	6.	IBM Db2	Relational, Multi-model	168.70	-2.65	-11.15
7.	7.	8.	Elasticsearch	Search engine, Multi-model	151.44	+1.19	+8.00
8.	8.	7.	Redis	Key-value, Multi-model	148.75	+2.51	-0.27
9.	9.	9.	Microsoft Access	Relational	128.58	-0.89	-13.04
10.	11.	10.	SQLite	Relational	122.14	+1.78	-4.66
11.	10.	11.	Cassandra	Wide column	120.66	-0.04	-2.32
12.	12.	12.	Splunk	Search engine	88.67	-1.85	+7.25
13.	13.	13.	MariaDB	Relational, Multi-model	87.45	+0.66	+8.63
14.	14.	15.	Hive	Relational	84.24	-1.81	+14.33
15.	15.	14.	Teradata	Relational, Multi-model	78.29	-0.21	+2.10

Рисунок 1.4 – Найбільш популярні системи управління базами даних 2020 року [9]

Основними перевагами MySQL є:

- широкий функціонал, що включає в себе майже всі необхідні інструменти, які є необхідними під час розробки проекту;
- MySQL є однією з найбільш швидких серед існуючих СУБД;
- масштабованість: MySQL може використовуватись у роботі з малими, а також великими об'ємами даних;
- безпека, що досягається за рахунок вбудованих систем, які працюють за замовчуванням;
- MySQL легко встановлюється, має зрозумілий інтерфейс, а різноманітні плагіни і додаткові програми спрощують процес роботи з базою даних [10].

У даній роботі було вирішено створити власний міні-фреймворк засобами PHP з метою поглиблення знань щодо структури та її основних елементів існуючих фреймворків. Необхідна інформація для повноцінної роботи інтернет-ресурсу зберігатиметься у системі управління базами даних MySQL.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Проектування структури інтернет-ресурсу

Інтернет-магазин «Домашня бібліотека» реалізований на основі шаблону MVC (Рис. 3.1), головними складовими якого є модель (model), вид (view), контролер (controller). Основною задачею контролеру є керування запитами користувача, що надходять у вигляді запитів HTTP GET або POST у той час, коли користувач натискає на елементи інтерфейсу для виконання різноманітних дій. Виклик і координування дій необхідних ресурсів та об'єктів, необхідних для виконання – основна функція контролеру. В основному контролер містить організаційну логіку самого додатку. Зазвичай контролер викликає відповідну модель для задачі і обирає необхідний вид. У свою чергу модель надає контролеру необхідні для користувача дані. Оскільки уся інформація, що є важливою для повноцінного функціонування інтернет-магазину, зберігається у базі даних, то процес отримання даних буде описано у відповідних моделях. Отримані дані з моделі будуть передані виду, який забезпечує різні способи представлення даних. Веб-додаток зазвичай складається з набору контролерів, моделей та видів.

Найбільш вагомою перевагою використання шаблону MVC є чіткий поділ логіки представлення (інтерфейсу користувача) та і логіки додатку. Наприклад, однією з найважливіших проблем на сьогодні є підтримка користувачів, які користуються різними типами пристроїв. Відображення вмісту веб-сторінки повинно відрізнятися, якщо запит надходить з персонального комп'ютеру чи з мобільного телефону. У даному випадку модель повертає однакові дані і єдиною відмінністю є те, що контролер обирає різне представлення для виводу даних [11].

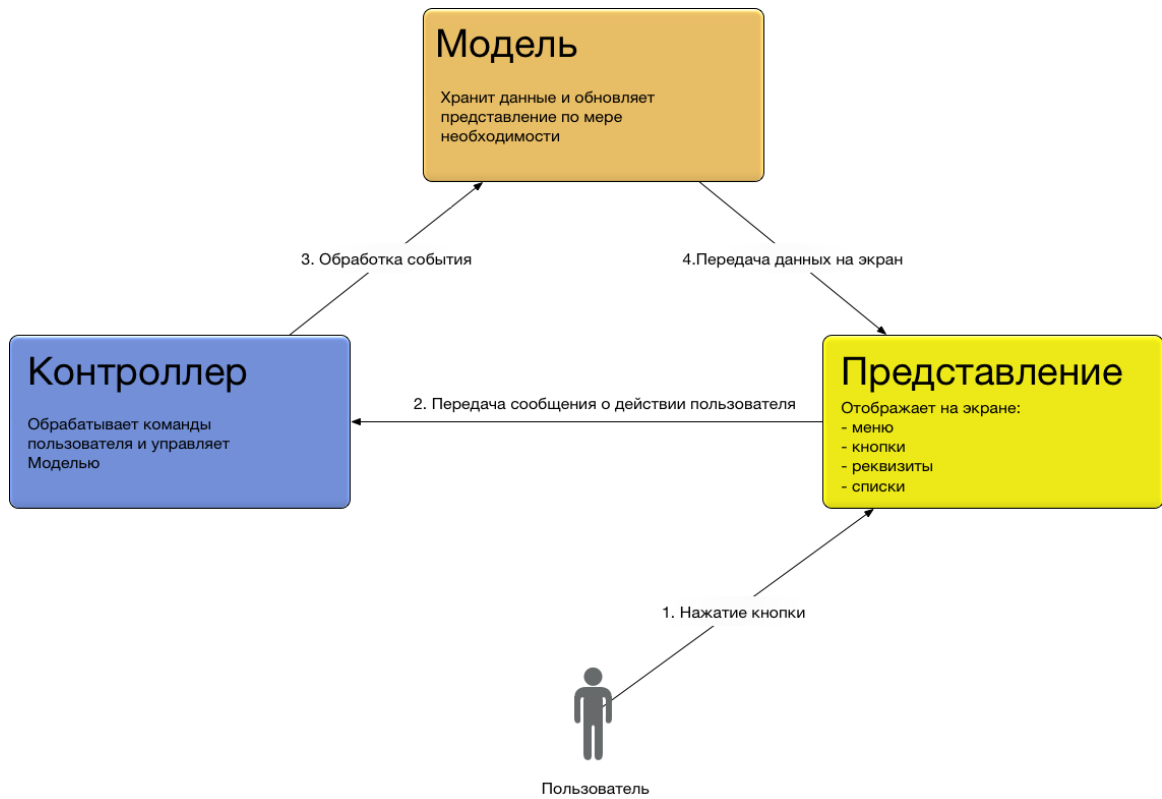


Рисунок 3.1 – Модель шаблону проектування MVC

Структура інтернет-ресурсу була створена самостійно і має наступний вигляд (Рис. 3.2).

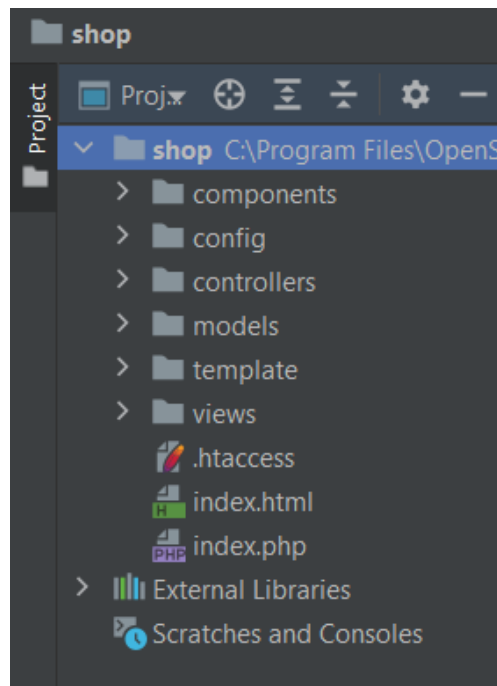


Рисунок 3.2 – Файлова структура інтернет-ресурсу

Інтернет-магазин називається «shop», саме тому його коренева папка має таку саму назву. У папці «components» розміщено клас «Db», у якому є функція getConnection() основною задачею якого є підключення до бази даних. У папці «config» розміщені основні налаштування (звичайні php-файли – масиви з ключами та значеннями), необхідні для коректної роботи всього інтернет-ресурсу. Так, наприклад, у файлі «db_params» описані основні параметри для підключення до бази даних («host», «database», «user», «password»). Підхід при якому основні налаштування виносяться до одного файлу є розповсюдженим, оскільки надалі, при можливих змінах, наприклад, параметрів підключення до бази даних внесення буде легко вносити, що забезпечить від небажаних помилок у процесі розробки додатку. Ще одним файлом у папці «config» є «routes», у якому описана інформація, необхідна для коректного відображення вмісту кожної зі сторінок інтернет-ресурсу. У папці «controllers» розміщені контролери, які використовуються у проекті, наприклад, ProductController чи CatalogController. Класи, в методах яких описані запити до бази даних чи до інших сторонніх сервісів називаються моделями проекту і зберігається у папці «models». Стили, що використовуються на сторінках інтернет-ресурсу описані у файлах папки «template». У додатку кожній сторінці відповідає певне представлення з необхідним в ньому вмістом, дані види розташовані у папці «views».

У процесі проектування структури інтернет-магазину було вирішено, що уся інформація буде зберігатися у базі даних. В якості системи управління базами даних було обрано MySQL. Усі налаштування, необхідні для коректної взаємодії бази даних та інтернет-ресурсу прописані у файлі «db_params.php». Лістинг файлу «db_params.php»:

```
<?php
return array(
    'host' => 'localhost',
    'database' => 'super_mag',
    'user' => 'root',
```

```
'password' => 'root'
);
```

Однією з основних задач у процесі створення інтернет-ресурсу є проектування бази даних. Даний процес є дуже важливим, оскільки з правильно спроектованою базою даних працювати простіше, а процес написання запитів до неї є більш швидким. Проектування бази даних складається з наступних етапів:

- виділення сутностей та їх атрибутів, що будуть зберігатися у базі, формування таблиць на їх основі;
- визначення первинних ключів таблиць;
- визначення зв'язків між таблицями за допомогою зовнішніх ключів;
- нормалізація бази даних [12].

На першому етапі проектування відбувається виділення сутностей. Сутність представляє собою тип об'єкта, що зберігається у базі даних. Кожна сутність – це окрема таблиця. Проаналізувавши процес роботи інтернет-магазину, було виділено наступні сутності:

- товар – невід'ємний елемент інтернет-магазину, що є цікавим для потенційного покупця та має власні характеристики (атрибути). Оскільки, задачею є створення інтернет-магазину, то дані характеристики будуть у повній мірі описувати книги (їх назву, автора, категорію ціну, кількість на складі);
- категорії товару; у магазині кожен товар відноситься до певної категорії, за наявності якої відбувається більш швидкий пошук продукту. Кожна категорія має свій номер, назву та є окремою сутністю у базі даних. Особливістю такого підходу є те, що у колонці «Категорія товару» таблиці «Товар» буде зберігатися не повна назва категорії, а лише її номер. Дана умова є необхідною у процесі нормалізації бази даних;
- автор книги, що також є окремою сутністю. Основними атрибутами даної сутності є унікальний номер автора, що однозначно

ідентифікує його в межах таблиці, а також повне ім'я. Дана сутність була виділена з метою забезпечення нормалізації бази даних за рахунок зберігання у колонці «Автор» таблиці «Товар» не повне ім'я автора, а лише його номер;

- користувач є сутністю, що переглядає наявні товари, додає їх до кошику, а також оформлює замовлення. Атрибутами користувача є його персональні дані: ім'я, телефон, електронна пошта. Створення даної сутності обумовлено тим, що персональні дані користувача мають бути збережені для подальшого зворотного зв'язку під час оформлення замовлення;

- замовлення – окрема сутність у базі даних, атрибутами якої є такі: унікальний номер замовлення, користувач, що його оформив, бажаний товар, кількість, дата створення замовлення. У стовпцях «Користувач» та «Товар» даної таблиці зберігаються лише ідентифікаційні номери, що є необхідною умовою для нормалізації бази даних;

- блог – окрема сутність, що не є зв'язаною з будь-якою іншою, оскільки стосується окремого розділу інтернет-магазину. У даній таблиці зберігається уся інформація щодо записів блогу (їх тема, дата публікації, інформаційне наповнення).

Кожна сутність має набір атрибутів, що описують її у повній мірі. Більш детальна структура кожної таблиці бази даних приведена нижче:

- «category» – містить інформацію про категорії (Рис. 3.3): номер категорії (category_id), її назва (name) порядковий номер при відображенні на сайті (sort_order) режим відображення на сайті (status);

- «author» – містить інформацію про авторів книг (Рис. 3.4): (ідентифікаційний номер автора (author_id), його ім'я (name));

- «product» – містить інформацію про товар (Рис. 3.5): код товару (product_id), назва товару (name), ідентифікаційний номер автора книги (author_id), номер категорії, до якої відноситься книга (category_id), ціна (price), доступність (availability), видавництво (brand), фото книги (image),

описання змісту книги (description), чи є новою (is_new), чи є рекомендованою для читання (is_recommended), режим відображення на сайті (status).

– «user» – містить інформацію про користувачів (Рис. 3.6): ідентифікаційний номер користувача (user_id), ім'я користувача (name), поштова скринька (email), телефон (phone), password (пароль), сіль (salt), роль (role);

– «product_order» – містить інформація про замовлення (Рис. 3.7): унікальний номер замовлення (id), коментар користувача до замовлення (user_comment), ідентифікаційний номер користувача (user_id), дата замовлення (data), статус замовлення (status), код товару (product_id), quantity (кількість замовлених одиниць товару).

– «blog» – містить інформацію, що буде відображатися у розділі «Блог» (Рис. 3.8): ідентифікаційний номер запису (id), тема статті (title), фото (image), дата публікації (date), стислий опис (short_content), record_link (посилання на статтю);

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	category_id	int(11)		No	None		AUTO_INCREMENT	
<input type="checkbox"/>	2	name	varchar(255)	utf8_general_ci	No	None			
<input type="checkbox"/>	3	sort_order	int(11)		No	0			
<input type="checkbox"/>	4	status	int(11)		No	1			

Рисунок 3.3 – Структура таблиці «category»

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	author_id	int(11)		No	None		AUTO_INCREMENT	
<input type="checkbox"/>	2	name	varchar(255)	utf8_general_ci	No	None			

Рисунок 3.4 – Структура таблиці «author»

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	product_id			No	None			
<input type="checkbox"/>	2	name	utf8_general_ci		No	None			
<input type="checkbox"/>	3	author_id			Yes	NULL			
<input type="checkbox"/>	4	category_id			No	None			
<input type="checkbox"/>	5	price			No	None			
<input type="checkbox"/>	6	availability			No	None			
<input type="checkbox"/>	7	brand	utf8_general_ci		No	None			
<input type="checkbox"/>	8	image	utf8_general_ci		No	None			
<input type="checkbox"/>	9	description	utf8_general_ci		No	None			
<input type="checkbox"/>	10	is_new			No	0			
<input type="checkbox"/>	11	is_recommended			No	0			
<input type="checkbox"/>	12	status			No	1			
<input type="checkbox"/>	13	in_stock			Yes	NULL			

Рисунок 3.5 – Структура таблиці «product»

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	user_id			No	None		AUTO_INCREMENT	
<input type="checkbox"/>	2	name	utf8_general_ci		No	None			
<input type="checkbox"/>	3	email	utf8_general_ci		No	None			
<input type="checkbox"/>	4	phone	utf8_general_ci		Yes	NULL			
<input type="checkbox"/>	5	password	utf8_general_ci		No	None			
<input type="checkbox"/>	6	salt	utf8_general_ci		Yes	NULL			
<input type="checkbox"/>	7	role	utf8_general_ci		Yes	NULL			

Рисунок 3.6 – Структура таблиці «user»

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id			No	None		AUTO_INCREMENT	
<input type="checkbox"/>	2	user_comment	utf8_general_ci		No	None			
<input type="checkbox"/>	3	user_id			Yes	NULL			
<input type="checkbox"/>	4	date			No	CURRENT_TIMESTAMP		DEFAULT_GENERATED	
<input type="checkbox"/>	5	status			No	1			
<input type="checkbox"/>	6	product_id			Yes	NULL			
<input type="checkbox"/>	7	quantity			Yes	NULL			

Рисунок 3.7 – Структура таблиці «product_order»

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id			No	None		AUTO_INCREMENT	
<input type="checkbox"/>	2	title	utf8_general_ci		No	None			
<input type="checkbox"/>	3	image	utf8_general_ci		Yes	NULL			
<input type="checkbox"/>	4	date			No	CURRENT_TIMESTAMP		DEFAULT_GENERATED	
<input type="checkbox"/>	5	short_content	utf8_general_ci		No	None			

Рисунок 3.8 – Структура таблиці «blog»

Концептуальна модель бази даних має наступний вигляд (Рис. 3.9):

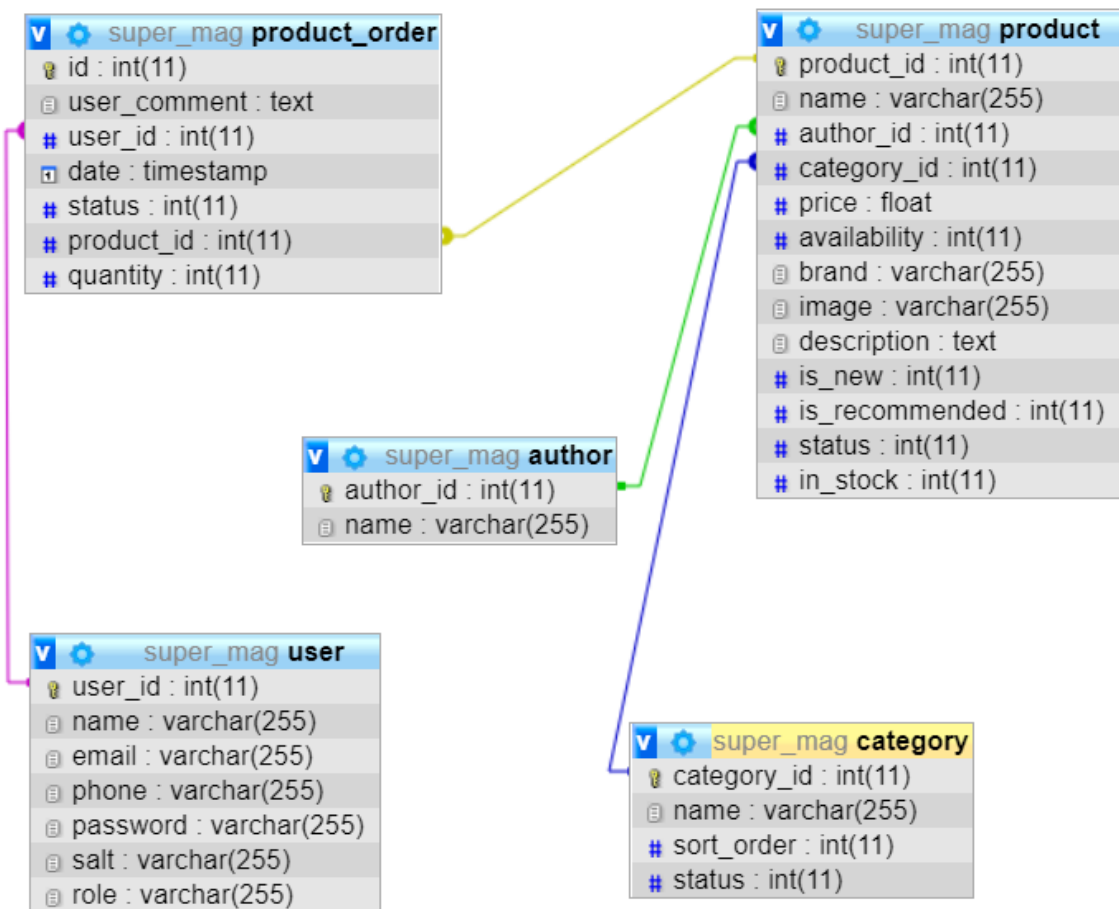


Рисунок 3.9 – Концептуальна модель бази даних «super_mag»

Важливим етапом у процесі проектування бази даних є визначення первинних (primary key) та зовнішніх ключів (foreign key). Первинним називають стовпець або комбінацію стовпців, які однозначно ідентифікують запис у базі даних. Первинний ключ, що складається з декількох полів називається складеним первинним ключем. Стовпець, який надалі буде первинним ключем, задається у процесі створення полів таблиці. При заповненні даними треба враховувати, що значення первинного ключа мають бути унікальними в межах таблиці. Головним завданням первинного ключа є підтримання цілісності бази даних. Зовнішнім ключем є стовпчик чи їх сукупність, які в одній таблиці не позначені первинними ключами, але є первинними ключами в іншій таблиці. Основною задачею зовнішнього ключа

є забезпечення реферальної цілісності бази даних. За допомогою foreign key ми поєднуємо дані з зовнішнього ключа з відповідними даними у первинному ключі іншої таблиці. Дані, що заносяться до зовнішнього ключа повинні відповідати даним з первинного ключа таблиці [13]. При спробі вставити до таблиці, що містить foreign key, дані, які не присутні в primary key, виникне помилка.

Для того, щоб створити foreign key по полю «category_id» в таблиці «product» необхідно виконати наступну команду:

```
ALTER TABLE product
ADD CONSTRAINT FK_product_category_id
FOREIGN KEY (category_id) REFERENCES category(category_id);
```

Набір правил, що накладають деяке обмеження при вставці, видаленні чи оновленні даних називають обмеженнями. У MySQL існують обмеження на рівні стовпця та обмеження рівні таблиці. Одними з найпоширеніших обмежень є такі: primary key, foreign key, unique key, not null, default, check.

Для того, щоб переглянути усі наявні в таблиці обмеження необхідно виконати наступну команду:

```
SELECT
    CONSTRAINT_NAME,
    CONSTRAINT_TYPE
FROM
    information_schema.TABLE_CONSTRAINTS
WHERE
    TABLE_NAME = 'product_order'
```

У результаті можна буде побачити назви обмежень та їх тип (Рис. 3.10).

CONSTRAINT_NAME	CONSTRAINT_TYPE
PRIMARY	PRIMARY KEY
FK_product_order_product_id	FOREIGN KEY
FK_product_order_user_id	FOREIGN KEY

Рисунок 3.10 – Обмеження таблиці «product_order»

Внутрішня структура веб-вузла, як правило, представлена картою інтернет-ресурсу, що візуально відображає ієрархію його сторінок, схему зв'язків і переходів між ними. Виділяють три основні типи: лінійну, деревоподібну та довільну. Для реалізації швидкого переходу між сторінками інтернет-ресурсу була створена проста система навігації, яка ґрунтується на лінійній структурі. Це означає, що користувач може швидко переходити від одного розділу до іншого і в будь-який час повернутися назад.

У системі наявні 3 типи користувачів: звичайний користувач, клієнт магазину та адміністратор сайту (Рис. 3.11). Кожен відвідувач сайту має доступ лише до тих функцій, що доступні йому відповідно його ролі. Так, наприклад, користувач, що не є зареєстрованим на сайті, не має можливості замовляти товари. Авторизованому клієнту доступне оформлення замовлень та редагування персональних даних. Адміністратор сайту має найбільш повний перелік доступних функцій. Найбільш важливими є робота з адміністративною панеллю та інформаційне наповнення сайту.

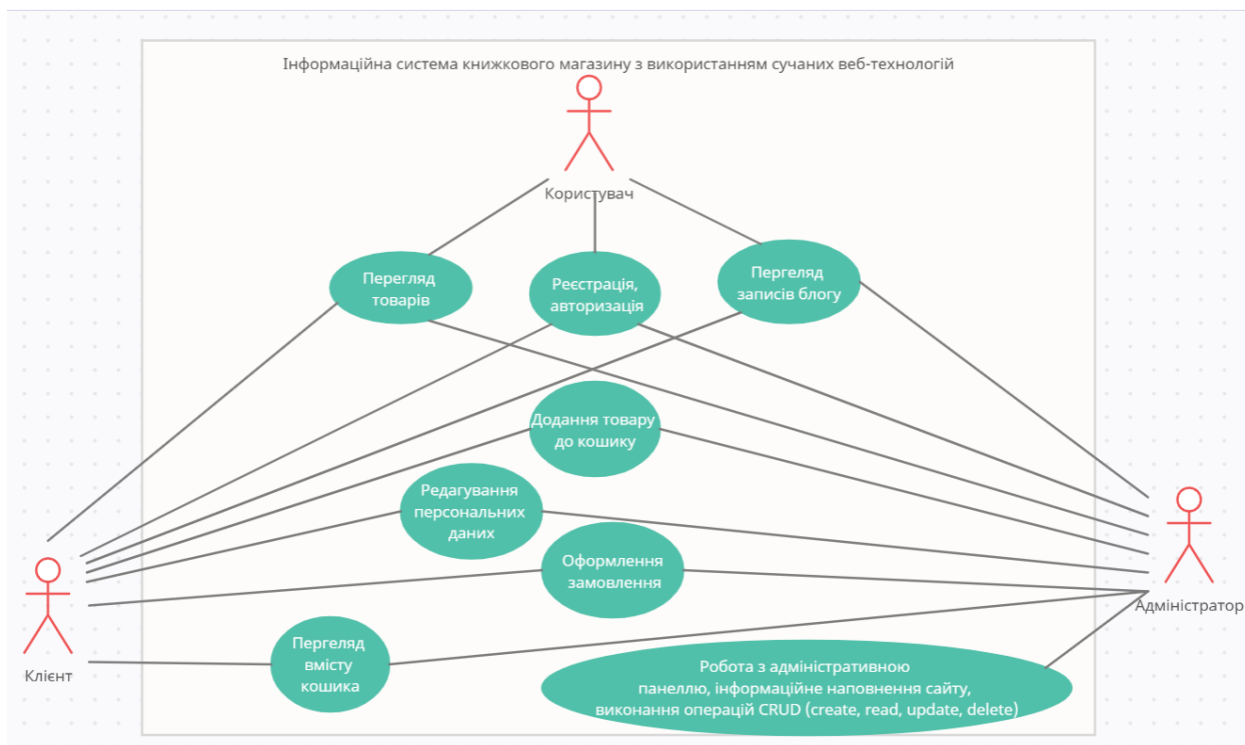


Рисунок 3.11 – UML діаграма

Розроблений додаток містить такі розділи: «Головна», «Магазин», «Блог», «Контакти». Головна сторінка відіграє ключову роль, оскільки відразу дає зрозуміти потенційному клієнту сферу діяльності та асортимент товару, який пропонує власник маленького бізнесу. На головній сторінці наступні елементи:

- блок з категоріями книжок;
- блок з прізвищами авторів та кількістю їх книг, що наявні в бібліотеці;
- блок з товарами та можливістю детального ознайомлення з кожним;
- блок з книгами, що рекомендовані для читання.

На сторінці з детальним описом товару можна прочитати відповідні характеристики (код книжки, назва, ціна, видавництво, короткий опис, а також кількість наявних на складі). Якщо товар є в наявності, то його можна додати до кошика.

Сторінка «Кошик» містить таблицю з товарами, що обрав користувач. На даній сторінці є можливість переглянути обраний товар або видалити його зі списку.

Сторінки «Реєстрація» та «Авторизація» передбачають заповнення усіх полів форми. Для того, щоб користувач без проблем зареєструвався необхідно ввести своє ім'я, телефон, електронну пошту та мобільний телефон. Для процесу авторизації достатньо ввести електронну пошту та пароль. Безпечність особистих даних в Інтернеті – це те, про що нам необхідно піклуватись. На сьогодні є велика кількість проблем, пов'язаних з викраденням персональних даних користувачів (їх паролів та іншої інформації). Існує декілька підходів до збереження паролів у базах даних:

- збереження паролів в такому вигляді, в якому вони були введені користувачем. Даний підхід є небезпечним, оскільки зловмисник отримує паролі у явному вигляді.

– збереження паролів у зашифрованому вигляді, використовуючи такі алгоритми шифрування, як md5, sha1 та інші. У разі злomu зловмисник отримує паролі, але не має можливості їх розшифрувати, оскільки має місце незворотне шифрування. Проте методом перебору він може шифрувати різні комбінації символів і порівнювати їх хеш з викраденим хешем пароля. Оскільки паролі більшої кількості користувачів складаються з цифр, простих слів та легких для злomu комбінацій символів, то даний підхід також не є безпечним. Наразі злом деяких паролів займає декілька секунд, оскільки існують сервіси, на яких збережено великі об'єми хешу паролів [14].

– Збереження паролів у зашифрованому вигляді, додаючи до нього декілька випадково обраних символів, що є унікальними для кожного користувача. Даний набір символів називають сіллю, що зберігається у базі даних. Пароль, що є зашифрованим таким чином є безпечним. Також пароль можна зберігати наступним чином – md5(md5(пароль) + сіль), що є ще більш безпечним при роботі з паролями користувачів.

Приклад функції, що генерує сіль:

```
function generateSalt() {
    $salt = '';
    $length = rand(5,10); // длина соли (от 5 до 10 символов)
    for($i=0; $i<$length; $i++) {
        $salt .= chr(rand(33,126)); // символ из ASCII-table
    }
    return $salt;
}
```

У процесі розробки широкого використовувались підготовлені запити. Концепція підготовлених запитів підтримується більшою кількістю сучасних баз даних. Підготовлені запити можна описано як деякий вид скомпільованого шаблону SQL-запиту, який буде запускатися програмним додатком та налаштовуватися за допомогою вхідних параметрів. Використання підготовлених запитів характеризується наступними перевагами:

– запит готується один раз, надалі його можна використовувати стільки, скільки необхідно. Слід додати, що параметри можуть бути такими ж

або відрізняться. Якщо запит є підготовленим, СУБД аналізує його, компілює та оптимізує план виконання запиту. Якщо запит є складним, то процес його виконання може займати багато часу, що призведе до більш повільної роботи додатку. Значення ефективності значно погіршиться у разі виконання запиту з різними параметрами. Під час використання підготовленого запиту СУБД аналізує, компілює та оптимізує запит будь-якого рівня складності лише один раз, а програмний додаток завантажує на виконання вже підготовлений шаблон. Використання значно меншої кількості ресурсів та висока швидкість виконання є основними особливостями підготовлених запитів [15].

– використання виключно підготовлених запитів у програмних додатках захищає від SQL-ін'єкцій.

Приклад використання підготовлених запитів у функції, що відповідає за зміну ім'я, номеру телефона та пароля користувача в інтернет-магазині:

```
public static function edit($id, $name, $phone, $password){
    $dbh = Db::getConnection();

    $sql1 = 'SELECT salt FROM user WHERE user_id = :id';
    $result1 = $dbh->prepare($sql1);
    $result1->bindParam(':id', $id, PDO::PARAM_INT);
    $result1->execute();
    $salt = $result1->fetchColumn();
    $sql = 'UPDATE user SET name = :name, phone = :phone,
password = :temp where user_id = :id';
    $temp = md5($password.$salt);
    $result = $dbh->prepare($sql);
    $result->bindParam(':id', $id, PDO::PARAM_INT);
    $result->bindParam(':name', $name, PDO::PARAM_STR);
    $result->bindParam(':phone', $phone, PDO::PARAM_STR);
    $result->bindParam(':temp', $temp, PDO::PARAM_STR);
    $result->execute();
}
```

Розділ «Оформлення замовлення» доступний тільки авторизованим користувачам. Даний розділ надає можливість створити замовлення. Аби адміністратор сайту міг з легкістю переглянути інформацію щодо користувацьких замовлень, вони зберігаються в базі даних.

Розділ «Блог» дозволяє відвідувачам знайомитися з цікавими та корисними статтями, читати повні матеріали відповідно до обраної теми.

3.2 Програмна реалізація

Першим кроком у процесі створення інтернет-магазину стало визначення його зовнішнього вигляду. Вибір зручного шаблону з простим для користувача інтерфейсом було основним завданням початкового етапу розробки сайту. Першочерговою вимогою до шаблону була наявність достатньої кількості блоків, до яких можна було б помістити необхідну інформацію. Для стилізації більшої частини елементів інтернет-магазину використано Bootstrap, що є відкритим та безкоштовним HTML, CSS та JS фреймворком. Bootstrap використовується веб-розробниками для швидкої верстки адаптивного дизайну сайтів і веб-додатків. Основною областю застосування даного фреймворку є frontend розробка сайтів та інтерфейсів адміністративних панелей. Bootstrap є найбільш популярним серед наявних аналогів, оскільки допомагає створювати верстку сайту у декілька разів швидше. Даний фреймворк є доступним та дуже легким в освоєнні [16].

У процесі розробки інтернет-ресурсу було визначено, що елементи «header» та «footer» є однаковими для більшої частини сторінок веб-ресурсу. Для того, щоб оптимізувати та скоротити кількість коду додатку було створено папку «layouts», до якої пізніше додалися два файли «header.php» та «footer.php», у яких прописано шлях до css- та js- файлів, що наявні в проєкті, а також інші базові елементи веб-сайту. У представленнях, які використовують дані файли було додано наступний код:

```
<?php include_once ROOT.'/views/layouts/header.php' ?>  
<?php include_once ROOT.'/views/layouts/footer.php' ?>
```

Головна сторінка інтернет-магазину має наступний вигляд (Рис. 3.12):

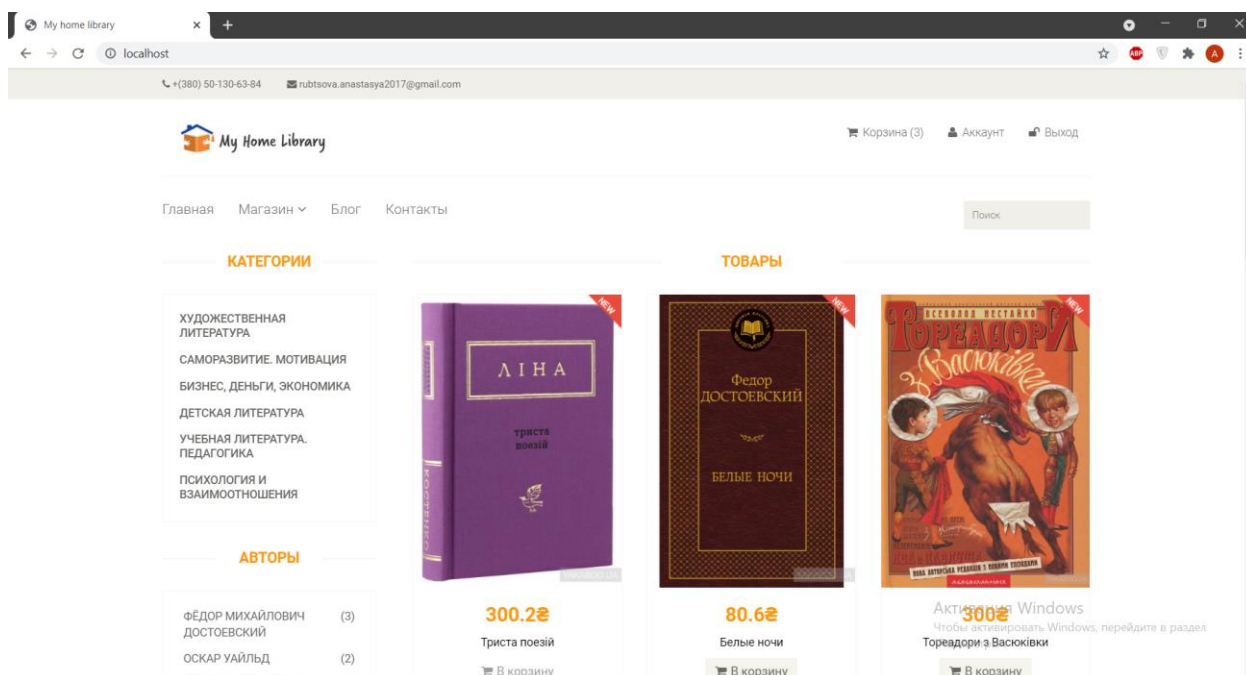


Рисунок 3.12 – Головна сторінка магазину «Моя домашня бібліотека»

Головний блок наповнений товарами, які наявні в інтернет-магазині. Основною інформацією є: назва книги, її ціна, фото, а також можливість додати до кошику. Якщо товару немає на складі, то кнопка «В корзину» буде неактивною. Дана задача реалізована за допомогою додання атрибуту «disabled» до кнопки. Якщо товар є новим у магазині, то у правому верхньому кутку фотографії книги буде додано відповідну позначку «new». Для того, щоб переглянути більш детальну інформацію про товар, необхідно натиснути на його назву (Рис. 3.13).

Користувач має можливість переглянути книги необхідного автора. Прізвища авторів та кількість їх книжок, наявних в магазині розміщені у відповідному блоці інтернет-ресурсу.

Лістинг запиту до бази даних, що відповідає за групування книжок відповідно до їх автора:

```
select author, author.name as name, count from
(select author.author_id as author, count(*) as count from
product
```

```

inner join author on product.author_id = author.author_id group
by author limit 7) temp
    inner join author on author.author_id = temp.author order by
count desc

```

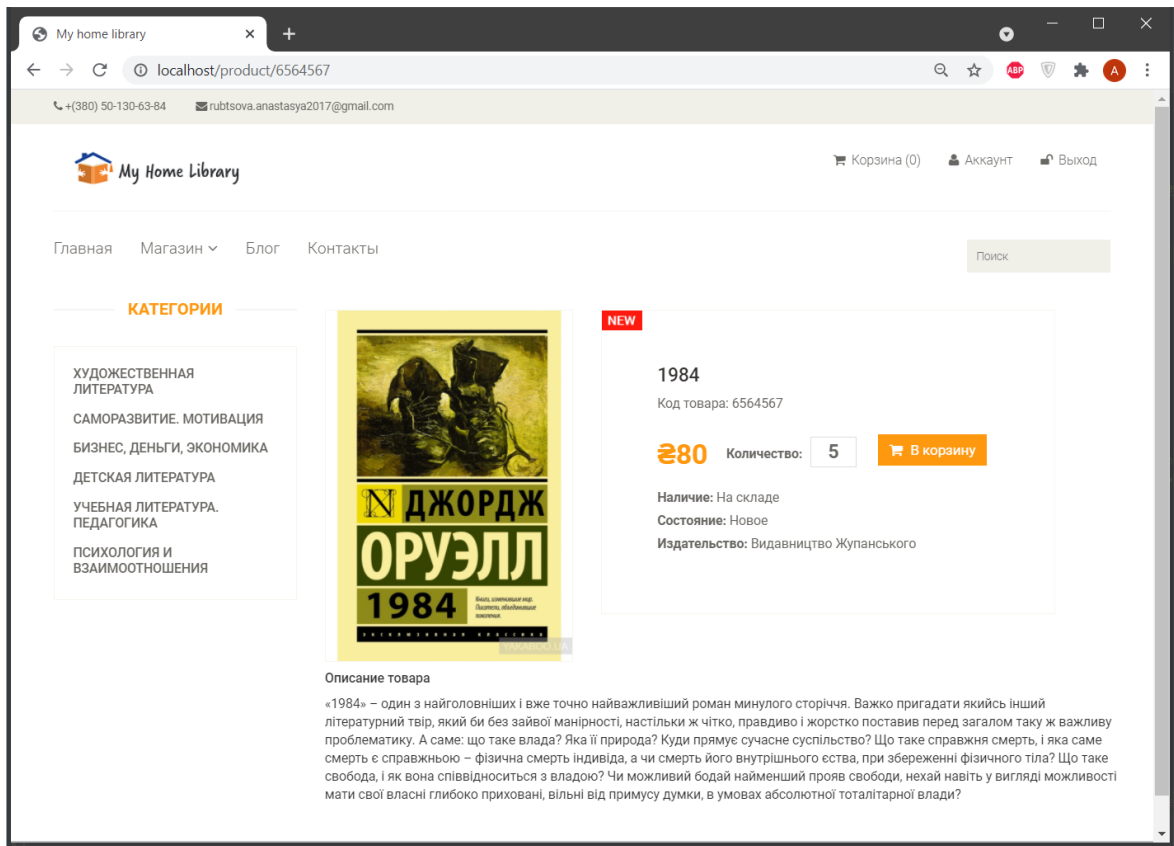


Рисунок 3.13 – Сторінка з детальною інформацією про товар

Інтернет-магазини зазвичай характеризуються великою кількістю товарів. Саме тому важливо важливим є створення зручного пошуку на сайті для того, щоб користувач мав можливість швидкого знаходження цікавої для нього книги. В інтернет-магазині «Домашня бібліотека» пошук має виконуватись за кодом товару (Рис. 3.16), назвою книжки (Рис. 3.17) та її автором (Рис. 3.18). Для того, щоб зробити процес пошуку більш швидким та комфортним для потенційного клієнта виникла необхідність реалізації живого пошуку на сайті. Даний вид пошуку характеризується тим, що у процесі введення інформації користувачем одразу виконується запит до бази даних, а також формується результат даного запиту у відповідне місце інтернет-ресурсу. Реалізація даного завдання можлива у разі використання технології

AJAX (Asynchronous JavaScript And XML), що характеризується зверненням до серверу без перезавантаження сторінки (Рис. 3.14) [17].

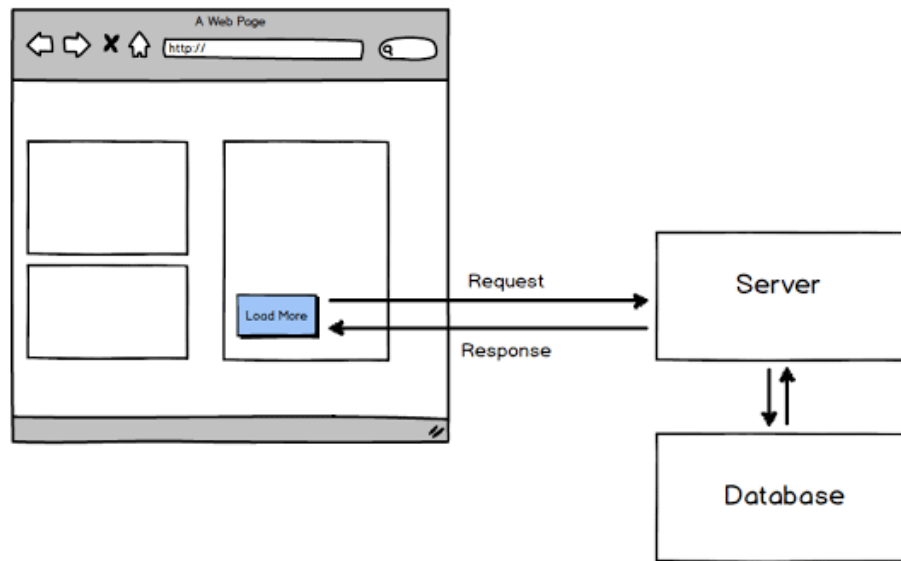


Рисунок 3.14 – Принцип роботи технології AJAX

Технологія AJAX в першу чергу є корисною для форм і елементів сторінки, пов'язаних з примітивними діями: додати до кошика, підписатись. За рахунок використання AJAX додаток стає більш швидким, оскільки зникає необхідність у перезавантаженні усєї сторінки сайту, натомість – стара інформація замінюється новою лише у необхідному блоці.

Технологія AJAX має наступні переваги:

- можливість створення зручного веб-інтерфейсу;
- забезпечення активної взаємодії з користувачем;
- часткове перезавантаження сторінки;
- зручність користування [18].

Процес реалізації пошуку на сайті виглядає наступним чином (Рис. 3.15):

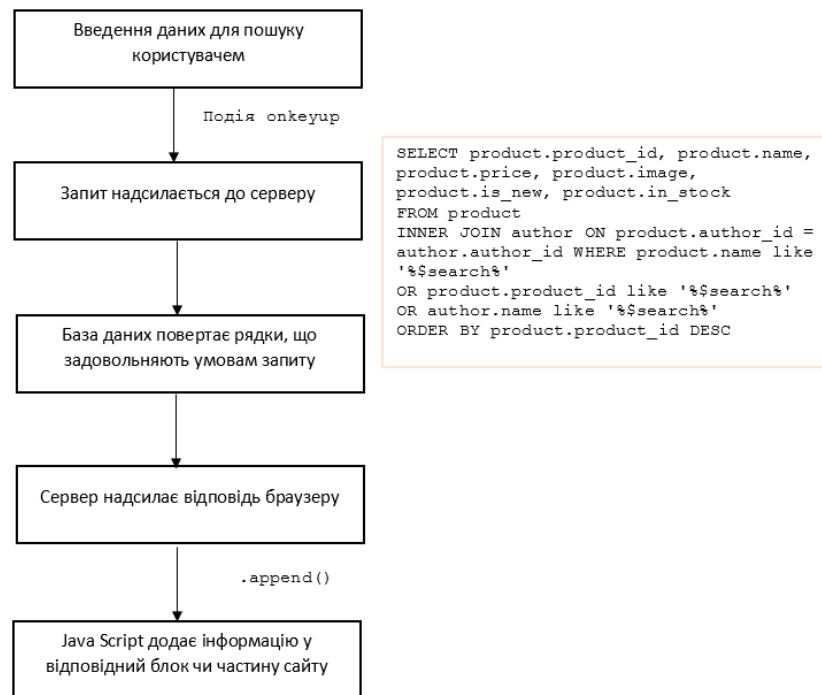


Рисунок 3.15 – Алгоритм реалізації пошуку на сайті

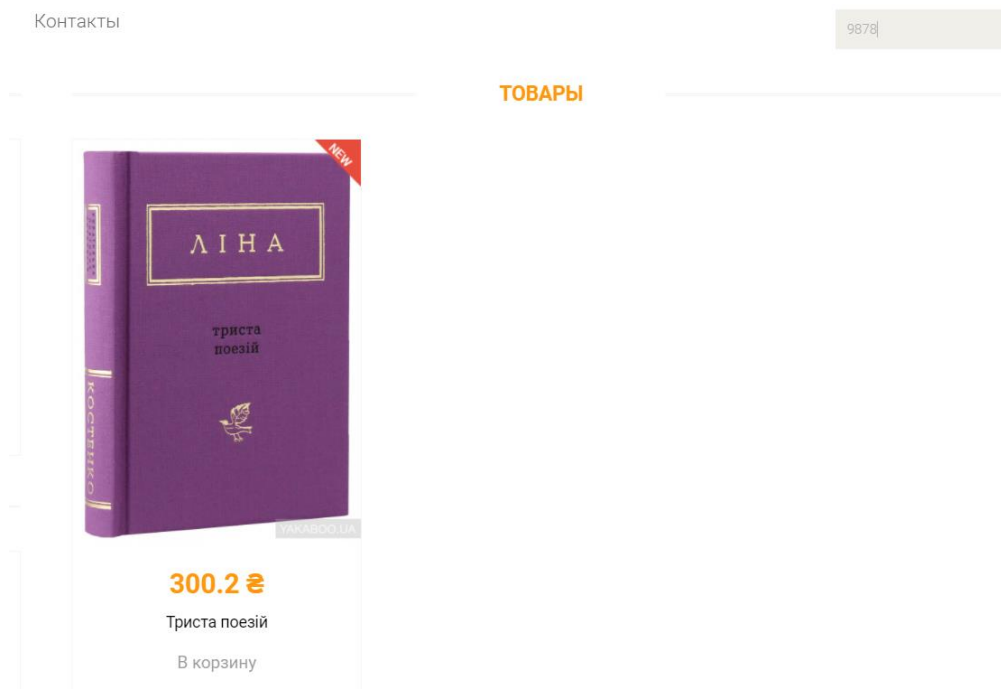


Рисунок 3.16 – Пошук книги за кодом

Контакты

1984

ТОВАРЫ



80 €

1984

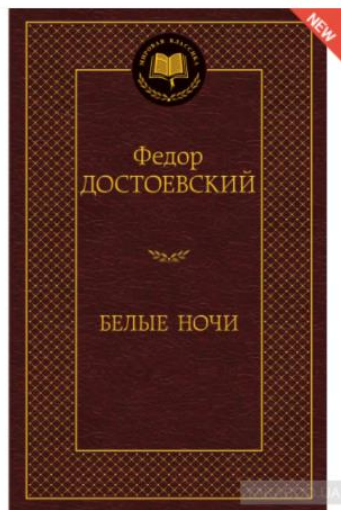
В корзину

Рисунок 3.17 – Пошук книги за її назвою

Контакты

досто

ТОВАРЫ



80.6 €

Белые ночи

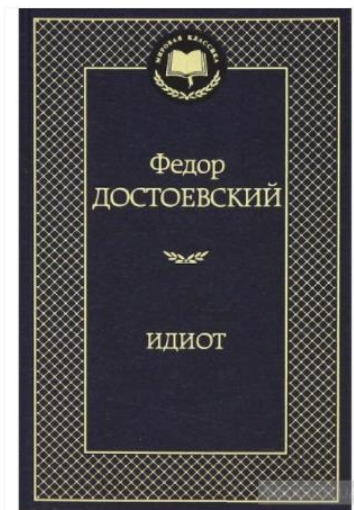
В корзину



59.5 €

Униженные и оскорбленные

В корзину



120.8 €

Идиот

В корзину

Рисунок 3.18 – Пошук книги за її автором

Головною відмінністю інтернет-магазину від звичайного веб-сайту є наявність широкого асортименту товарів. Даний товар користувач має можливість переглянути, обрати необхідну кількість, додати до кошика чи видалити з нього (Рис. 3.19). Під час перегляду кошика користувачу доступна найбільш важлива інформація про товар (код, назва, ціна, кількість), а також загальна ціна замовлення.

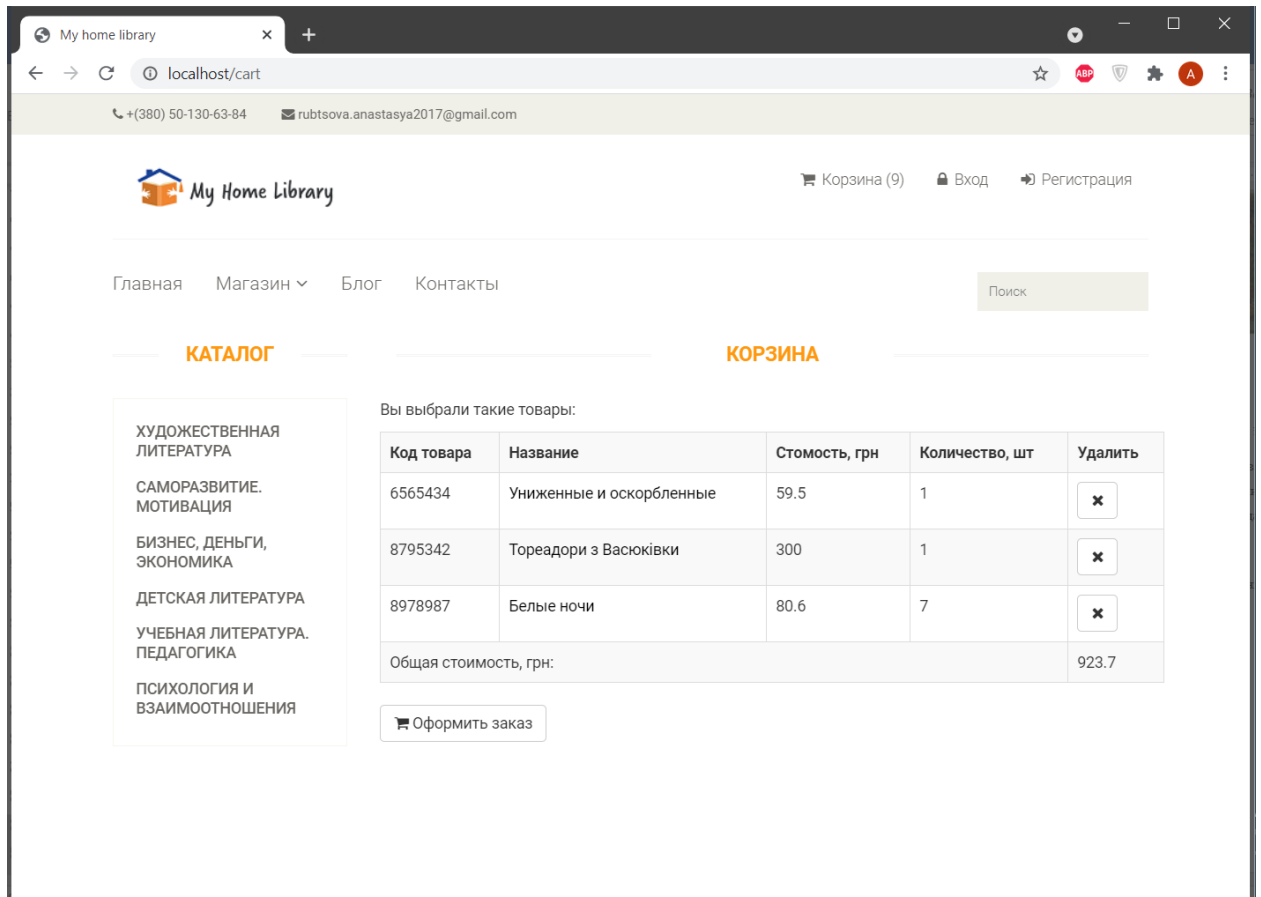


Рисунок 3.19 – Кошик користувача

Подальшими діями користувача є оформлення замовлення. Якщо користувач авторизований в системі, то після натиснення кнопки «Оформить заказ» йому буде доступна форма з полями «имя пользователя», що заповнюється автоматично (дані про користувача отримуються з сесії) та «комментарий к заказу» (Рис. 3.20). Якщо користувач не є авторизованим у системі, то результатом натиснення кнопки «Оформит заказ» стане форма для авторизації. Якщо користувач не зареєстрований, то він заповнити поля форми

для реєстрації, що доступна за рахунок натиснення кнопки «Регистрация» у шапці сайту (Рис. 3.21).

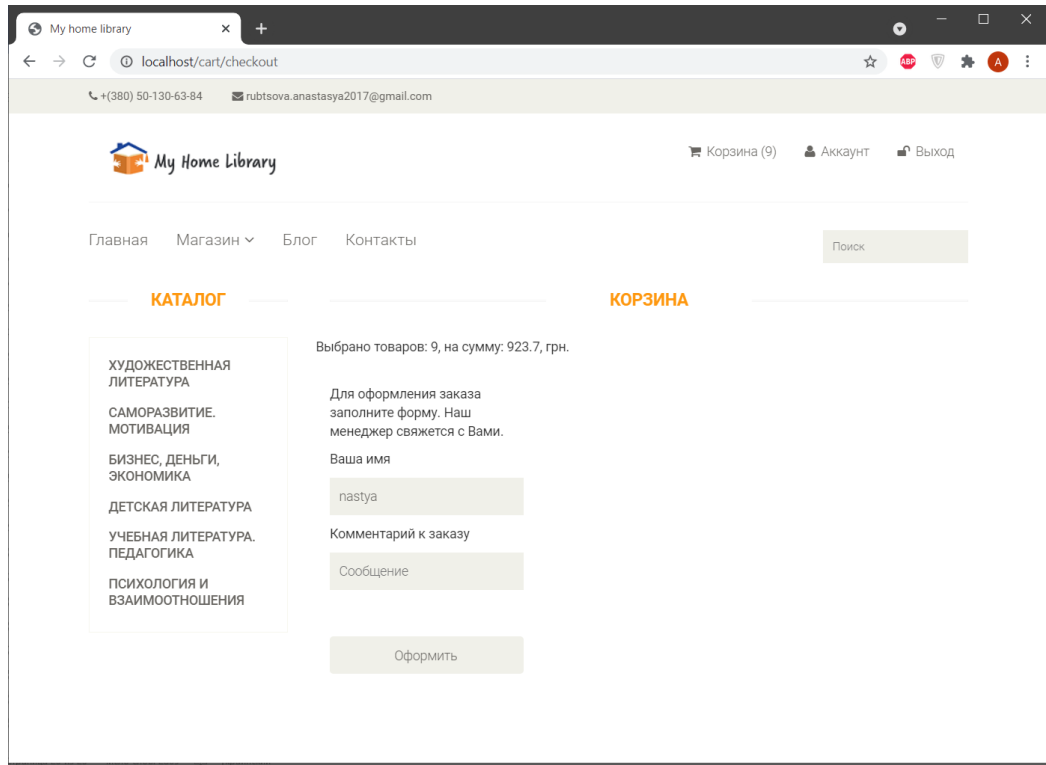


Рисунок 3.20 – Форма оформлення замовлення

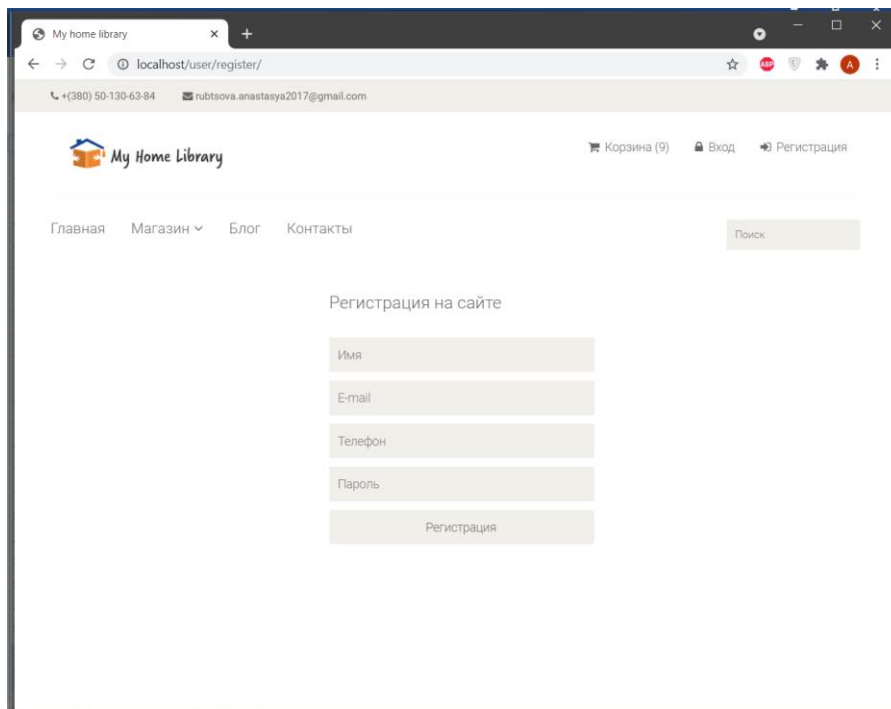


Рисунок 3.21 – Форма реєстрації користувача

Користувач, що є авторизованим у системі має можливість переглянути перелік своїх нещодавніх покупок, натиснувши кнопку «Аккаунт», дана сторінка є кабінетом користувача (Рис. 3.22).

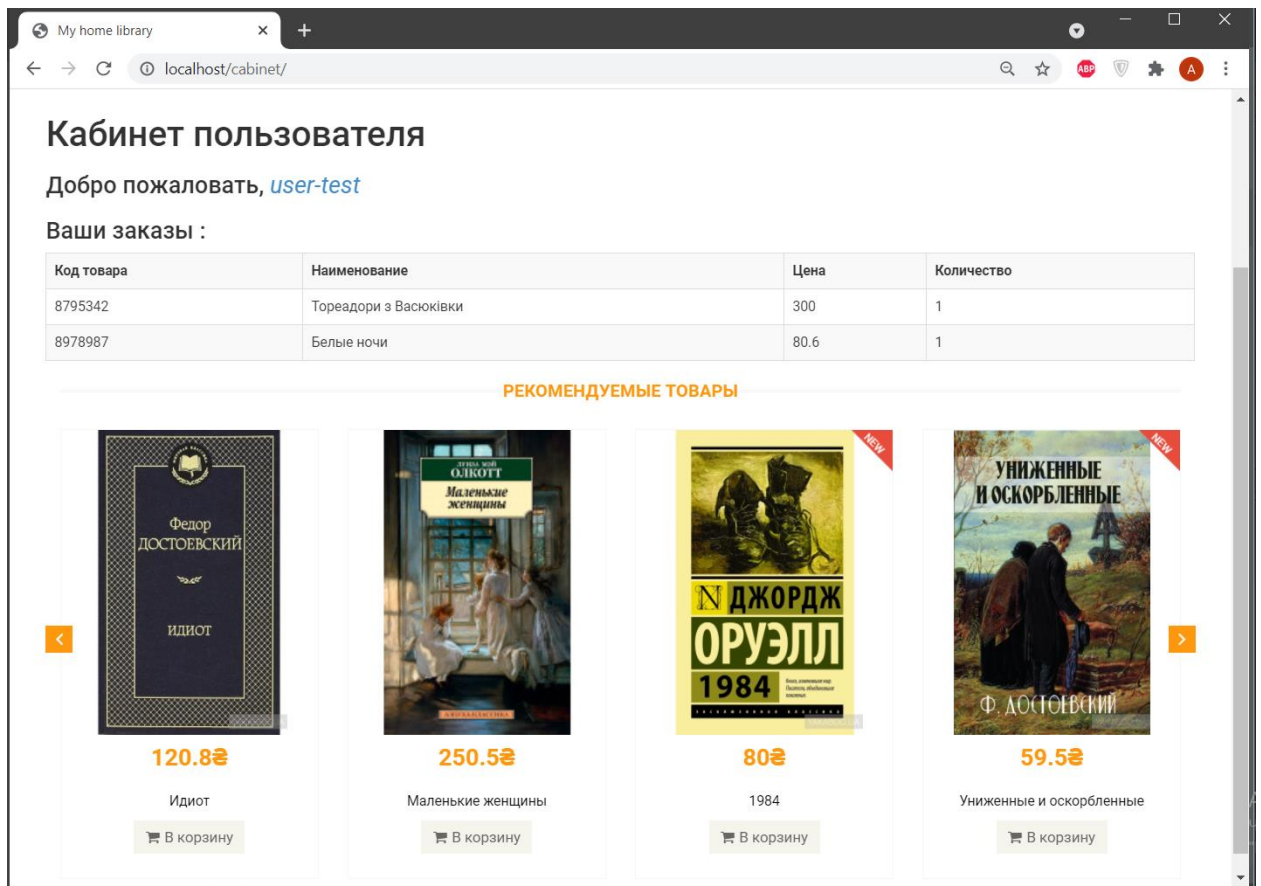


Рисунок 3.22 – Кабинет користувача

В залежності від попередніх замовлень користувача, формується слайдер з товарами, що є рекомендованими особисто для нього.

Лістинг запиту до бази даних, що повертає перелік рекомендованих для користувача товарів:

```
$sql= "SELECT product.product_id, product.name,
product.price, product.image, product.in_stock, product.is_new
FROM product
inner join category on product.category_id =
category.category_id
WHERE product.product_id not in (select product_id from
product_order where user_id = :id) and category.category_id in
```

```
(select product.category_id from product inner join category
on product.category_id = category.category_id
inner join product_order on product.product_id =
product_order.product_id where product_order.user_id = :id)
LIMIT 4";
```

Однією з головних особливостей інтернет-магазину є блог, де користувач має можливість знайти та прочитати цікаві для нього статті. Записи блогу розміщені у хронологічній послідовності для того, щоб користувач мав змогу ознайомитися з новою та актуальною інформацією (Рис. 3.23).

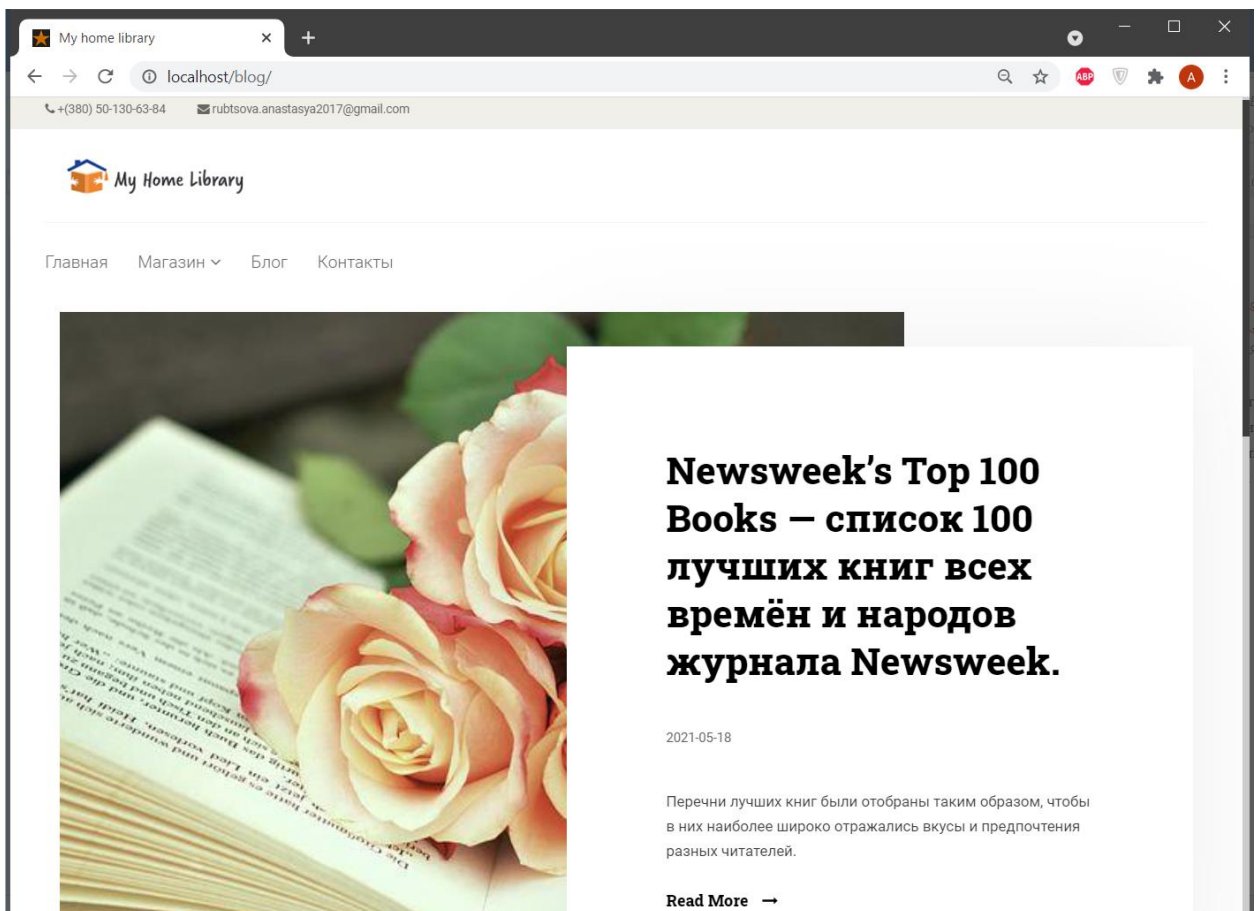


Рисунок 3.23 – Сторінка блогу

Запис блогу складається з наступної інформації: тема публікації, її дата, стислий опис, фото та посилання на повну статтю (Рис. 3.24).

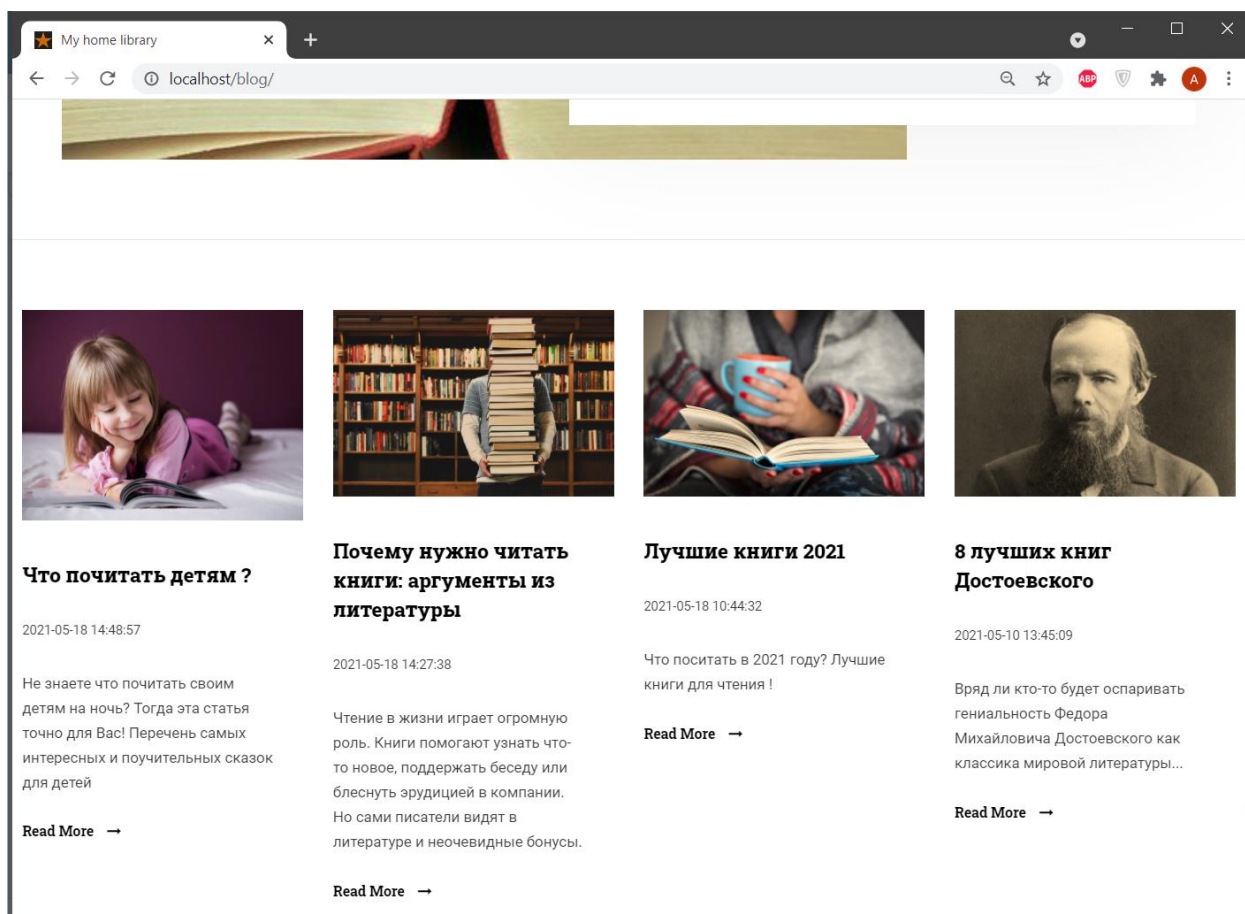


Рисунок 3.24 – Сторінка блогу, продовження

Будь-який сайт повинен часу від часу поповнюватися новою та актуальною інформацією. Саме тому одним з найважливіших етапів є створення зручної адміністративної панелі. Дана панель представляє собою частину сайту, що є прихованою від користувачів. Вхід до неї можливий лише деяким користувачам сайту, тобто адміністраторам інтернет-ресурсу. Для цього до таблиці «user» було додано поле «role». Якщо користувач є адміністратором сайту дане поле набуває значення «admin», якщо ні – залишається порожнім. Перед тим, як потрапити до адміністративної частини сайту користувач має авторизуватися на сайті, ввівши поштову скриньку та пароль (Рис. 3.25).

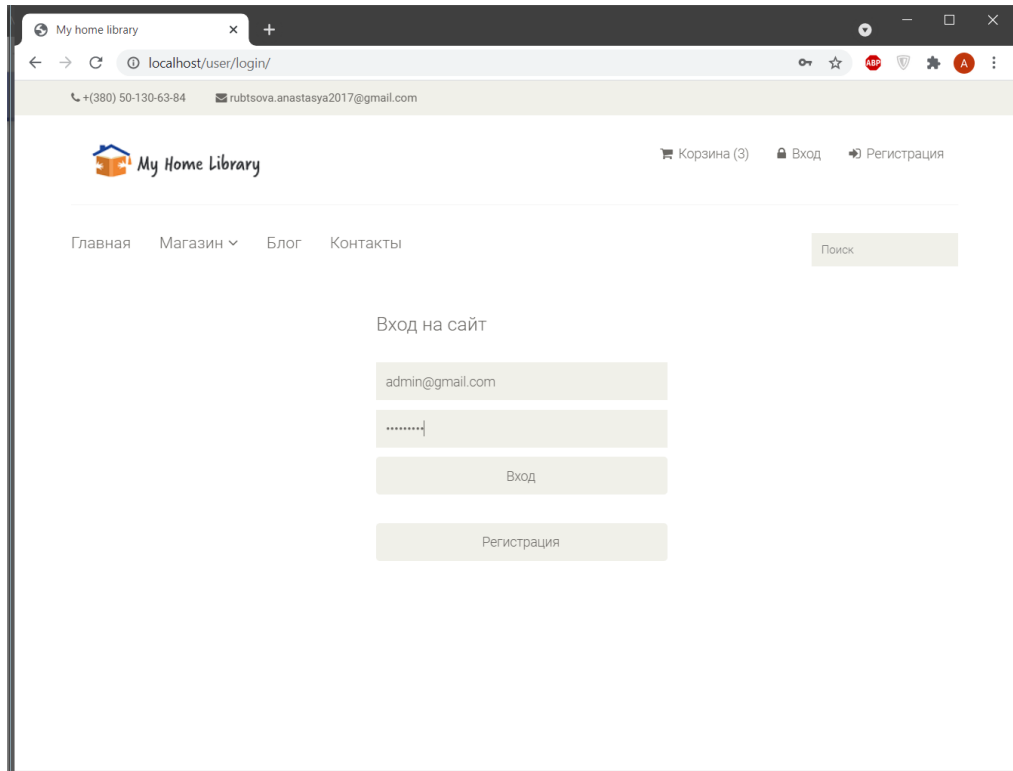


Рисунок 3.25 – Вхід до адміністративної панелі

Якщо авторизація пройшла успішно, то ми маємо можливість перейти до адміністративної частини, ввівши у пошуковому рядку «localhost/admin». Дана сторінка містить перелік доступних для адміністратора можливостей (Рис. 3.26).

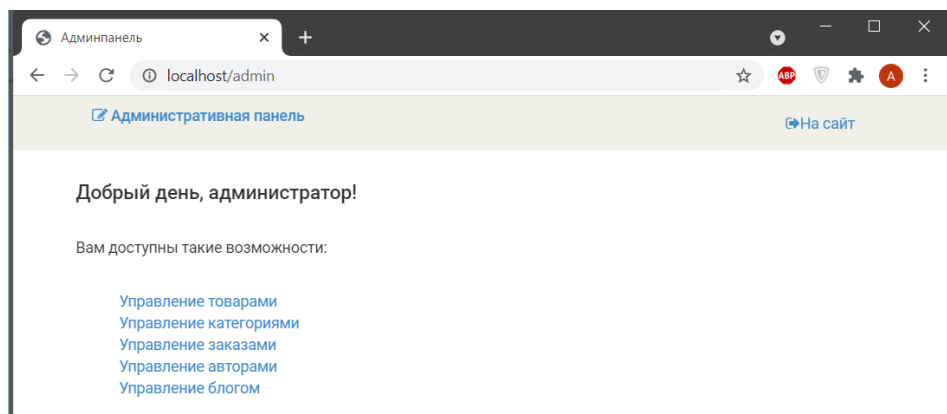


Рисунок 3.26 – Перелік доступних для адміністратора можливостей

Адміністративна панель інтернет-магазину повинна підтримувати операції CRUD: create, read, update, delete. У MySQL даним операціям

відповідають запити insert, select, update, delete [19]. Адміністратору доступні маніпуляції над наявними товарами, категоріями, замовленнями, а також авторами книжок. Для внесення необхідних змін необхідно обрати відповідний розділ (Рис. 3.27).

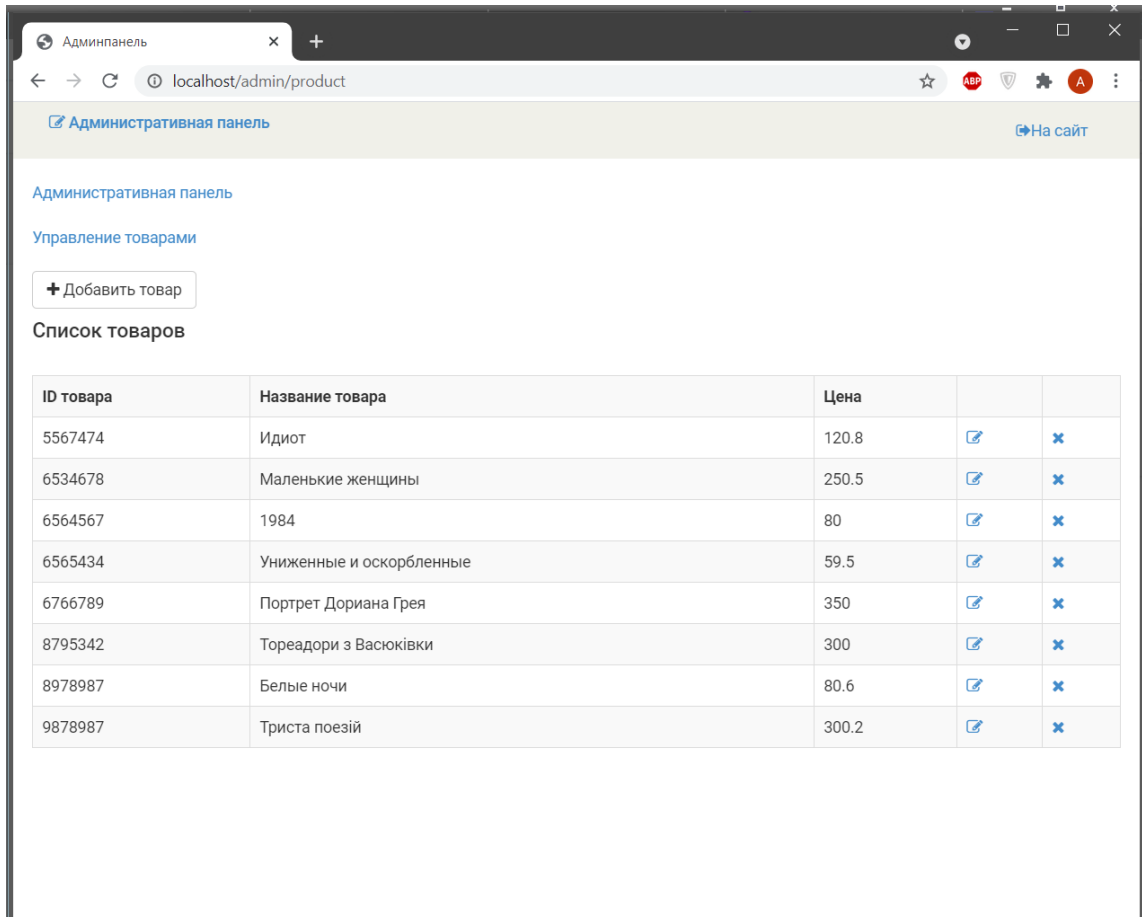


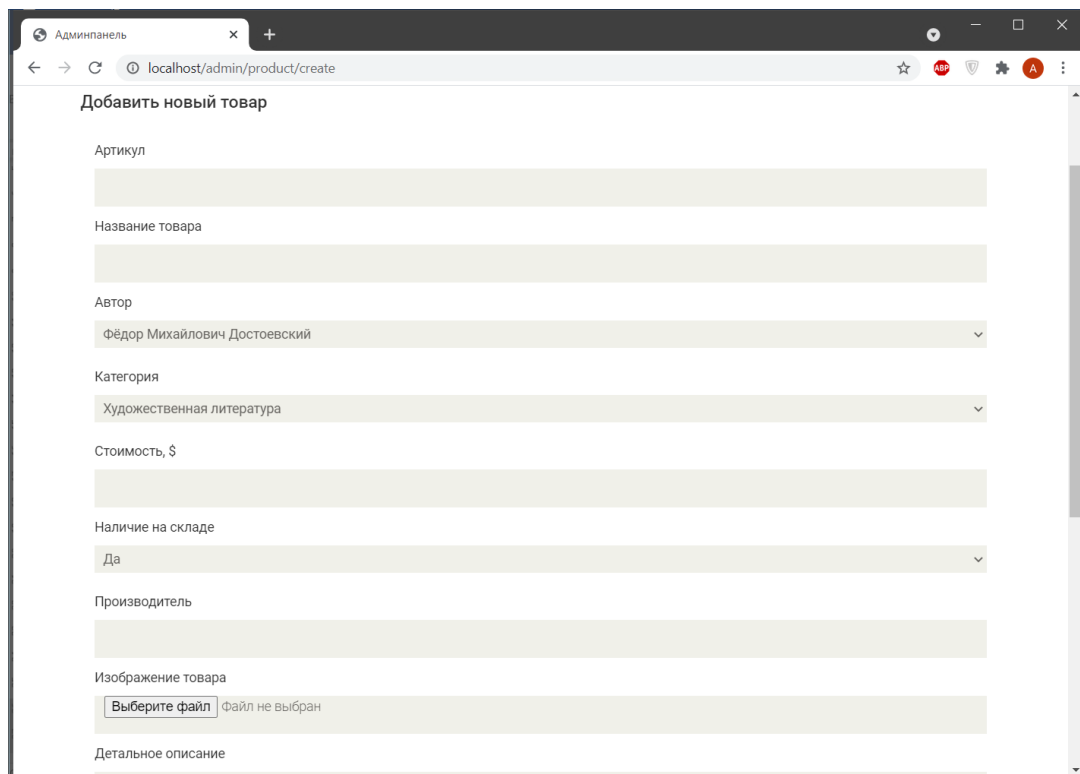
Рисунок 3.27 – Адміністративна панель, управління товарами, перегляд

Адміністратор сайту має можливість переглянути основну інформацію (код, назву, ціну) про товар, що представлена у вигляді таблиці, а також редагувати чи видалити необхідний товар, натиснувши на відповідні значки.

Функція перегляду списку наявних товарів:

```
public static function getProductsList()
{
    $db = Db::getConnection();
    $result = $db->query('SELECT product_id, name, price
FROM product ORDER BY product_id ASC');
    $productsList = array();
    $i = 0;
    while ($row = $result->fetch()) {
        $productsList[$i]['product_id'] = $row['product_id'];
        $productsList[$i]['name'] = $row['name'];
        $productsList[$i]['price'] = $row['price'];
        $i++;
    }
    return $productsList;
}
```

Для того, щоб додати новий товар необхідно натиснути кнопку «Добавить товар», з'явиться форма для заповнення новою інформацією (Рис. 3.28).



The screenshot shows a web browser window with the URL `localhost/admin/product/create`. The page title is "Добавить новый товар". The form contains the following fields:

- Артикул:
- Название товара:
- Автор:
- Категория:
- Стоимость, \$:
- Наличие на складе:
- Производитель:
- Изображение товара: файл не выбран
- Детальное описание:

Рисунок 3.28 – Адміністративна панель, управління товарами, додавання нового товару

Функція додавання нового товару виглядає наступним чином:

```

public static function createProduct($options){
    $db = Db::getConnection();
    $sql = 'INSERT INTO product '
        . '(product_id, name, author_id, category_id, price,
availability, brand, '
        . 'image, description, is_new, is_recommended, status,
in_stock) '
        . 'VALUES '
        . '(:product_id, :name, :author_id, :category_id,
:price, :availability, :brand, '
        . ':image, :description, :is_new, :is_recommended,
:status, :in_stock)';
    $result = $db->prepare($sql);
    $result->bindParam(':product_id', $options['product_id'],
PDO::PARAM_INT);
    $result->bindParam(':name', $options['name'],
PDO::PARAM_STR);
    $result->bindParam(':author_id', $options['author_id'],
PDO::PARAM_INT);
    $result->bindParam(':category_id', $options['category_id'],
PDO::PARAM_INT);
    $result->bindParam(':price', $options['price'],
PDO::PARAM_STR);
    $result->bindParam(':availability',
$options['availability'], PDO::PARAM_INT);
    $result->bindParam(':brand', $options['brand'],
PDO::PARAM_STR);
    $result->bindParam(':image', $options['image'],
PDO::PARAM_STR);
    $result->bindParam(':description', $options['description'],
PDO::PARAM_STR);
    $result->bindParam(':is_new', $options['is_new'],
PDO::PARAM_INT);
    $result->bindParam(':is_recommended',
$options['is_recommended'], PDO::PARAM_INT);
    $result->bindParam(':status', $options['status'],
PDO::PARAM_INT);
    $result->bindParam(':in_stock', $options['in_stock'],
PDO::PARAM_INT);
    return $result->execute();
}

```

У процесі редагування наявної інформації адміністратор працює також з формою, яка містить дані про товар (Рис. 3.29).

Админпанель

localhost/admin/product/update/5567474

Редактировать товар #5567474

Артикул
5567474

Название товара
Идиот
Фёдор Михайлович Достоевский

Категория
Художественная литература

Стоимость, \$
120.8

Наличие на складе
Да

Производитель
Азбука

Изображение товара
Выберите файл book4.jpg

Детальное описание
"Главная идея... - писал Ф.М.Достоевский о своем романе "Идиот", - изобразить положительно-прекрасного человека. Труднее этого нет ничего на свете..." Не для того ли писатель явил миру "князя-Христа", чтобы мы не забывали: "Страдание есть главнейший и может быть"

Рисунок 3.29 – Адміністративна панель, управління товарами, редагування наявного товару

Функція редагування наявного товару виглядає наступним чином:

```
public static function updateProductById($id, $options)
{
    $db = Db::getConnection();
    $sql = "UPDATE product
        SET product_id = :product_id, name = :name, author_id =
:author_id, category_id = :category_id, price = :price,
availability = :availability, brand = :brand, image = :image,
description = :description, is_new = :is_new, is_recommended =
:is_recommended, status = :status, in_stock = :in_stock WHERE
product_id = :id";
    $result = $db->prepare($sql);
    $result->bindParam(':id', $id, PDO::PARAM_INT);
    $result->bindParam(':product_id', $options['product_id'],
PDO::PARAM_INT);
    $result->bindParam(':name', $options['name'],
PDO::PARAM_STR);
```

```

        $result->bindParam(':author_id', $options['author_id'],
PDO::PARAM_INT);
        $result->bindParam(':category_id', $options['category_id'],
PDO::PARAM_INT);
        $result->bindParam(':price', $options['price'],
PDO::PARAM_STR);
        $result->bindParam(':availability',
$options['availability'], PDO::PARAM_INT);
        $result->bindParam(':brand', $options['brand'],
PDO::PARAM_STR);
        $result->bindParam(':image', $options['image'],
PDO::PARAM_STR);
        $result->bindParam(':description', $options['description'],
PDO::PARAM_STR);
        $result->bindParam(':is_new', $options['is_new'],
PDO::PARAM_INT);
        $result->bindParam(':is_recommended',
$options['is_recommended'], PDO::PARAM_INT);
        $result->bindParam(':status', $options['status'],
PDO::PARAM_INT);
        $result->bindParam(':in_stock', $options['in_stock'],
PDO::PARAM_INT);
        return $result->execute();
    }

```

Для видалення товару з переліку необхідно натиснути відповідний значок та підтвердити видалення обраного товару (Рис. 3.30).

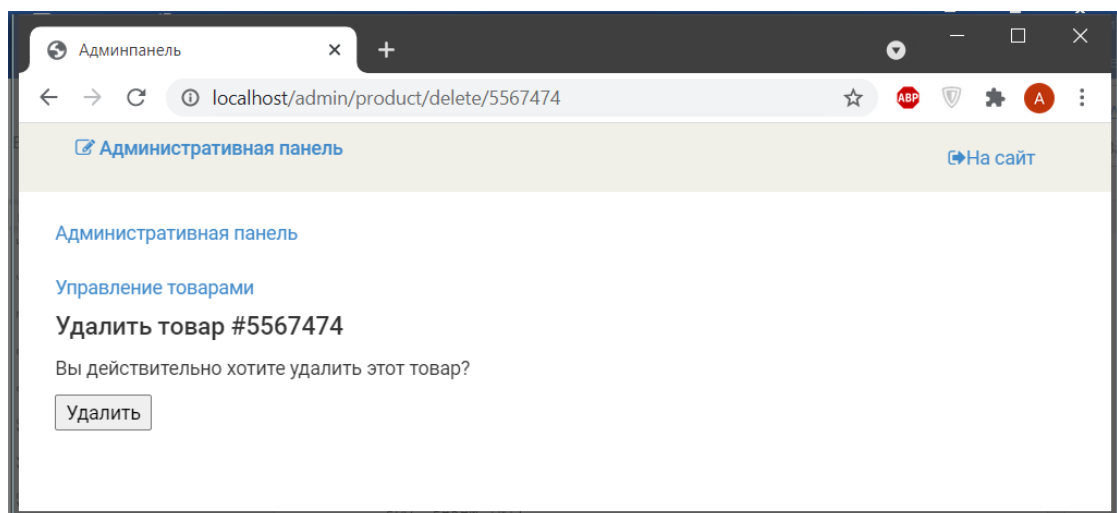


Рисунок 3.30 – Адміністративна панель, управління товарами, видалення товару

Функція видалення обраного товару:

```

public static function deleteProductById($id)
{
    $db = Db::getConnection();
    $sql = 'DELETE FROM product WHERE product_id = :id';
    $result = $db->prepare($sql);
    $result->bindParam(':id', $id, PDO::PARAM_INT);
    return $result->execute();
}

```

Процес перегляду, додавання, редагування та видалення категорій та авторів книжок є схожим до наведених вище маніпуляцій з товарами. Таблиця з даними про замовлення є більш інформативною для більш зручної роботи адміністратора з замовленнями користувачів (Рис. 3.31).

ID заказа	Имя покупателя	Телефон покупателя	Дата оформления	Статус			
27	fedor0506	0501645674	2021-05-03 21:08:03	Новый заказ	👁	✎	✖
26	fedor0506	0501645674	2021-05-03 20:29:33	Новый заказ	👁	✎	✖
25	fedor0506	0501645674	2021-05-03 20:14:25	Новый заказ	👁	✎	✖
24	fedor0506	0501645674	2021-05-03 20:14:25	Новый заказ	👁	✎	✖
23	fedor0506	0501645674	2021-05-03 20:08:21	Новый заказ	👁	✎	✖
22	fedor0506	0501645674	2021-05-03 20:07:45	Новый заказ	👁	✎	✖
21	milana	506745654	2021-05-03 19:43:40	Новый заказ	👁	✎	✖
20	anastasia0507	0501306384	2021-05-03 19:24:54	Новый заказ	👁	✎	✖
19	anastasia0507	0501306384	2021-05-03 19:24:54	Новый заказ	👁	✎	✖
18	anastasia0507	0501306384	2021-05-03 19:10:38	Новый заказ	👁	✎	✖
17	anastasia0507	0501306384	2021-05-03 19:10:38	Новый заказ	👁	✎	✖
16	anastasia0507	0501306384	2021-05-03 18:37:27	Новый заказ	👁	✎	✖
14	anastasia0507	0501306384	2021-05-03 18:30:18	Новый заказ	👁	✎	✖
13	anastasia0507	0501306384	2021-05-03 18:28:21	Новый заказ	👁	✎	✖
12	anastasia0507	0501306384	2021-05-03 17:52:41	Новый заказ	👁	✎	✖
11	anastasia0507	0501306384	2021-05-03 17:51:18	Новый заказ	👁	✎	✖

Рисунок 3.31 – Адміністративна панель, управління замовленнями, перегляд

Для отримання детальної інформації про користувача необхідним стало об'єднання двох таблиць «user» та «product_order».

Функція перегляду наявних замовлень:

```
public static function getOrderById($id)
{
    $db = Db::getConnection();
    $sql = 'SELECT user.name, user.phone, product_order.* FROM
product_order INNER JOIN user ON user.user_id =
product_order.user_id WHERE product_order.id = :id';

    $result = $db->prepare($sql);
    $result->bindParam(':id', $id, PDO::PARAM_INT);
    $result->setFetchMode(PDO::FETCH_ASSOC);
    $result->execute();
    return $result->fetch();
}
```

Для створення повноцінного інтернет-ресурсу програмістам необхідно приділити увагу усім деталям, що стосуються зовнішнього вигляду сайту (Рис. 3.32). Одним з найбільш вагомих є забезпечення адаптивності. Адаптивний інтернет-ресурс – це інтернет-ресурс з таким дизайном, що однаково добре виглядає на усіх пристроях: ПК, ноутбуках, планшетах чи телефонах (Рис. 3.33). Дизайн сайту адаптується під параметри вікна браузера тим самим забезпечуючи користувачу комфортне сприйняття інформації.

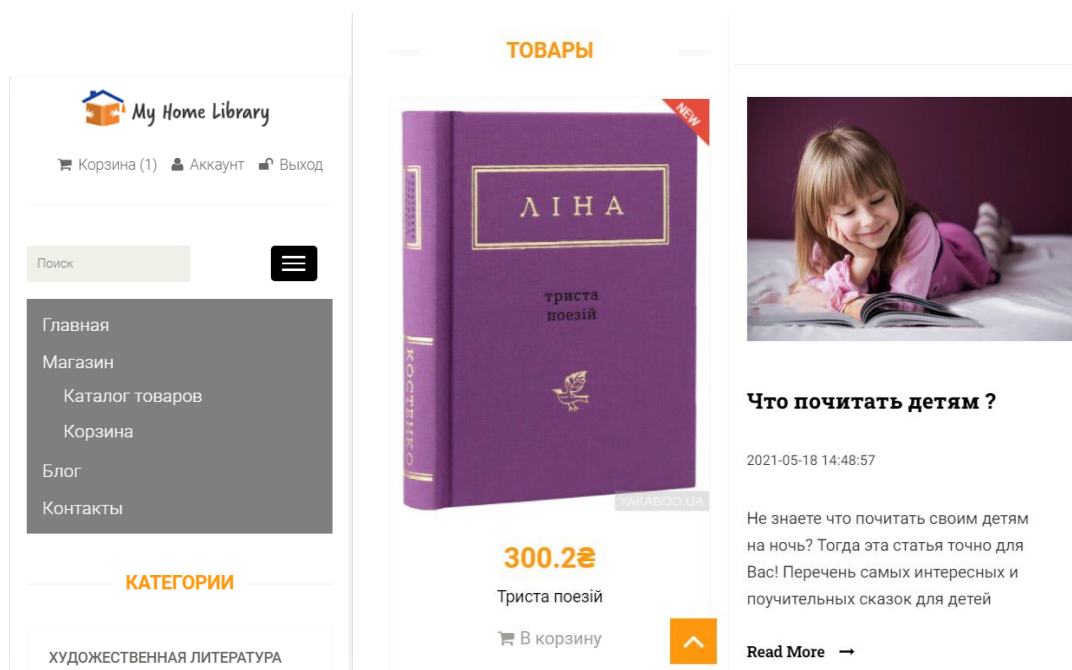


Рисунок 3.32 – Вигляд інтернет-магазину «Домашня бібліотека» з телефону

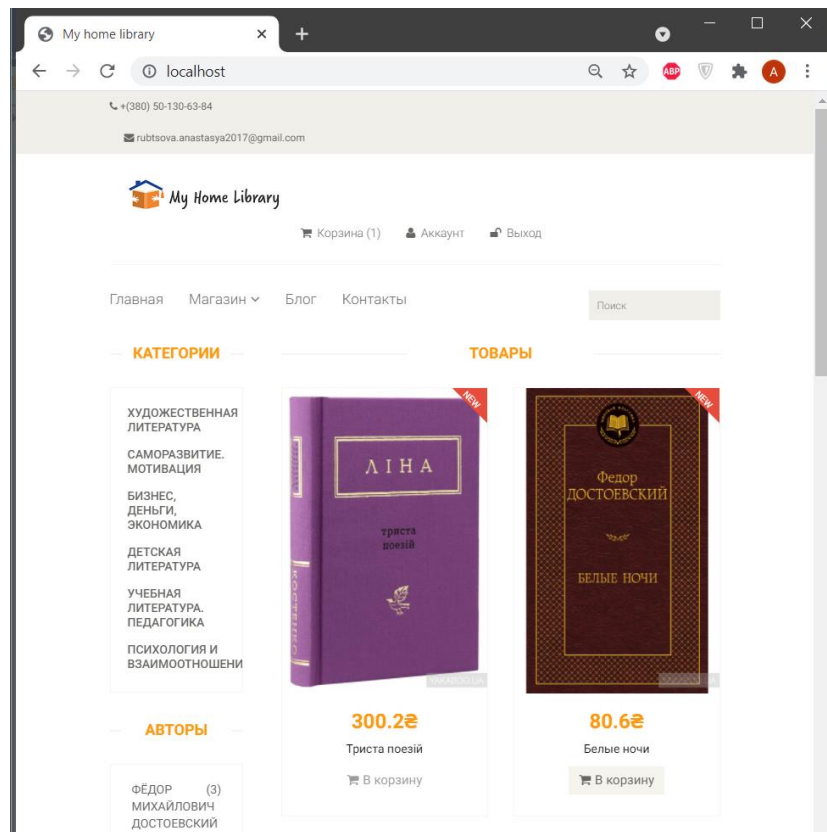


Рисунок 3.33 – Вигляд інтернет-магазину «Домашня бібліотека» при зміні параметрів вікна браузера

ВИСНОВКИ

Нині важко переоцінити роль, яку відіграє інформація в нашому житті. Вона стала основним ресурсом суспільства, базою його соціального і технічного розвитку. Всі галузі людської діяльності зазнали великих змін завдяки процесу інформатизації, що почався в середині ХХ століття. Як наслідок, відбувся бурхливий розвиток інформаційних технологій, вплив яких став неминучим.

Використання новітніх розробок значно спрощує наше життя. Більшість з них пропонує автоматизацію наявних процесів, шляхом експлуатації програмних продуктів, розроблених для вирішення проблем користувачів.

Метою роботи є автоматизація наявних робочих процесів власників магазину книг «Домашня бібліотека». Розробка інтернет-магазину з можливістю зручного перегляду товарів, швидкого оформлення замовлення є найкращим рішенням даного завдання. У процесі роботи над даним проектом було розглянуто та проаналізовано наявні книжкові інтернет-магазини, визначено основні функціональні вимоги до даного виду інтернет-ресурсів. Важливим етапом став вибір методів вирішення задачі. З метою поглиблення наявних знань та вдосконалення практичних навичок програмування було вирішено створити власний міні-фреймворк, що є реалізованим на основі шаблону проектування MVC. Уся інформація, що необхідна для повноцінного функціонування інтернет-ресурсу зберігається у СУБД MySQL. За рахунок використання сучасних технологій, AJAX та Javascript, інтернет-магазин став більш швидким та динамічним.

Отже, наразі інтернет-ресурс є функціональним та інформаційним. Користувачі сайту мають можливість у будь-який час переглянути та замовити бажані товари з будь-якого пристрою, оскільки сайт є адаптивним. Завдяки реалізації поставленого завдання адміністратори магазину «Домашня бібліотека» мають можливість вести бізнес онлайн.

СПИСОК ЛІТЕРАТУРИ

- 1 Мальцев И. П. Проектирование сайтов : монографія /И.П. Мальцев – Нижний Новгород : Издательство SelfPub, 2018. – 12 с.
- 2 Шиколенков Т. А. Ваш интернет-магазин от А до Я : монографія / Т. А. Шиколенков – Россия : Издательство Шиколенков Т. А., 2018. – 43 с.
- 3 Ткаченко В., Шварц Б., Зайцев П. MySQL по максимуму монографія / В. Ткаченко, Б. Шварц, П. Зайцев – Санкт-Петербург: Издательство Питер, 2018. – 20 с.
- 4 Интернет-магазин – необходимая составляющая бизнеса [Электронный ресурс] / Электрон. текст. дані. – Режим доступа : <https://www.alkosto.ru/blog/internet-magazin-neobhodimaya-sostavlyayushhaya-biznesa/>.
- 5 Laravel – лидер среди PHP фреймворков, одобренный разработчиками [Электронный ресурс] / Электрон. текст. дані. – Режим доступа : <https://webformymself.com/laravel-lider-sredi-php-frejmworkov-odobrennyj-razrabotchikami/>.
- 6 WordPress [Электронный ресурс] / Электрон. текст. дані. – Режим доступа : <https://astwellsoft.com/ru/blog/cms/wordpress.html>.
- 7 Фреймворк Laravel [Электронный ресурс] / Электрон. текст. дані. – Режим доступа : <https://unetway.com/blog/laravel-framework-review>.
- 8 Лучшие backend фреймворки и библиотеки PHP в 2020 [Электронный ресурс] / Ю. Мусиенко – Электрон. текст. дані. – Режим доступа : <https://merehead.com/ru/blog/php-frameworks-2020/>.
- 9 Рейтинги популярности языков программирования и СУБД в 2019 году [Электронный ресурс] / Электрон. текст. дані. – Режим доступа : <https://www.opennet.ru/opennews/art.shtml?num=52174>.
- 10 Что такое MySQL [Электронный ресурс] / Макхост– Электрон. текст. дані. – Режим доступа : <https://mchost.ru/articles/chto-takoe-mysql/>.

11 Концепция MVC для чайников [Электронный ресурс/ С. Фастунов – Электрон. текст. дані. – Режим доступа : <https://ruseller.com/lessons.php?id=666>.

12 Основы проектирования баз данных. Создание баз данных и таблиц [Электронный ресурс] / Электрон. текст. дані. – Режим доступа : <https://metanit.com/sql/tutorial/1.1.php>.

13 100+ питань для співбесід та технічних інтерв'ю – бази даних, зокрема SQL (частина 1) [Электронный ресурс] / Электрон. текст. дані. – Режим доступа : <https://www.quality-assurance-group.com/super-puper-shpargalka-100-pytan-dlya-spivbesid-ta-tehnichnyh-intervyu-bazy-danyh-zokrema-sql-chastyna-1/>.

14 Хранение паролей в базе данных. Md5() + соль [Электронный ресурс] / Электрон. текст. дані. – Режим доступа : https://housecomputer.ru/technology/algorithm/md5/md5_salt/md5_salt.html.

15 Подготовленные запросы и хранимые процедуры [Электронный ресурс]: зб. наук. пр. – 2000-2021. – Электрон. текст. дані. – Режим доступа: <https://www.php.net/manual/ru/pdo.prepared-statements.php>.

16 Что такое Bootstrap и зачем он нужен? [Электронный ресурс] / А.Мальцев – Электрон. текст. дані. – Режим доступа : <https://itchief.ru/bootstrap/introduction>.

17 Введение в AJAX для фронтенд-разработчиков [Электронный ресурс] / G. Martsoukos – Электрон. текст. дані. – Режим доступа : <https://webdesign.tutsplus.com/ru/tutorials/an-introduction-to-ajax-for-front-end-designers--cms-25099>.

18 AJAX для новичков [Электронный ресурс] / Д. Вершинин – Электрон. текст. дані. – Режим доступа : <https://habr.com/ru/post/14246/>.

19 CRUD операции [Электронный ресурс] / А. Ивашкевич – Электрон. текст. дані. – Режим доступа : <https://webshake.ru/oop-v-php-prodvinityj-kurs/chto-takoe-crud-operacii>.

ДОДАТКИ

Додаток А

Контролер «CatalogController»:

```

<?php
class CatalogController
{
    public function actionIndex(){
        $categories = array();
        $categories = Category::getCategoriesList();
        $authors = array();
        $authors = Author::getAuthorsList();
        $latestProducts = array();
        $latestProducts = Product::getLatestProducts(10);
        require_once (ROOT.'/views/catalog/index.php');
        return true;
    }
    public function actionCategory($categoryId, $page = 1){
        $categories = array();
        $categories = Category::getCategoriesList();
        $authors = array();
        $authors = Author::getAuthorsList();
        $categoryProducts = array();
        $categoryProducts =
Product::getProductsListByCategory($categoryId, $page);
        $total =
Product::getTotalProductsInCategory($categoryId);
        $pagination = new Pagination($total, $page,
Product::SHOW_BY_DEFAULT, 'page-');
        require_once (ROOT.'/views/catalog/category.php');
        return true;
    }
    public function actionAuthor($authorId){
        $categories = array();
        $categories = Category::getCategoriesList();
        $authors = array();
        $authors = Author::getAuthorsList();
        $categoryProducts = array();
        $categoryProducts =
Product::getProductsListByAuthor($authorId);
        require_once (ROOT.'/views/catalog/author.php');
        return true;
    }
}
}

```

Додаток Б

Модель «Product»:

```

<?php

class Product
{
    const IMAGES_COUNT = 3;
    public static function getLatestProducts($count =
self::IMAGES_COUNT)
    {
        $db = Db::getConnection();
        $productList = array();
        $result = $db->query('SELECT product_id, name,
price, image, is_new, in_stock FROM product WHERE status = "1"
ORDER BY product_id DESC LIMIT ' . $count);
        $i = 0;
        while ($row = $result->fetch_row()) {
            $productList[$i]['product_id'] = $row[0];
            $productList[$i]['name'] = $row[1];
            $productList[$i]['price'] = $row[2];
            $productList[$i]['image'] = $row[3];
            $productList[$i]['is_new'] = $row[4];
            $productList[$i]['in_stock'] = $row[5];
            $i++;
        }
        return $productList;
    }

    public static function
getProductsListByCategory($categoryId = false, $page = 1)
    {
        if ($categoryId) {
            $offset = ($page - 1) * self::IMAGES_COUNT;
            $db = Db::getConnection();
            $products = array();
            $result = $db->query("SELECT product_id, name,
price, image, is_new, in_stock
                                FROM product WHERE
status = '1'
                                AND category_id =
'$categoryId'
                                ORDER BY product_id DESC
LIMIT " . self::IMAGES_COUNT . " OFFSET " . $offset);
            $i = 0;
            while ($row = $result->fetch_row()) {
                $products[$i]['product_id'] = $row[0];
                $products[$i]['name'] = $row[1];
                $products[$i]['price'] = $row[2];
                $products[$i]['image'] = $row[3];
                $products[$i]['is_new'] = $row[4];
            }
        }
    }
}

```

```

        $products[$i]['in_stock'] = $row[5];
        $i++;
    }
    return $products;
}
}

public static function getProductsListByAuthor($authorId
= false)
{
    if ($authorId) {
        $db = Db::getConnection();
        $products = array();
        $result = $db->query("SELECT product.product_id,
product.name, product.price, product.image, product.is_new,
product.in_stock
                                FROM product inner join
author on product.author_id = author.author_id
                                WHERE status = '1'
                                AND product.author_id =
'$authorId'
                                ORDER BY product_id DESC
");
        $i = 0;
        while ($row = $result->fetch_row()) {
            $products[$i]['product_id'] = $row[0];
            $products[$i]['name'] = $row[1];
            $products[$i]['price'] = $row[2];
            $products[$i]['image'] = $row[3];
            $products[$i]['is_new'] = $row[4];
            $products[$i]['in_stock'] = $row[5];
            $i++;
        }
        return $products;
    }
}

public static function getProductById($id)
{
    if ($id) {
        $db = Db::getConnection();
        $result = $db->query("SELECT * FROM product
WHERE product_id = " . $id);
        return $result->fetch_assoc();
    }
}

public static function
getTotalProductsInCategory($categoryId)
{
    $db = Db::getConnection();

```

```

        $result = $dbh->query("SELECT count(product_id) AS
count FROM product WHERE status = '1' AND category_id = " .
$categoryId);
        $row = $result->fetch_assoc();
        return $row['count'];
    }

    public static function getProductsByIds($idsArray)
    {
        $products = array();
        $db = Db::getConnection();
        $idsString = implode(', ', $idsArray);
        $sql = "SELECT * FROM product WHERE status = '1' AND
product_id IN ($idsString)";
        $result = $dbh->query($sql);
        $result->setFetchMode(PDO::FETCH_ASSOC);
        $i = 0;
        while ($row = $result->fetch()) {
            $products[$i]['product_id'] =
$row['product_id'];
            $products[$i]['name'] = $row['name'];
            $products[$i]['price'] = $row['price'];
            $i++;
        }
        return $products;
    }

    public static function productSearch($search)
    {
        $products = array();
        $db = Db::getConnection();
        $result = $db->query("SELECT product.product_id,
product.name, product.price, product.image, product.is_new,
product.in_stock
FROM product
INNER JOIN author ON product.author_id = author.author_id
WHERE product.name like '%$search%'

OR product.product_id like '%$search%'

OR author.name like '%$search%'
ORDER BY product.product_id DESC");
        $i = 0;
        while ($row = $result->fetch_row()) {
            $products[$i]['product_id'] = $row[0];
            $products[$i]['name'] = $row[1];
            $products[$i]['price'] = $row[2];
            $products[$i]['image'] = $row[3];
            $products[$i]['is_new'] = $row[4];
            $products[$i]['in_stock'] = $row[5];
            $i++;
        }
    }

```

```

        return $products;
    }

    public static function getProductsList()
    {
        $db = Db::getConnection();

        $result = $db->query('SELECT product_id, name, price
FROM product ORDER BY product_id ASC');
        $productsList = array();
        $i = 0;
        while ($row = $result->fetch()) {
            $productsList[$i]['product_id'] =
FROM product ORDER BY product_id ASC');
            $productsList[$i]['name'] = $row['name'];
            $productsList[$i]['price'] = $row['price'];
            $i++;
        }
        return $productsList;
    }

    public static function deleteProductById($id)
    {
        $db = Db::getConnection();

        $sql = 'DELETE FROM product WHERE product_id = :id';

        $result = $db->prepare($sql);
        $result->bindParam(':id', $id, PDO::PARAM_INT);
        return $result->execute();
    }

    public static function createProduct($options)
    {
        $db = Db::getConnection();

        $sql = 'INSERT INTO product '
            . '(product_id, name, author_id, category_id,
price, availability, brand, '
            . 'image, description, is_new, is_recommended,
status, in_stock)'
            . 'VALUES '
            . '(:product_id, :name, :author_id,
:category_id, :price, :availability, :brand, '
            . ':image, :description, :is_new,
:is_recommended, :status, :in_stock)';

        $result = $db->prepare($sql);
        $result->bindParam(':product_id',
FROM product ORDER BY product_id ASC');
        $options['product_id'], PDO::PARAM_INT);
    }

```

```

        $result->bindParam(':name', $options['name'],
PDO::PARAM_STR);
        $result->bindParam(':author_id',
$options['author_id'], PDO::PARAM_INT);
        $result->bindParam(':category_id',
$options['category_id'], PDO::PARAM_INT);
        $result->bindParam(':price', $options['price'],
PDO::PARAM_STR);
        $result->bindParam(':availability',
$options['availability'], PDO::PARAM_INT);
        $result->bindParam(':brand', $options['brand'],
PDO::PARAM_STR);
        $result->bindParam(':image', $options['image'],
PDO::PARAM_STR);
        $result->bindParam(':description',
$options['description'], PDO::PARAM_STR);
        $result->bindParam(':is_new', $options['is_new'],
PDO::PARAM_INT);
        $result->bindParam(':is_recommended',
$options['is_recommended'], PDO::PARAM_INT);
        $result->bindParam(':status', $options['status'],
PDO::PARAM_INT);
        $result->bindParam(':in_stock',
$options['in_stock'], PDO::PARAM_INT);
        if ($result->execute()) {
            return $db->lastInsertId();
        }
        return 0;
    }
    public static function updateProductById($id, $options)
    {
        $db = Db::getConnection();

        $sql = "UPDATE product
        SET
            product_id = :product_id,
            name = :name,
            author_id = :author_id,
            category_id = :category_id,
            price = :price,
            availability = :availability,
            brand = :brand,
            image = :image,
            description = :description,
            is_new = :is_new,
            is_recommended = :is_recommended,
            status = :status,
            in_stock = :in_stock
        WHERE product_id = :id";

        $result = $db->prepare($sql);
        $result->bindParam(':id', $id, PDO::PARAM_INT);
    }
}

```



```

        $result->bindParam(':product_id',
$options['product_id'], PDO::PARAM_INT);
        $result->bindParam(':name', $options['name'],
PDO::PARAM_STR);
        $result->bindParam(':author_id',
$options['author_id'], PDO::PARAM_INT);
        $result->bindParam(':category_id',
$options['category_id'], PDO::PARAM_INT);
        $result->bindParam(':price', $options['price'],
PDO::PARAM_STR);
        $result->bindParam(':availability',
$options['availability'], PDO::PARAM_INT);
        $result->bindParam(':brand', $options['brand'],
PDO::PARAM_STR);
        $result->bindParam(':image', $options['image'],
PDO::PARAM_STR);
        $result->bindParam(':description',
$options['description'], PDO::PARAM_STR);
        $result->bindParam(':is_new', $options['is_new'],
PDO::PARAM_INT);
        $result->bindParam(':is_recommended',
$options['is_recommended'], PDO::PARAM_INT);
        $result->bindParam(':status', $options['status'],
PDO::PARAM_INT);
        $result->bindParam(':in_stock',
$options['in_stock'], PDO::PARAM_INT);
        return $result->execute();
    }

    public static function getProductOrder($id)
    {
        $db = Db::getConnection();
        $sql= "SELECT product.product_id, product.name,
product.price FROM product inner join product_order on
product.product_id = product_order.product_id WHERE
product_order.id = :id";
        $result = $db->prepare($sql);
        $result->bindParam(':id', $id, PDO::PARAM_INT);
        $result->setFetchMode(PDO::FETCH_ASSOC);
        $result->execute();
        return $result->fetchAll();
    }

    public static function getProductForCabinet($user_id)
    {
        $db = Db::getConnection();
        $sql= "SELECT product.product_id, product.name,
product.price, product_order.quantity FROM product inner join
product_order on product.product_id = product_order.product_id
WHERE product_order.user_id = :id";
        $result = $db->prepare($sql);
        $result->bindParam(':id', $user_id, PDO::PARAM_INT);

```

```

        $result->setFetchMode(PDO::FETCH_ASSOC);
        $result->execute();
        return $result->fetchAll();
    }
    public static function getRecommendedItems($user_id)
    {
        $db = Db::getConnection();
        $sql= "SELECT product.product_id, product.name,
product.price, product.image, product.in_stock, product.is_new
FROM product
    inner join category on product.category_id =
category.category_id
    WHERE product.product_id not in (select product_id from
product_order where user_id = :id) and category.category_id in
    (select product.category_id from product inner join
category on product.category_id = category.category_id
    inner join product_order on product.product_id =
product_order.product_id where product_order.user_id = :id)
LIMIT 4";

        $result = $db->prepare($sql);
        $result->bindParam(':id', $user_id, PDO::PARAM_INT);
        $result->setFetchMode(PDO::FETCH_ASSOC);
        $result->execute();
        return $result->fetchAll();
    }
}

```

Додаток В

Модель «Author»:

```

<?php
class Author
{
    public static function getAuthorsList()
    {
        $db = Db::getConnection();
        $authorsList = array();
        $result = $db->query('select author, author.name as
name, count from (select author.author_id as author, count(*) as
count from product
inner join author on product.author_id = author.author_id group
by author limit 7) temp inner join author on author.author_id =
temp.author order by count desc');
        $i = 0;
        while ($row = $result->fetch_row()) {
            $authorsList[$i]['author'] = $row[0];
            $authorsList[$i]['name'] = $row[1];
            $authorsList[$i]['count'] = $row[2];
            $i++;
        }
        return $authorsList;
    }

    public static function getAuthorsListAdmin()
    {
        $db = Db::getConnection();
        $result = $db->query('SELECT author_id, name FROM
author');

        $authorList = array();
        $i = 0;
        while ($row = $result->fetch()) {
            $authorList[$i]['author_id'] = $row['author_id'];
            $authorList[$i]['name'] = $row['name'];
            $i++;
        }
        return $authorList;
    }

    public static function createAuthor($name)
    {
        $db = Db::getConnection();
        $sql = 'INSERT INTO author (name) '
            . 'VALUES (:name)';
    }
}

```

```

        $result = $db->prepare($sql);
        $result->bindParam(':name', $name, PDO::PARAM_STR);
        return $result->execute();
    }

    public static function updateAuthorById($id, $name)
    {
        $db = Db::getConnection();

        $sql = "UPDATE author
                SET
                    name = :name
                WHERE author_id = :id";

        $result = $db->prepare($sql);
        $result->bindParam(':id', $id, PDO::PARAM_INT);
        $result->bindParam(':name', $name, PDO::PARAM_STR);
        return $result->execute();
    }

    public static function getAuthorById($id)
    {
        $db = Db::getConnection();

        $sql = 'SELECT * FROM author WHERE author_id = :id';

        $result = $db->prepare($sql);
        $result->bindParam(':id', $id, PDO::PARAM_INT);
        $result->setFetchMode(PDO::FETCH_ASSOC);
        $result->execute();
        return $result->fetch();
    }

    public static function deleteAuthorById($id)
    {
        $db = Db::getConnection();

        $sql = 'DELETE FROM author WHERE author_id = :id';

        $result = $db->prepare($sql);
        $result->bindParam(':id', $id, PDO::PARAM_INT);
        return $result->execute();
    }
}

```

Додаток Г

Функція пошуку на сайті:

```

<script>
$(document).ready(function () {
    let sectionBooks = $('.section__books');

    $('#s_i').on('keyup', function (event) {
        $.post("/search", {search_input:
event.target.value}, function (data) {
            sectionBooks.empty();

            console.log($(".productinfo"));
            $.each(JSON.parse(data), function (index,
{image, price, product_id, name, in_stock, is_new}) {
                let colSm = $('<div/>').addClass('col-sm-
4');

                let imageWrapper =
$('<div/>').addClass('product-image-wrapper');
                let singleProduct =
$('<div/>').addClass('single-products');
                let productInfo =
$('<div/>').addClass('productinfo text-center');
                let imageTag = $('<img/>')
                    .attr('src',
`/shop/template/images/product-details/${image}`)
                    .attr('style', 'height: 360px;');
                let imageNew;
                if(parseInt(is_new)){
                    imageNew = $('<img/>')
                        .attr('src',
'/shop/template/images/home/new.png')
                        .addClass('new');
                }
                let priceTag = $('<h2/>').text(`${price}
€`);

                let titleTag = $('<a>').addClass('product-
name__link')
                    .attr('href', `/product/${product_id}`)
                    .text(name);
                let cartElement = $('<a>').addClass('btn
btn-default add-to-cart')
                    .attr('href', '#').attr('data-id',
product_id)
                    .text("В корзину");
                if (parseInt(in_stock) === 0) {
                    cartElement.attr("disabled", "disabled")

```

```

    }
    cartElement.click(function () {
        var id = $(this).attr("data-id");
        $.post("/cart/addAjax/" + id, {},
function (data) {
            $("#cart-count").html(data);
        });
        return false;
    });

    productInfo.append(imageTag);
    singleProduct.append(productInfo);
    imageWrapper.append(singleProduct);
    colSm.append(imageWrapper);
    sectionBooks.append(colSm);
    productInfo.append(priceTag);
    productInfo.append(titleTag);
    productInfo.append(cartElement);
    singleProduct.append(imageNew);
});
});
});
});
</script>

```