

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет

Факультет електроніки та інформаційних технологій

Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

(підпис)

10 червня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавр

зі спеціальності 122 - Комп'ютерних наук,
освітньо-професійної програми «Інформатика»
на тему: «Інформаційне та програмне забезпечення веб-орієнтованої платформи
книгарні»
здобувача групи ІН-94-1 Василевської Анастасії Романівни

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело.

Анастасія ВАСИЛЕВСЬКА

(підпис)

Керівник,

доцент,

кандидат технічних наук

Ігор ШЕЛЕХОВ

(підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

(підпис)

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавра

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»
здобувача групи ІН-94-1 Василевської Анастасії Романівни

1. Тема роботи: « Інформаційне та програмне забезпечення веб-орієнтованої платформи книгарні» затверджую наказом по СумДУ від _____
2. Термін здачі здобувачем кваліфікаційної роботи до 09 червня 2023 року
3. Вхідні дані до кваліфікаційної роботи _____
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз проблеми предметної області, постановка й формування завдань дослідження.
2) Огляд технологій, що використовуються у інформаційних системах веб-орієнтованих платформ. *3) Розробка інформаційного та програмного забезпечення веб-орієнтованої платформи книгарні.* *4) Аналіз результатів.*
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання « ____ » _____ 20 ____ р.

Завдання прийняв до виконання _____

(підпис)

Керівник _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, постановка й формування завдань дослідження</i>		
2	<i>Огляд технологій, що використовуються у інформаційних системах веб-орієнтованих платформ.</i>		
3	<i>Розробка інформаційного та програмного забезпечення веб-орієнтованої платформи книгарні.</i>		
4	<i>Аналіз отриманих результатів</i>		
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>		

Здобувач вищої освіти _____

(підпис)

Керівник _____

(підпис)

АНОТАЦІЯ

Записка: 59 ст., 50 рис., 2 табл., 20 використаних джерел.

Обґрунтування актуальності теми роботи – за останні роки спостерігається стрімкий розвиток електронної комерції, включаючи онлайн-торгівлю книгами. Веб-орієнтовані платформи книгарні є важливим інструментом для виробників, продавців та покупців книг, що сприяє ефективній обміну та продажу літератури, а також дозволяє авторам просувати свої книжки та залучати нових читачів. Можливість переглядати широкий асортимент книг, шукати за авторами, жанрами, видавництвами та іншими критеріями, здійснювати покупки у будь-який зручний для час, без необхідності виходити з дому. Додаток може надавати можливість збирати дані про вподобання та покупки споживачів. Ця інформація може бути використана для аналізу та прогнозування попиту, а також для персоналізованої реклами та рекомендацій.

Об'єкт дослідження – інформаційне та програмне забезпечення веб-орієнтованої платформи книгарні.

Мета роботи – реалізація інформаційного та програмного забезпечення веб-орієнтованої платформи книгарні.

Методи дослідження – аналіз існуючих рішень, технології створення інтернет-каталогів.

Результати – створено архітектуру системи та базу даних, дизайн інтерфейсу користувача, реалізовано програмний код, який містить вимоги до функціональності платформи, протестовано різними видами тестування, а саме: «ручне» тестування та з використанням засобів для автоматичного тестування.

ВЕБ-ОРІЄНТОВАНА ПЛАТФОРМА КНИГАРНІ, HTML, CSS, PHP,
JAVASCRIPT, JQUERY, AJAX, MYSQL, PYTHON, SELENIUM

ЗМІСТ

ВСТУП	5
1 АНАЛІТИЧНИЙ ОГЛЯД	6
1.1 Сучасні онлайн книгарні	6
1.2 Постановка задачі.....	14
2 ПРОЄКТУВАННЯ СИСТЕМИ.....	15
2.1 Технології проєктування веб-орієнтованих систем.....	15
2.2 Інформаційна модель системи	15
2.3 Структура бази даних	16
3 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	20
3.1 Вибір засобів для програмної реалізації	20
3.2 Розробка системи	24
3.3 Тестування веб-застосунку	37
ВИСНОВКИ.....	40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	41
ДОДАТКИ.....	43
Додаток А. Програмний код адміністративної панелі	43
Додаток Б. Програмний код сторінки «Каталог».....	50
Додаток В. Програмний код обробника події кнопки «вгору»	56
Додаток Г. Тестування веб-застосунку	57

ВСТУП

Електронно-обчислювальна техніка призвела до перебудови технічних основ виробництва. Інформаційні технології з'явилися на основі цієї техніки під час четвертої науково-технічної революції, що охопила інтелектуальну діяльність. Інформація в цих процесах є «вихідним матеріалом» і «продукцією». Як виробничі, так і інформаційні технології виникають в результаті модернізації певних процесів, тобто за допомогою впливу людини або групи людей на галузі виробництва і трансформації їх на базі машинної техніки.

Інформаційні технології – цілеспрямована організована сукупність інформаційних процесів з використанням засобів обчислювальної техніки, що забезпечують високу швидкість обробки даних, швидкий пошук інформації, розосередження даних, доступ до джерел інформації незалежно від місця їх розташування [1].

Інтернет-каталоги, які полегшують життя, з'явилися завдяки інформаційним технологіям. За допомогою мережі Інтернет та спеціальних сайтів можна відразу побачити інформацію про ціну, знижки та наявність товару.

Особливо це актуально для великих та середніх підприємств, для культурної та навчальної сфери, а саме, книгарні.

Книжковий онлайн каталог – сайт, де можна обрати потрібний товар спираючись на опис та ціну.

Важливими елементами інтернет-каталогу є: оновлення наявного асортименту (товари, можливо їх кількість), актуальність ціни наявного асортименту товарів, реєстрація та вхід для зареєстрованих користувачів.

1 АНАЛІТИЧНИЙ ОГЛЯД

1.1 Сучасні онлайн книгарні

Зараз, в еру інформаційних технологій, зручно економити час та переглядати товари в інтернет-магазинах чи інтернет-каталогах.

Серед основних переваг інтернет-каталогу визначають такі:

- Мінімальні постійні витрати. Оплата хостингу і доменного імені.
- Цілодобовий доступ до інтернет-каталогу. В даний момент в мережі Інтернет з'являється все більше інтернет-магазинів та інтернет-каталогів, але перенасичення ринку ще не спостерігається.

- Фільтрація даних може спростити пошук необхідної інформації, тому сайт надає можливість сортування даних за допомогою функції фільтрації.

Найбільш поширеними фільтрами є: сортування за ціною, сортування за роком випуску, фільтрація за авторами.

Процес виведення записів із списку спрощено фільтрами. При фільтрації, записи, які не відповідають зазначеним критеріям, приховуються, але не вилучаються з бази [2].

Незважаючи на велику кількість інтернет-каталогів, цільовій аудиторії зараз легше поїхати до оф-лайн магазину подивитися та одразу придбати потрібне. Наразі, проблема полягає в збільшенні термінів доставки та страху оплати наперед, оскільки через військове вторгнення на територію нашої держави до більшості населених пунктів доставка відсутня або має загрозливий характер. Незважаючи на це, є компанії які відновили роботу в межах міст або з доставкою на деокуповані регіони.

Першим аналогом інтернет-каталогу книгарні є «bookinstein», інтернет-магазин сумського дистриб'ютора [3]. Даний інтернет-магазин пропонує два способи доставки по Україні. На період війни магазин зачинено до поліпшення безпекової ситуації, про що свідчить банер в заголовку сторінки. У «bookinstein» різноманітний асортимент, наявна можливість перекладу українською та російською мовами. Є форма зворотного зв'язку (рис. 1.1, 1.2).

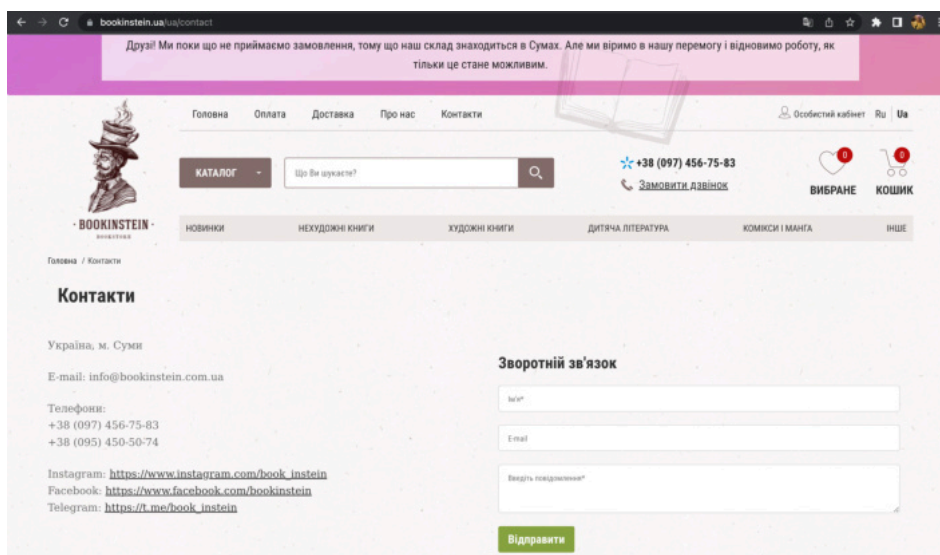


Рисунок 1.1 – Зворотній зв'язок bookinstitute

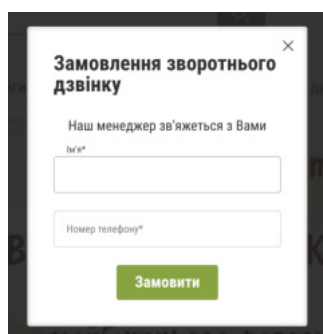


Рисунок 1.2 – Зворотній дзвінок bookinstitute

Головна сторінка зустрічає зручними іконками для навігації по сайту та інтуїтивно зрозумілим інтерфейсом (рис. 1.3, 1.4).

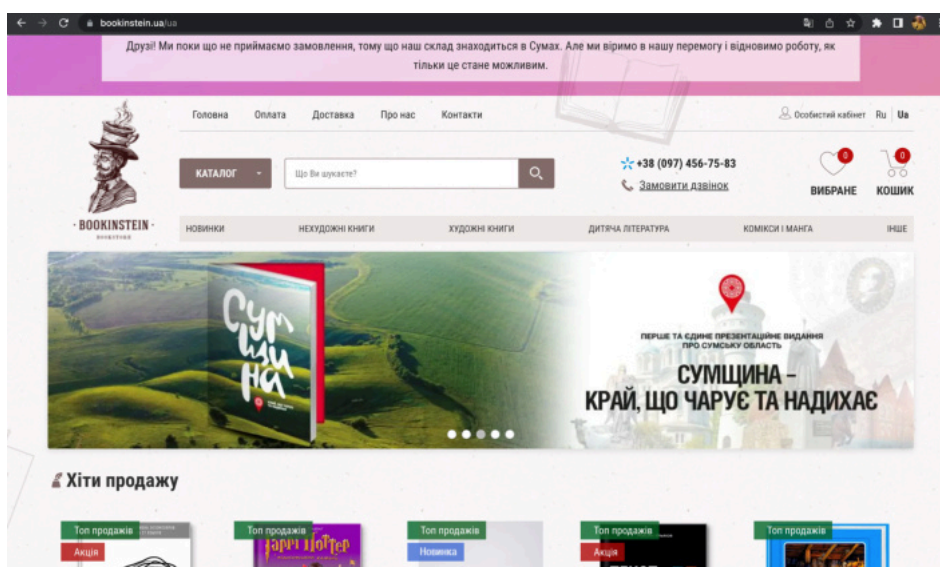


Рисунок 1.3 – Початкова сторінка bookinstitute

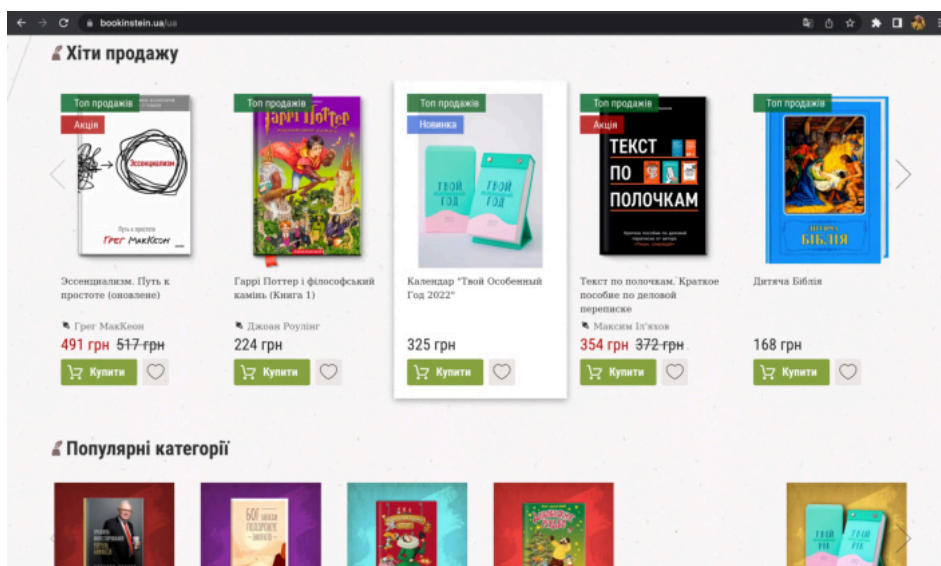


Рисунок 1.4 – Продовження початкової сторінки bookinsein

Авторизуватися на даному сайті можна лише за наявністю пошти та номеру телефону, форма реєстрації містить в собі лише важливі пункти, що пришвидшує даний процес для кінцевого користувача (рис. 1.5).

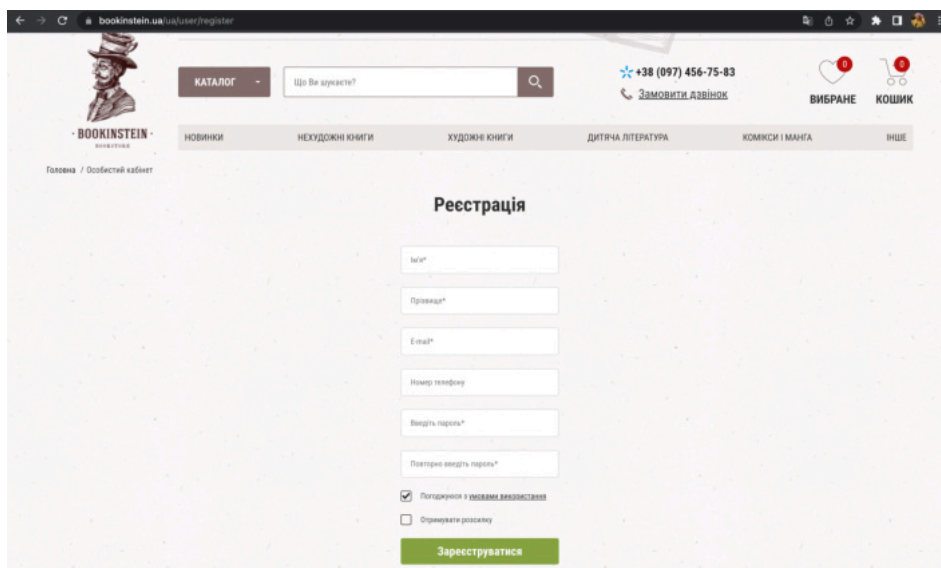


Рисунок 1.5 – Реєстраційна форма bookinsein

У каталозі є багато фільтрів, що полегшують пошук інформації. Є можливість сортування за ціною, назвою або рейтингом. При наведенні мишею обраний товар підсвічується. Наявна зручна пагінація сторінок. Незначним недоліком можна вважати відсутність можливості вибору кількості товарів що відображаються на одній сторінці (рис. 1.6).

The image shows a screenshot of the BookinStein website's catalog page. The page features a top navigation bar with links for 'Головна', 'Оплата', 'Доставка', 'Про нас', and 'Контакти'. A search bar is located in the top right, along with a phone number '+38 (097) 456-75-83' and a 'Замовити дзвінок' button. The main header includes the BookinStein logo and navigation tabs for 'НОВИНИ', 'НЕХУДОЖНІ КНИГИ', 'ХУДОЖНІ КНИГИ', 'ДИТЯЧА ЛІТЕРАТУРА', 'КОМІКСИ І МАНГА', and 'ІНШЕ'. The left sidebar contains various filters: 'Ціна' (Price) with a range from 0.00 to 27800.00, 'Зі знижкою' (With discount), 'Хіти продажів' (Bestsellers), 'Виробник' (Publisher) with options like 'Всі', 'Наш Формат', 'Форс Україна', 'Альпіна Паблішер', and 'Добра книга', 'Дивитись все', 'Автори' (Authors) with a search box and options like 'Всі', 'Сам Харрис', 'Галкін', 'LEBO', 'Дивитись все', 'Мова видання' (Language of publication) with options like 'Всі', 'Російська', 'Українська', 'Німецька', 'Українська / Німецька', 'Дивитись все', 'Серія' (Series) with options like 'Всі', 'I Love English. Моя перша бібліотека англійською', 'Легкий спосіб' Адлена Карра, 'Бізнес', '100 головних книг', 'Дивитись все', 'Палітурка' (Binding) with options like 'Всі', 'Шкіряна палітурка', 'М'яка обкладинка', 'Тверда палітурка', 'Обкладинка з клапаном', 'Дивитись все', and 'Папір' (Paper) with options like 'Всі', 'Офсетний', 'Газетний'. The main content area displays a grid of 12 book covers with their titles, authors, and prices. The books shown are: 'ВИБОР' (382 грн), 'ПРАВИЛА ЖИТТЯ' (185 грн), 'ПРАВИЛА МИСЛЕННЯ' (185 грн), 'ПРАВИЛА КАР'ЄРИ' (185 грн), 'ГОДИНИ' (105 грн), 'ДИВАНКА, ЩО ЗАКОХАЛАСЯ В МОЗОК' (320 грн), 'ХРАНИТЕЛІ СМЕРТІ' (175 грн), 'ФУНДАЦІЯ ТА ЗЕМЛЯ' (205 грн), 'ЙДИ ГЕТЬ ЗІ СВОГО ШЛЯХУ!' (185 грн), 'ЗРОБИ ЦЕ ЗАРАЗ! 22 ПОРАДКІ ДЛЯ УСПІШНОГО НАВЧАННЯ' (180 грн), 'ЗОВНІШНІ ІСТОРІЇ. РЕНІС. Керівництво з експлуатації' (200 грн), 'Кайдзен. Японський метод трансформації привичок мільярдних шагланів' (295 грн), 'Опра Вінфрі. Видатні особистості для дітей (тверда палітурка)' (255 грн), 'ЛІДАР СІМОХ КОРОЛІВСТВА' (285 грн), 'Екзистенціалізм. Путь к простоте (оновление)' (491 грн / 617 грн), and 'Нікола Тесла. Видатні особистості для дітей (тверда палітурка)' (255 грн). At the bottom, there is a pagination bar showing '1 2 3 4 5 ... 302 >>'.

Рисунок 1.6 – Каталог товарів bookinStein

Недоліком є недостатня адаптивність сайту. На рисунку 1.7 можна побачити сайт в стандартному відображенні.

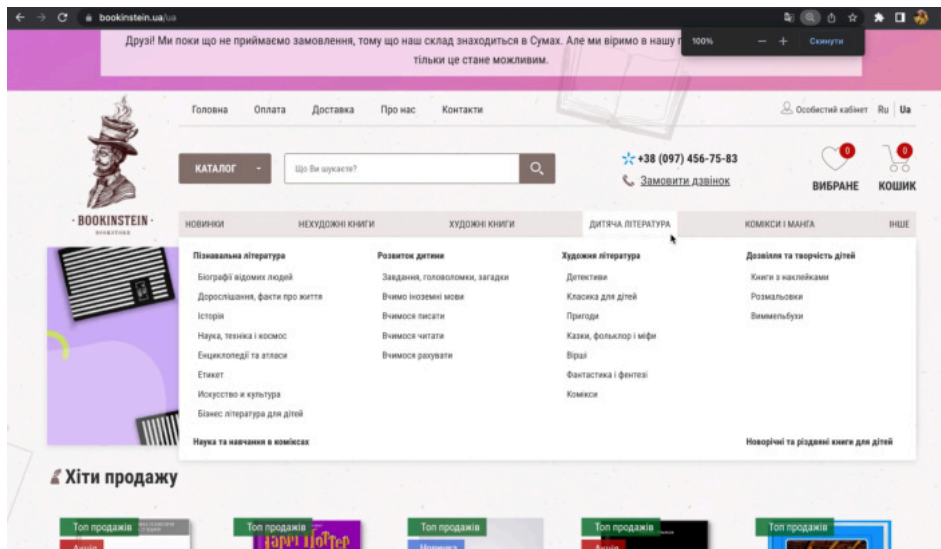


Рисунок 1.7 – Стандартне відображення bookinstitute

При збільшенні масштабу блоки часто відображені некоректно, випадваюче меню не доступне (рис. 1.8).

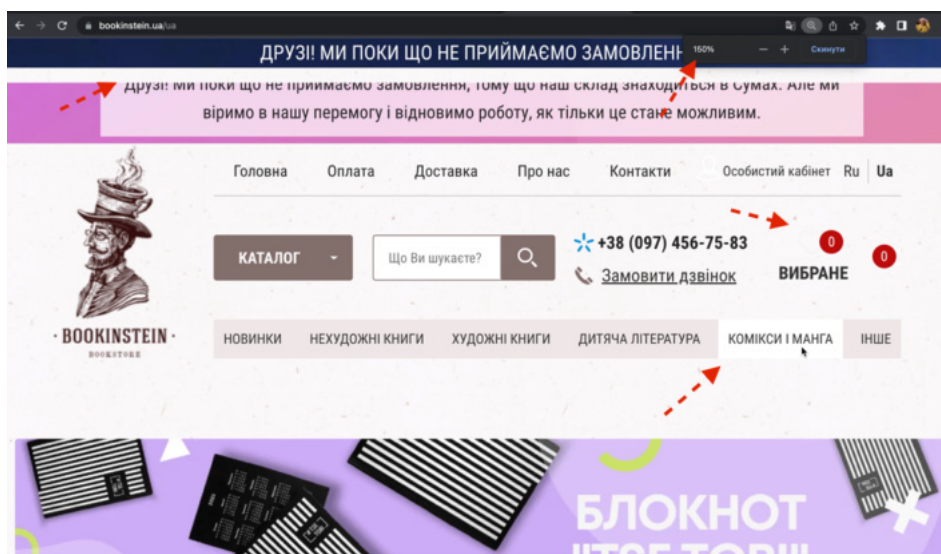


Рисунок 1.8 – Адаптивність bookinstitute

Наступний аналог інтернет-каталогів – «укабоо». В інформації про себе вказують що на ринку з 2009 року, співпрацюють з понад 500 українськими та закордонними видавництвами, їх асортимент багатий на переклади понад 300 тисяч книжок 71 мовою у паперовому, електронному або аудіо форматах [4].

Головна сторінка зустрічає зручним та ємким меню для навігації. Центрує увагу інформаційний банер з актуальною інформацією про відновлення роботи, міжнародну доставку, знижки та інше (рис. 1.9).

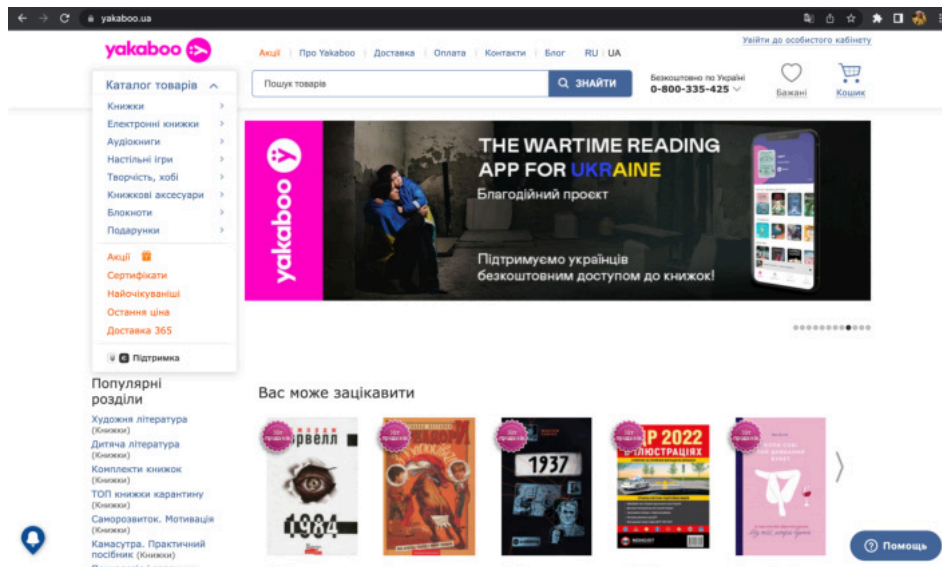


Рисунок 1.9 – Початкова сторінка yakaboo

Платформа передбачає реєстрацію користувача. Наявна можливість авторизації за допомогою соціальних мереж, це позитивно впливає на кількість користувачів, оскільки витрачається менше часу на авторизацію (рис. 1.10).

Рисунок 1.10 – Реєстрація yakaboo

На рисунку 1.11 зображено форму зворотного зв'язку.

Рисунок 1.11 – Замовити дзвінок yakaboo

Переглядаючи каталог можна побачити наліпку «Хіт продажів» на деяких товарах, це зручно та допомагає користувачам у виборі книги (рис. 1.12).

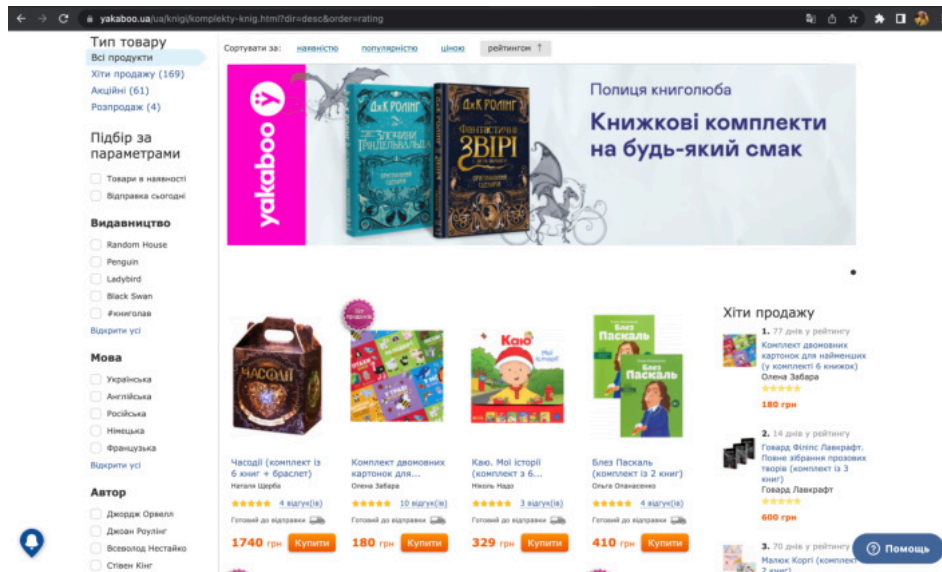


Рисунок 1.12 – Каталог yakaboo

У кінці сторінки наявна пагінація, крім того можна відобразити товари без переходу на іншу сторінку. В окремих випадках це зручно, адже не потрібно запам'ятовувати сторінку з книгою що зацікавила, вистачить виконати пошук на сторінці (рис. 1.13).



Рисунок 1.13 – Пагінація yakaboo

Книгарня містить свій блог, це позитивно впливає на рекламу компанії та зацікавленість читачів (рис 1.14).

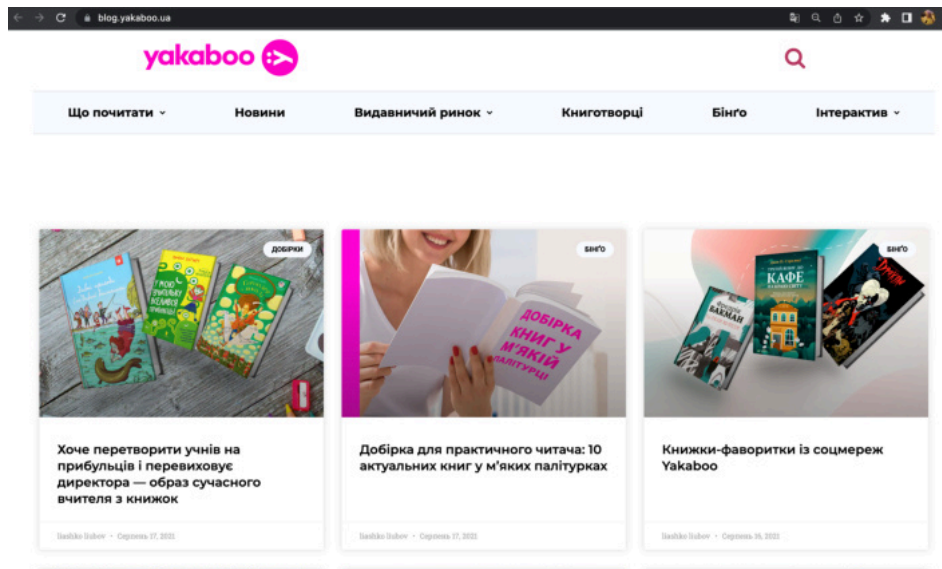


Рисунок 1.14 – Блог yakaboo

Аналоги об'єднує різноманітність брендів, майже єдина цінова політика. Під час пошуку аналогів, не було виявлено саме інтернет-каталогів. У всіх присутня форма онлайн-замовлень.

Згідно аналізу аналогів, була сформована порівняльна таблиця розглянутих прикладів та майбутнього інтернет-каталогу. Результати аналізу представлено у таблиці 1.1.

Таблиця 1.1 – Порівняльна характеристика аналогів

Назва критерію	Назва ресурсу		
	bookinstein	yakaboo	(майбутній додаток)
1. Використання ресурсу без реєстрації	+	+	+
2. Зручність навігації	+	+	+
3. Належність до одного бренду	-	-	-
4. Адаптивність сайту	-	+	+
5. Форма зворотного зв'язку	+	+	+
6. Сортування	+	+	+
7. Фільтрація	+	+	+

Результатом проведеного аналізу сайтів зі схожою тематикою було вирішено розробити вебдодаток який не матиме недоліків виявлених на сайтах представлених в таблиці 1.1.

Дана розробка матиме всі переваги даних сайтів.

1.2 Постановка задачі

Метою розробки сайту інтернет-каталогу є моніторинг наявності та актуальної ціни на видавничу продукцію книгарні, замовлення товарів. Основним завданням – створення комфортного середовища для перегляду інформації про товар та підприємство. Також, створити адміністративну панель, щоб було зручно керувати вмістом каталогу. Актуальність теми полягає в тому, що сучасне суспільство неможливо уявити без веб-ресурсів, що спрощують та пришвидшують пошук інформації.

Для досягнення зазначеної мети потрібно дотримуватися таких етапів:

- 1) Аналіз вимог: визначення визначення технології проектування веб-орієнтованої системи, потреб і вимог, що стосуються платформи книгарні, таких як функціональність, інтерфейс користувача, тощо;
- 2) Проектування: розробка архітектури системи, дизайну інтерфейсу користувача та бази даних, яка буде використовуватися для зберігання інформації про книги, замовлення, користувачів;
- 3) Розробка програмного забезпечення: реалізація програмного коду, який містить вимоги до функціональності платформи;
- 4) Розробка адміністративної панелі: створення окремого інтерфейсу, доступного адміністраторам книгарні, для керування вмістом каталогу, оновлення цін, додавання та редагування товарів;
- 5) Тестування: використання різних видів тестування, а саме: «ручне» тестування та з використанням засобів для автоматичного тестування.

2 ПРОЄКТУВАННЯ СИСТЕМИ

2.1 Технології проєктування веб-орієнтованих систем

Більшість вебдодатків базуються на архітектурі клієнт-сервер, куди клієнт вводить інформацію, а сервер отримує та обробляє її. Для розробки зазвичай використовують саме цю комбінацію сценаріїв: на стороні сервера (PHP) та на стороні клієнта (HTML, Javascript).

Більшість популярних вебдодатків, наприклад Gmail, змінилися за ці роки, стали більш складними. Це витонченість у значній мірі пов'язана з AJAX. AJAX – це підхід, за яких вебсторінка, не перезавантажуючись, у фоновому режимі надсилає запити на сервер і довантажує потрібні користувачу дані [5]. Набір методів веброботи, які дозволяють вебдодаткам працювати асинхронно та обробляти будь-які запити до сервера у фоновому режимі. Простіше кажучи, модель програмування для створення більш чуйних вебдодатків.

Результатом використання AJAX-технологій є можливість відправляти та отримувати дані з сервера без необхідності перезавантаження всієї сторінки.

jQuery – швидка, невелика та багатофункціональна бібліотека JavaScript. Значно спрощує використання AJAX-технологій завдяки простому у використанні API, який працює в багатьох браузерах [6].

Спираючись на зазначені етапи розробки та необхідність присутності певних елементів, для вирішення задачі було обрано наступні технології:

- HTML та CSS;
- PHP;
- JavaScript, AJAX, jQuery;
- MySQL;
- Bootstrap.

2.2 Інформаційна модель системи

Фізична структура сайту являє собою динамічні PHP-сторінки оформлені CSS-стилями та JS-скриптами, до яких підключена база даних.

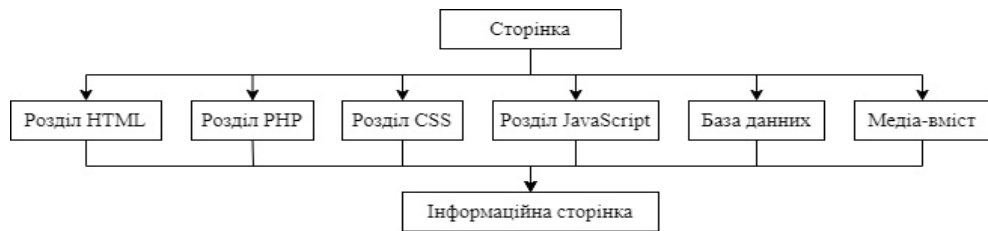


Рисунок 2.1 – Фізична структура сайту

Логічна структура сайту представляє собою взаємозв'язок сторінок. Саме на цьому етапі прослідковується наскільки користувачу буде зручно ним користуватися; скільки часу знадобиться для переходу між категоріями сторінок; чи не забагато, з точки зору користувача «пустих» сторінок, якими технологіями їх можна зменшити.

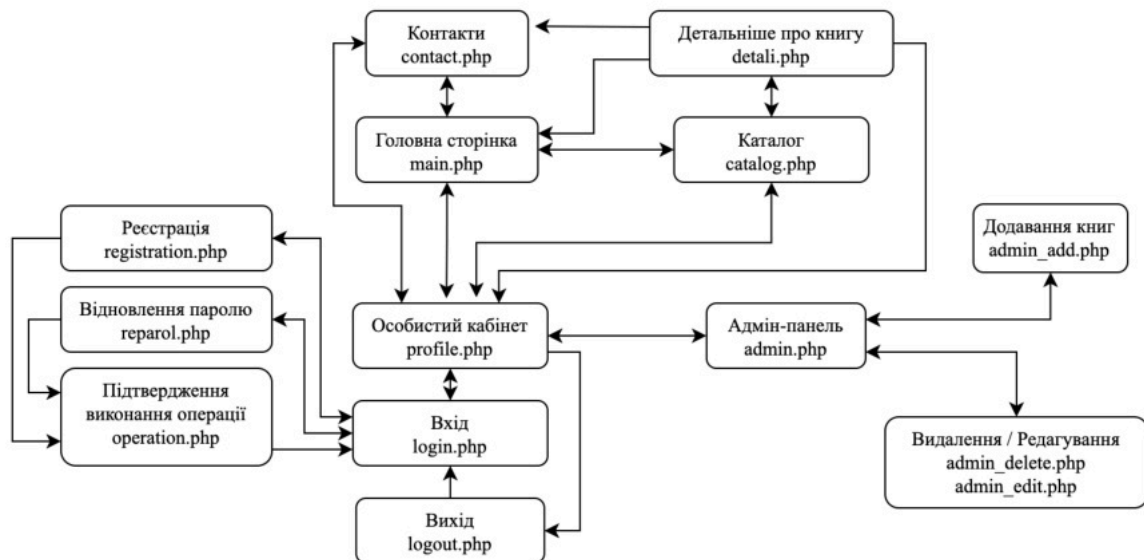


Рисунок 2.2 – Логічна структура сайту

2.3 Структура бази даних

Для управління базами даних використано веб-додаток phpMyAdmin. PHPMyAdmin – це сукупність скриптів написаних на PHP, що забезпечують повноцінну, в тому числі віддалену роботу з базами даних MySQL, через веб-інтерфейс [7].

Таблиці мають такі тип зв'язку один до багатьох. Для уникнення дублювання даних в таблицях використовується зв'язок один до багатьох.

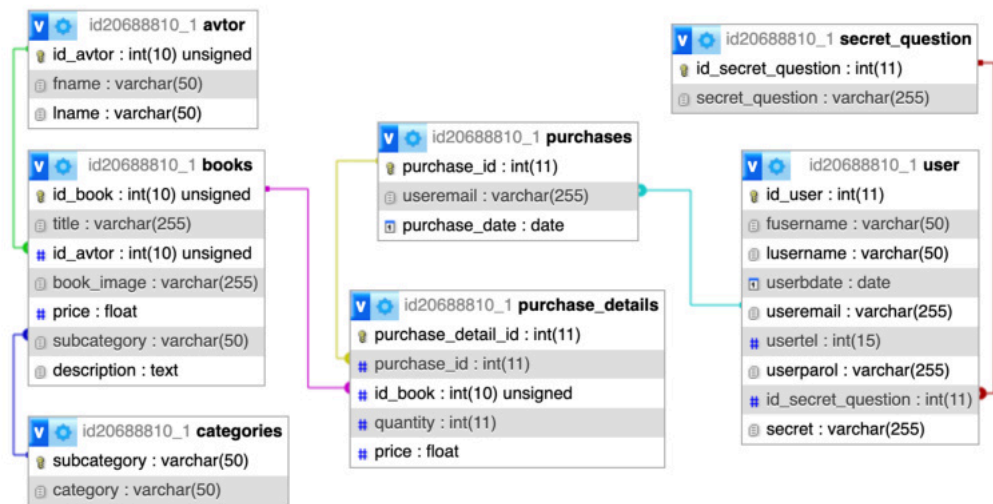


Рисунок 2.3 – Структура бази даних

Створення таблиці user, що містить інформацію про користувача: id, прізвище, ім'я, дату народження, email, контактний номер телефону, пароль, id секретного запитання, що використовується для відновлення паролю та відповідь на нього (рис. 2.4).

#	Ім'я	Тип	Порівняння	Атрибути	Null	За замовчуванням	Коментарі	Додатково
1	id_user	int(11)			Hi	Hi		AUTO_INCREMENT
2	fusername	varchar(50)	utf8_general_ci		Hi	Hi		
3	lusername	varchar(50)	utf8_general_ci		Hi	Hi		
4	userbdate	date			Hi	Hi		
5	useremail	varchar(255)	utf8_general_ci		Hi	Hi		
6	usertel	int(15)			Hi	Hi		
7	userparol	varchar(255)	utf8_general_ci		Hi	Hi		
8	id_secret_question	int(11)			Hi	Hi		
9	secret	varchar(255)	utf8_general_ci		Hi	Hi		

Рисунок 2.4 – Створення таблиці «user»

Створення таблиці categories, що містить назви категорій та підкатегорій товарів (рис. 2.5).

#	Ім'я	Тип	Порівняння	Атрибути	Null	За замовчуванням	Коментарі	Додатково
<input type="checkbox"/>	subcategory	varchar(50)	utf8_general_ci		Hi	Hi		
<input type="checkbox"/>	category	varchar(50)	utf8_general_ci		Hi	Hi		

Рисунок 2.5 – Створення таблиці «categories»

Створення таблиці books, що містить інформацію про асортимент книг, а саме: id, назву, автора, зображення, ціну, категорію та детальну інформацію (рис. 2.6).

#	Ім'я	Тип	Порівняння	Атрибути	Null	За замовчуванням	Коментарі	Додатково
1	id_book	int(10)		UNSIGNED	Hi	Hi		AUTO_INCREMENT
2	title	varchar(255)	utf8_general_ci		Hi	Hi		
3	id_avtor	int(10)		UNSIGNED	Hi	Hi		
4	book_image	varchar(255)	utf8_general_ci		Hi	Hi		
5	price	float			Hi	Hi		
6	subcategory	varchar(50)	utf8_general_ci		Hi	Hi		
7	description	text	utf8_general_ci		Hi	Hi		

Рисунок 2.6 – Створення таблиці «books»

Створення таблиці avtor що містить інформацію про авторів, а саме: id, прізвище та ім'я (рис. 2.7).

#	Ім'я	Тип	Порівняння	Атрибути	Null	За замовчуванням	Коментарі	Додатково
1	id_avtor	int(10)		UNSIGNED	Hi	Hi		AUTO_INCREMENT
2	fname	varchar(50)	utf8_general_ci		Hi	Hi		
3	lname	varchar(50)	utf8_general_ci		Hi	Hi		

Рисунок 2.7 – Створення таблиці «avtor»

Створення таблиці secret_question що містить інформацію про секретне запитання, а саме: id секретного запитання, що використовується для відновлення паролю та його зміст (рис. 2.8).

#	Ім'я	Тип	Порівняння	Атрибути	Null	За замовчуванням	Коментарі	Додатково
1	id_secret_question	int(11)			Hi	Hi		AUTO_INCREMENT
2	secret_question	varchar(255)	utf8_general_ci		Hi	Hi		

Рисунок 2.8 – Створення таблиці «secret_question»

Створення таблиці purchase що містить інформацію про замовлення, а саме: id, email користувача на яке створено замовлення та дата замовлення (рис. 2.9).

#	Ім'я	Тип	Порівняння	Атрибути	Null	За замовчуванням	Коментарі	Додатково
1	purchase_id	int(11)			Hi	Hi		AUTO_INCREMENT
2	useremail	varchar(255)	utf8_general_ci		Hi	Hi		
3	purchase_date	date			Так	NULL		

Рисунок 2.9 – Створення таблиці «purchase»

Створення таблиці `purchase_details` що містить інформацію про деталі замовлення, а саме: унікальний `id` деталей, `id` замовлення, `id` книги в замовленні їх кількість та ціна на момент замовлення (рис. 2.10).

#	Ім'я	Тип	Порівняння	Атрибути	Null	За замовчуванням	Коментарі	Додатково
1	<code>purchase_detail_id</code> 🔑	<code>int(11)</code>			Hi	Hi		AUTO_INCREMENT
2	<code>purchase_id</code> 🔑	<code>int(11)</code>			Так	NULL		
3	<code>id_book</code> 🔑	<code>int(10)</code>		UNSIGNED	Hi	Hi		
4	<code>quantity</code>	<code>int(11)</code>			Так	NULL		
5	<code>price</code>	<code>float</code>			Так	NULL		

Рисунок 2.10 – Створення таблиці «`purchase_details`»

Підключення до бази даних виконується за допомогою коду на мові програмування PHP, що знаходиться у файлі `BDconnect.php`.

Програмний код налаштування бази даних:

```
<?php
$host = 'localhost';
$user = 'id20688810_user';
$pass = 'Id20688810_1';
$db_name = 'id20688810_1';

$link = mysqli_connect($host, $user, $pass, $db_name);
if (!$link){echo 'Не можу з'єднатися з БД.';
    exit;}
mysqli_query($link, "SET NAMES 'utf8' COLLATE
'utf8_general_ci'");
mysqli_query($link, "SET CHARACTER SET 'utf8'");
?>
```

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Вибір засобів для програмної реалізації

На сьогодні, існує безліч як безкоштовних, так і платних додатків: від текстових редакторів до IDE (інтегрованих середовищ розробки). Оскільки зручність використання того чи іншого додатку річ суб'єктивна, важко відразу вирішити, яке середовище краще підходить для поставленого завдання.

Першим з розглянутих був додаток PhpStorm від компанії JetBrains – це комерційна IDE вона може похвалитися значним набором корисних функцій. Включає в себе: редактор HTML / CSS, редактор JavaScript і додає повноцінну підтримку PHP та баз даних, та на цьому перелік не закінчується [8].

PhpStorm надає багатий і інтелектуальний редактор коду з підсвіткою синтаксису, розширеною конфігурацією форматування коду, перевіркою на наявність помилок на льоту і розумним автодоповненням. Інтеграція з системами управління версіями (Git — включаючи спеціальний функціонал для роботи з GitHub) дозволяє робити багато дій, наприклад commit, merge та інші, прямо з PhpStorm.

Графічний відладчик не вимагає додаткового налаштування. Він дуже наочно візуалізує, що відбувається у вашому додатку на кожному етапі налагодження.

Студентська підписка PhpStorm надає можливість використовувати PhpStorm з повною функціональністю безкоштовно.

На рисунку 3.1 зображено інтерфейс програми PhpStorm.

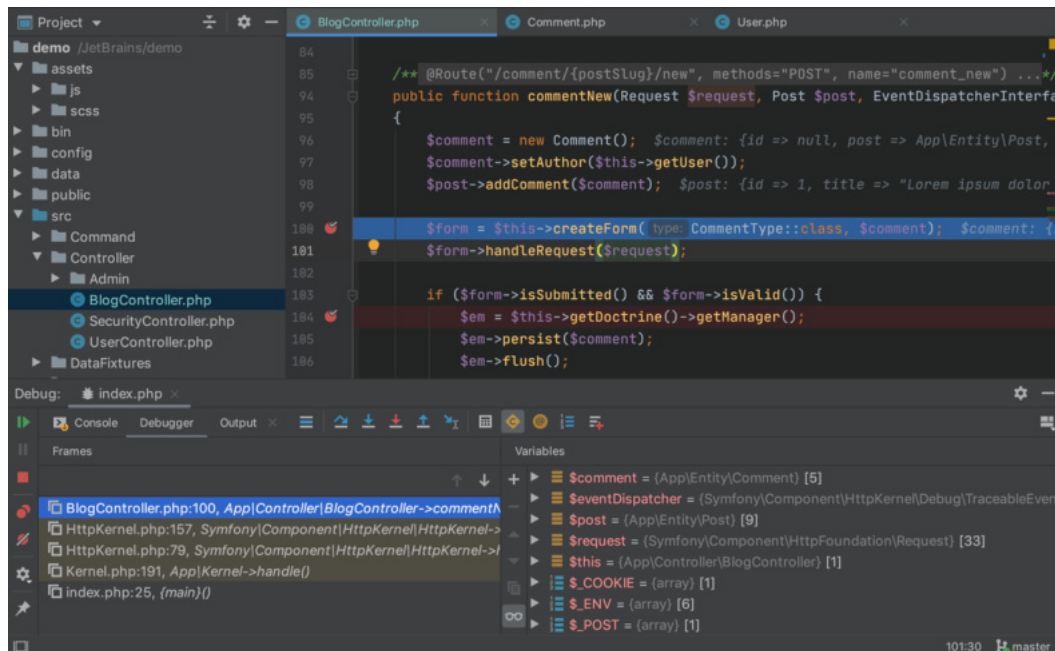


Рисунок 3.1 – Інтерфейс програми PhpStorm

Компанія співпрацює з освітніми платформами, такими як Stepik. Це дозволяє активним користувачам платформи безкоштовно отримати на достатньо довгий період ліцензійний ключ для активації продуктів JetBrains. Також є можливість подачі заявок на безкоштовні індивідуальні ліцензії для освіти. Вони доступні для студентів і викладачів з акредитованих навчальних закладів (шкіл, коледжів і вузів).

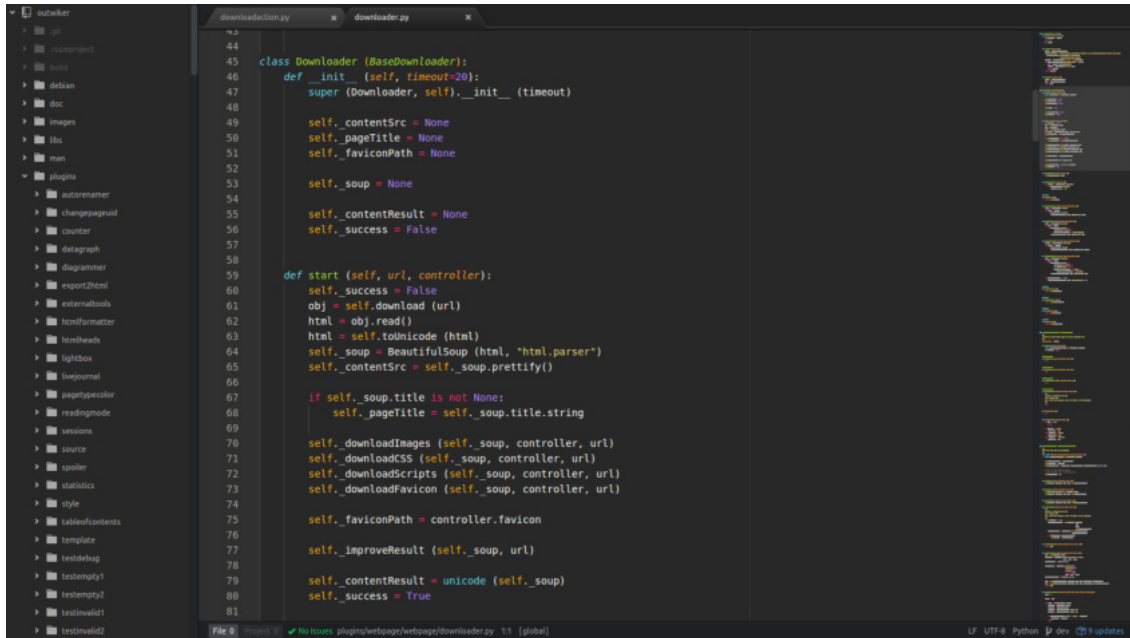
Наступним було розглянуто Atom створений командою GitHub – текстовий редактор з множиною налаштувань. Надає засоби редагування та одночасної спільної роботи з кодом, має інтелектуальну систему автодоповнення вводу [9].

Як і в PhpStorm, кілька файлів можуть бути відкриті в різних вкладках і одночасно відображені з використанням вертикального або горизонтального розбиття панелей. Інтерфейс може налаштовуватися через теми оформлення, підтримуються вкладки, закладки, розумний контекстний пошук коду, автодоповнення та перевірка коду.

Функціональність редактора формується за рахунок надання набору пакетів-доповнень. Через доповнення реалізовані всі функції, що виходять за

рамки базового редагування коду, в тому числі панелі, підсвічування синтаксису, оформлення інтерфейсу, форми роботи з файлами тощо.

На рисунку 3.2 зображено інтерфейс програми Atom.



Рисунк 3.2 – Інтерфейс програми Atom

Недоліком є те, що при редагуванні спостерігається повільне завантаження і неоднорідна продуктивність, особливо при роботі з великими файлами.

Останнім з розглянутих середовищ розробки є Notepad++. Безкоштовний редактор текстових файлів з підтримкою синтаксису великої кількості мов програмування. Програма має у своєму розпорядженні широкий набір опцій і відрізняється мінімальним споживанням ресурсів процесора [10]. Серед основних можливостей виділяють такі:

- Підсвітка синтаксису тексту залежно від мови програмування, режим якої може налаштувати користувач.
- Можливість згортання блоків згідно з синтаксисом мови програмування.
- Авто-завершення слова, що набирається.
- Одночасна робота з безліччю документів.
- Підтримка регулярних виразів для пошуку й заміни.
- Динамічна зміна вікон перегляду.

– Підтримка великої кількості мов.

На рисунку 3.3 зображено інтерфейс програми Notepad++.

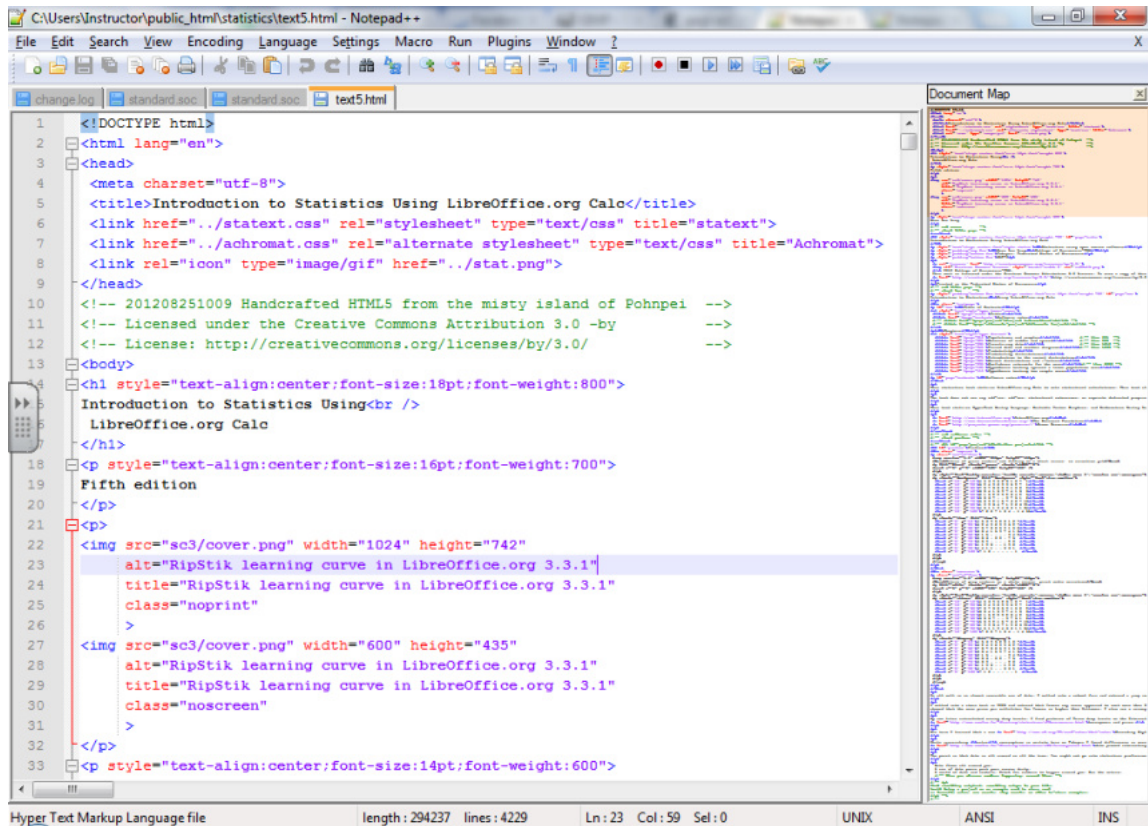


Рисунок 3.3 – Інтерфейс програми Notepad++

Під досить простим зовнішнім виглядом ховається потужний, налаштовуваний редактор коду з безліччю корисних функцій.

Згідно з аналізом розглянутих середовищ розробки, була сформована порівняльна таблиця. Результати аналізу представлено у таблиці 3.1.

Таблиця 3.1 – Порівняльна характеристика розглянутих середовищ розробки

Назва критерію	Назва середовища розробки		
	PhpStorm	Atom	Notepad++
1. Безкоштовна за студентською підпискою	+	+	+
2. Вбудований відладчик	+	+	-
3. Мовна локалізація (українська)	-	+	+
4. Швидкодія	+	-	+
5. Простота у використанні	+	-	+
6. Зручність у масштабних проєктах	+	-	-

Згідно з порівняльною таблицею найліпшим вибором на цей дипломний проєкт є інтегроване середовище розробки PhpStorm.

3.2 Розробка системи

Вебдодаток виконано у сучасному мінімалістичному стилі. Інтерфейс простий, максимально інформативний, наявна анімація.

На рисунку 3.4 зображено головну сторінку (index.php).

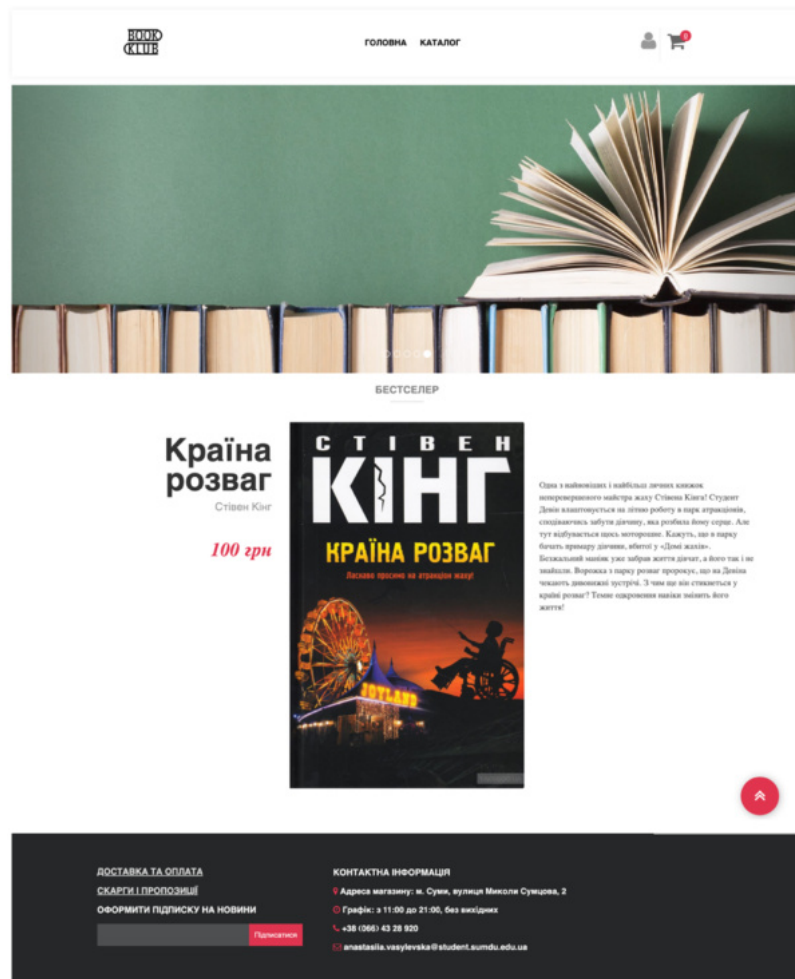


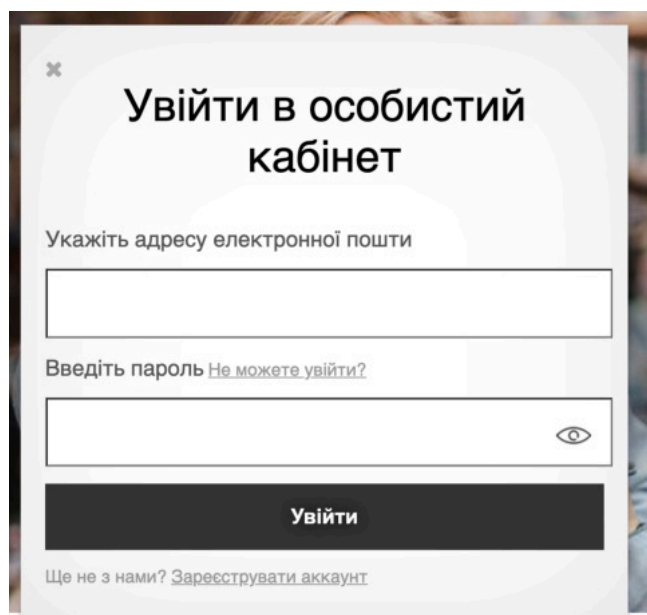
Рисунок 3.4 – Головна сторінка

У верхній частині сторінки розташована «шапка» сайту, що знаходиться в окремому файлі та підключена до кожної сторінки за допомогою функції «require» в PHP. Має гіперпосилання на головну сторінку сайту, каталог, форму авторизації/переходу до особистого кабінету(за успішної авторизації) та кошику з підрахунком загальної кількості доданих товарів.

Далі наведено програмний код перевірки чи авторизовано користувача, а разом з тим, зміни посилання.

```
<?php
// Перевірка, чи користувач авторизований
if (isset($_SESSION['email'])) {
    // Відображення посилання на особистий кабінет
    echo '<a href="personal_page.php" class="icons__link"><i
class="fa fa-user"></i></a>';
} else {
    // Відображення посилання на форму авторизації
    echo '<a onclick="openlogin()" class="icons__link"><i
class="fa fa-user"></i></a>';
}
?>
```

На рисунку 3.5 зображено форму авторизації користувача, що містить посилання на зміни паролю та реєстрації.



The image shows a login form with the following elements:

- Close button (x) in the top left corner.
- Title: **Увійти в особистий кабінет**
- Label: **Укажіть адресу електронної пошти**
- Input field for email.
- Label: **Введіть пароль** with a link [Не можете увійти?](#)
- Input field for password with a visibility icon (eye).
- Button: **Увійти**
- Footer: [Ще не з нами? Зареєструвати аккаунт](#)

Рисунок 3.5 – Форма авторизації користувача

При кліку на елемент під назвою «Не можете увійти?» відкривається форма відновлення паролю (рис. 3.6).

The image shows a mobile application form for password recovery. At the top, there is a close button (x) and the title 'Відновлення паролю'. Below the title, there is a red dashed line. The form contains two input fields: the first is labeled 'Укажіть адресу електронної пошти' and the second is labeled 'Укажіть Ваше ім'я'. At the bottom of the form is a dark button with the text 'Перейти до відновлення паролю'.

Рисунок 3.6 – Форма для переходу на сторінку відновлення паролю

Після коректного заповнення форми з рисунку 3.6 (введений зареєстрований email) та кліку на кнопку «Перейти до відновлення паролю» буде перенаправлення на сторінку відновлення паролю (рис. 3.7).

The image shows a mobile application page for password recovery. The page is titled 'Відновлення паролю'. It features a background image of a library with bookshelves. The form includes a greeting 'Привіт, Настя', the email address 'lolalo8888888@gmail.com', and a secret question: 'Відповідь на секретне запитання. Яке дівоче прізвище матері?'. Below this, there are two password input fields: 'Новий пароль' and 'Повторіть пароль', both with eye icons for visibility. A dark button labeled 'Зберегти' is at the bottom. A link 'Повернутися на сторінку входу' is also present. The footer indicates 'Powered by 000webhost'.

Рисунок 3.7 – Сторінка відновлення паролю

На сторінці розміщено форму відновлення паролю. Для цього потрібно правильно відповісти на секретне питання та ввести новий однаковий пароль.

Наявні повідомлення про невірний ввід даних (рис. 3.8-3.9).

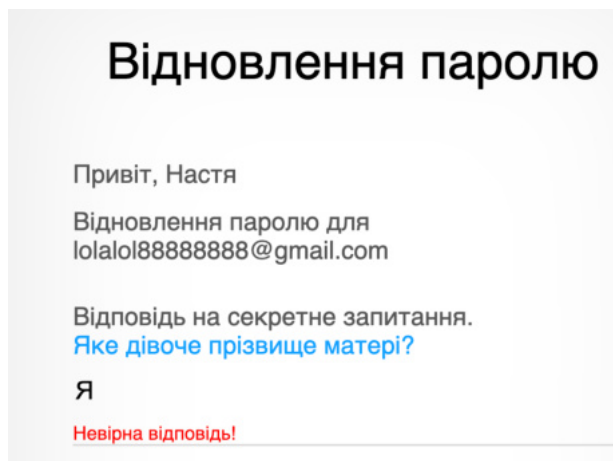


Рисунок 3.8 – Повідомлення щодо помилкової відповіді на секретне запитання

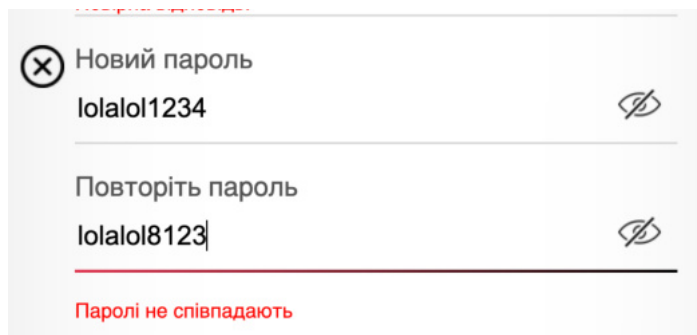


Рисунок 3.9 – Повідомлення про неспівпадіння паролей

На рисунку 3.10 зображено підтвердження вірного вводу паролей.

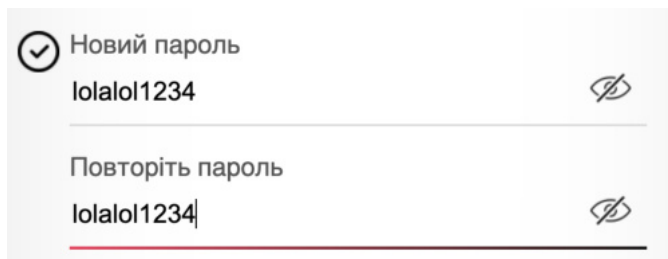


Рисунок 3.10 – Паролі введено вірно

При спробі відправити форму без виправлення ситуацій описаних вище чи успішній зміні паролю з'являється спливаюче вікно повідомлення (рис. 3.11).

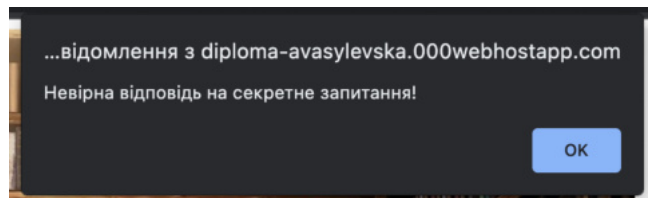


Рисунок 3.11 – Спливаюче повідомлення

На рисунку 3.12 зображено сторінку реєстрації користувача. При реєстрації, як і при відновленні паролю, здійснюється перевірка на заповненість полів перед відправкою форми. Якщо користувач спробує відправити форму з червоними повідомленнями про помилки, система відображає вспливаюче повідомлення без перезавантаження сторінки.

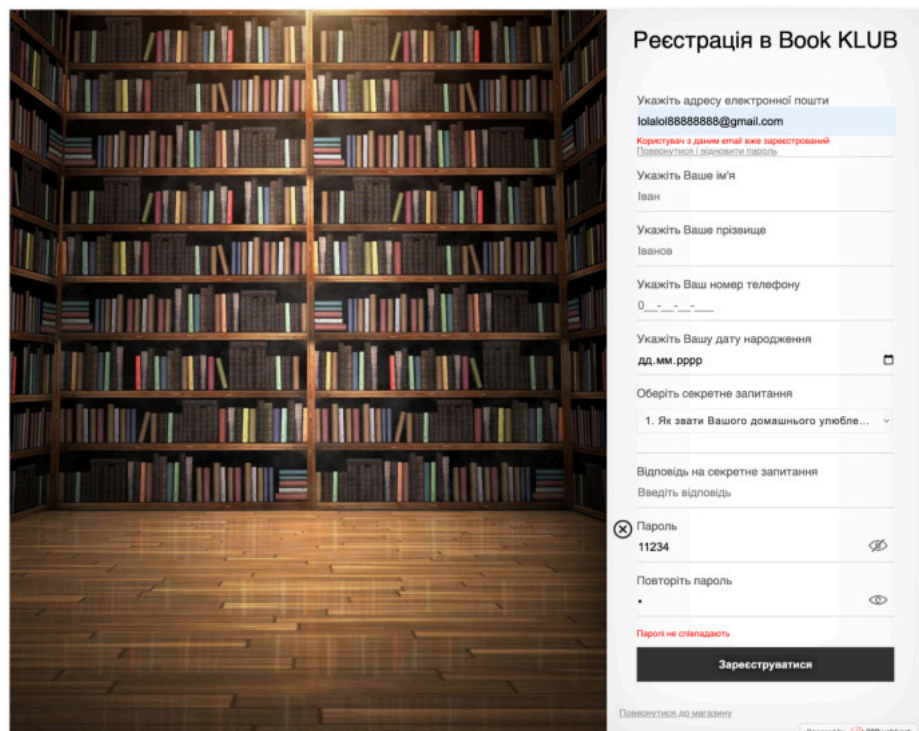


Рисунок 3.12 – Сторінка реєстрації користувача

Для поля номеру телефону передбачена можливість вводу лише цифр.

Після успішної реєстрації здійсниться перенаправлення на домашню сторінку зображену на рисунку 3.4.

При некоректному заповненню полів форми авторизації користувач отримує повідомлення з інформацією, наявна кнопка для приховання/відображення паролю (рис. 3.13, 3.14).

Рисунок 3.13 – Форма авторизації користувача з повідомленням про помилку

Рисунок 3.14 – Пароль відображено

За це відповідає наступний скрипт написаний мовою JavaScript:

```
function show_hide_password(target) {
    var input = document.getElementById('password-input');
    if (input.getAttribute('type') == 'password') {
        target.classList.add('view');
        input.setAttribute('type', 'text');
    } else {
        target.classList.remove('view');
        input.setAttribute('type', 'password');
    }
    return false;
}
```

Після успішної авторизації організовано перенаправлення на особисту сторінку користувача, де можна переглянути та змінити інформацію про користувача, пароль для входу в особистий кабінет (рис. 3.15, 3.16).

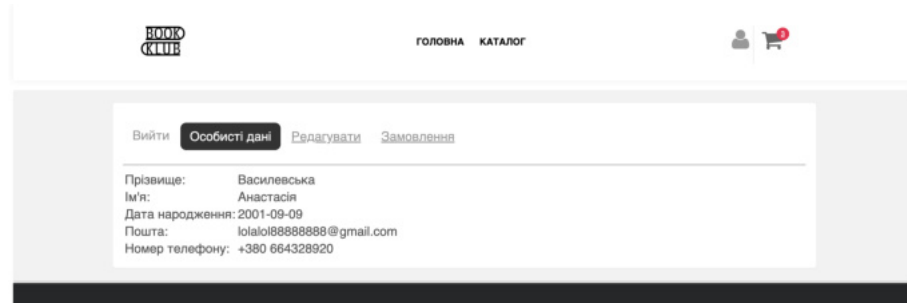


Рисунок 3.15 – Інформація про користувача

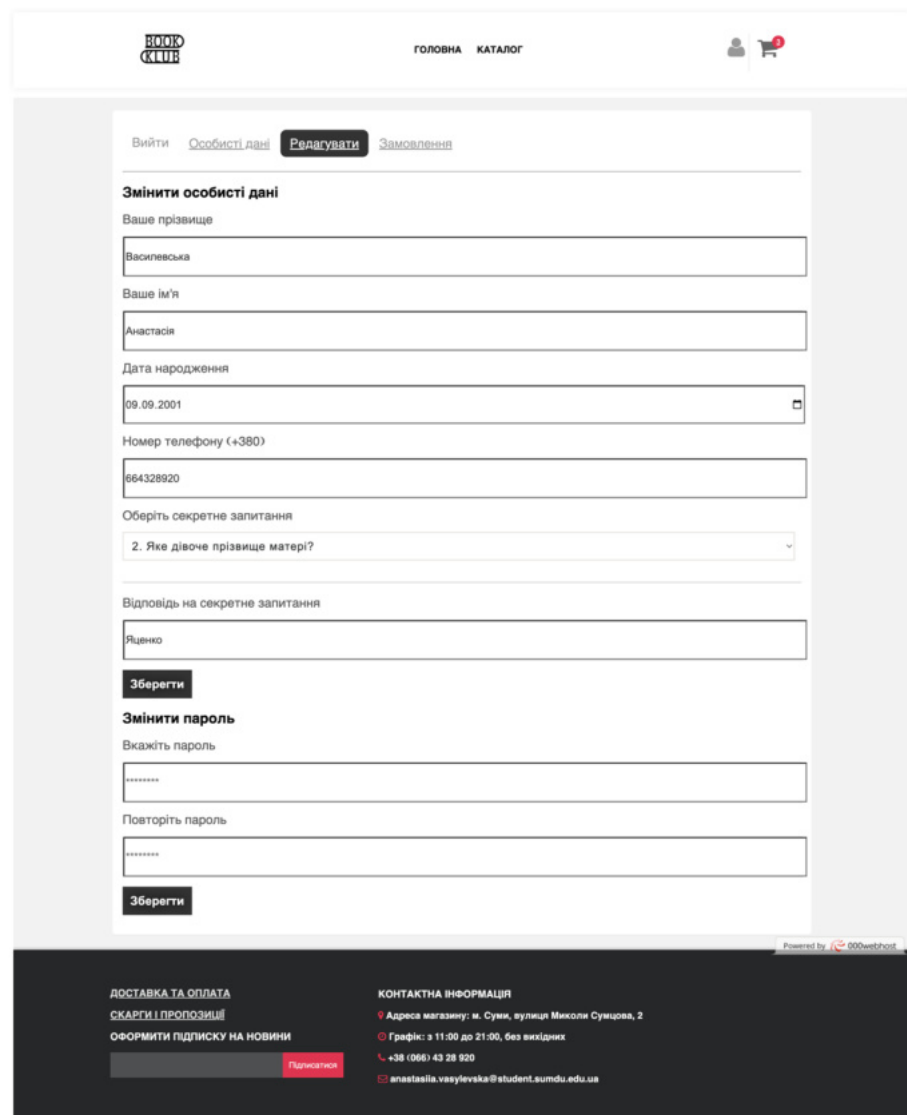


Рисунок 3.16 – Зміна інформації про користувача чи паролю

Також, форма авторизації слугує доступом до адміністративної панелі. Для цього в формі «Вхід до особистого кабінету» ввести логін «admin@admin» та пароль «admin». Користувача «адміністратор» буде перенаправлено на сторінку керування сайтом (рис. 3.17, 3.18).

The screenshot displays the admin interface with three main sections:

- Додавання товару (Add product):** Includes fields for 'Назва книги' (Book name), 'Автор книги' (Author), 'Категорія' (Category), 'Ціна' (Price), and 'Опис книги' (Description). It also features a file upload section for the book cover image.
- Додавання автору (Add author):** Includes fields for 'Ім'я' (Name) and 'Прізвище' (Surname).
- Додавання категорії (Add category):** Includes a 'Категорія' (Category) field.
- База (Database):** A table listing existing items:

Назва	Автор	Категорія	Ціна	Додати	Редагувати
<input type="checkbox"/> Українська абетка із завданнями	Катерина Трофимова	Абетки	319	Додати	Редагувати
<input type="checkbox"/> Книжка для малця. Пухнаста абетка	Марія Морозенко	Абетки	215	Додати	Редагувати

Рисунок 3.17 – Адмін панель

The screenshot shows the 'Редагування товару' (Edit product) form with the following details:

- Назва книги:** Українська абетка із завданнями (Max 255 characters)
- Автор книги:** 4 Катерина Трофимова
- Назва фотографії книги:** Вибрати зображення для завантаження (PNG, JPG, JPEG) розміром менше 10MB. File selected.
- Категорія:** Абетки
- Ціна:** 319
- Опис книги:** Книга познайомить малюка з абеткою та допоможе зробити перші кроки у читанні. На кожній сторінці багато предметів на певну літеру, а також завдання для кращого засвоєння абетки.

Рисунок 3.18 – Редагування товару

Аби уникнути переходу до адміністративної панелі з адресної строки без авторизації виконується перевірка чи розпочата сесія потрібним користувачем.

Скрипт перевірки на сторінці адміністративної панелі:

```
<?php
session_start();
ob_start();
if (empty($_SESSION['email']) || $_SESSION['email'] !==
'admin@admin') {
    $host = $_SERVER['HTTP_HOST'];
    $extra = 'index.php';
    header("Location: http://$host/$extra");
}
?>
```

Скрипт авторизації користувача та перевірки для переходу до адміністративної панелі з форми авторизації:

```
<?php
session_start();
ob_start();
require 'BDconnect.php';
$loguseremail = $_POST['loguseremail'];
$_SESSION['email'] = $loguseremail;
$pass = $_POST['pass'];
$result = mysqli_query($link, "SELECT userparol FROM user
WHERE useremail = '$loguseremail'");
$row = mysqli_fetch_assoc($result);
if ($row) {
    $hashedPasswordFromDatabase = $row['userparol'];
// Порівняти введений пароль з розхешованим паролем з бази даних
    if (password_verify($pass, $hashedPasswordFromDatabase)){
        if($loguseremail=='admin@admin' && $pass=='admin'){
            $host = $_SERVER['HTTP_HOST'];
            $extra = 'php/admin.php';
            header("Location: http://$host/$extra");}
        else{
            $host = $_SERVER['HTTP_HOST'];
            $extra = 'personal_page.php';
            header("Location: http://$host/$extra?".SID);}
    }
}
```



```

    }
}
else exit;
?>

```

При натисканні на іконку кошика відкривається його вміст, загальна сума та форма замовлення. Якщо товари не було додано то буде виведено інформаційне повідомлення про те, що кошик порожній (рис. 3.19, 3.20).

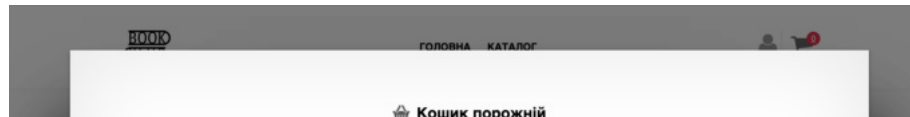


Рисунок 3.19 – Кошик порожній

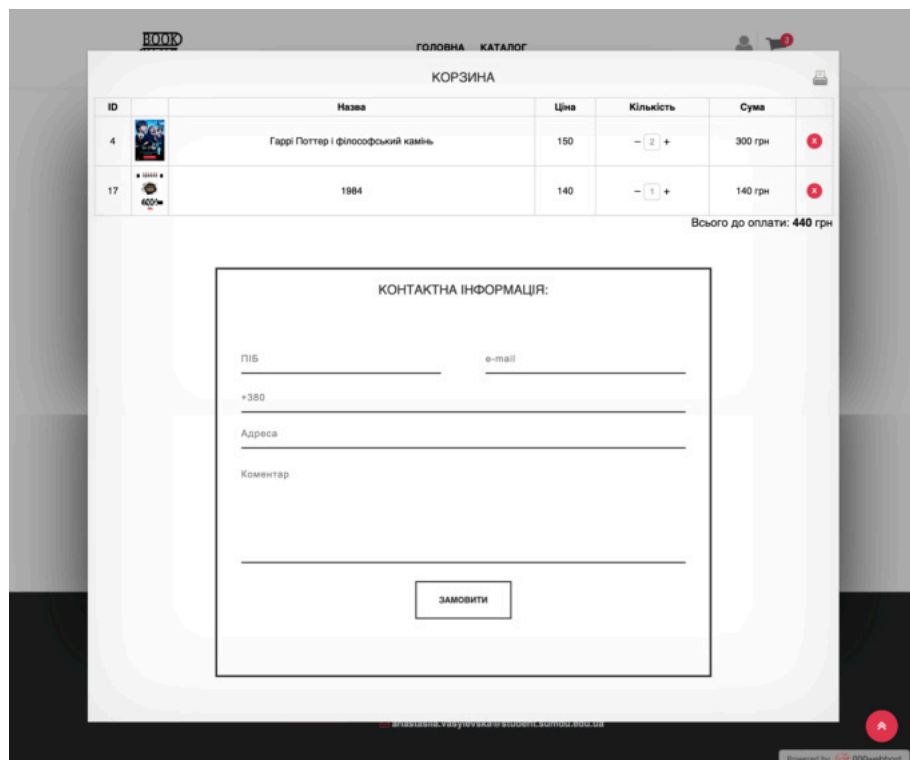


Рисунок 3.20 – Кошик заповнений

Після замовлення на пошту, що вказана у замовленні та пошту адміністратора приходять довідковий лист зі змістом замовлення (рис 3.21).

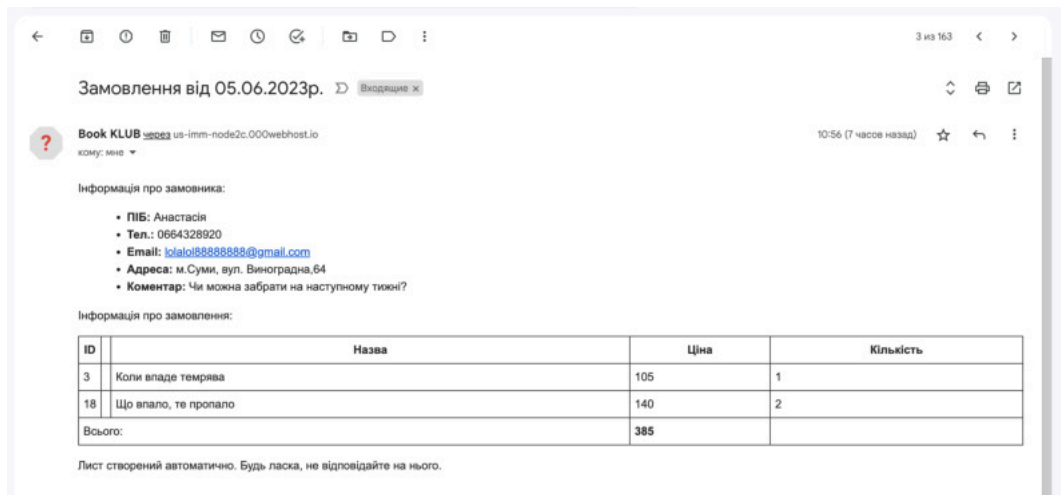


Рисунок 3.21 – Деталі замовлення на пошті користувача

У випадку, якщо користувача з даним email вже зареєстровано на сайті книгарні, історія його замовлень з моменту реєстрації буде збережена та відображена в особистому кабінеті (рис. 3.22).

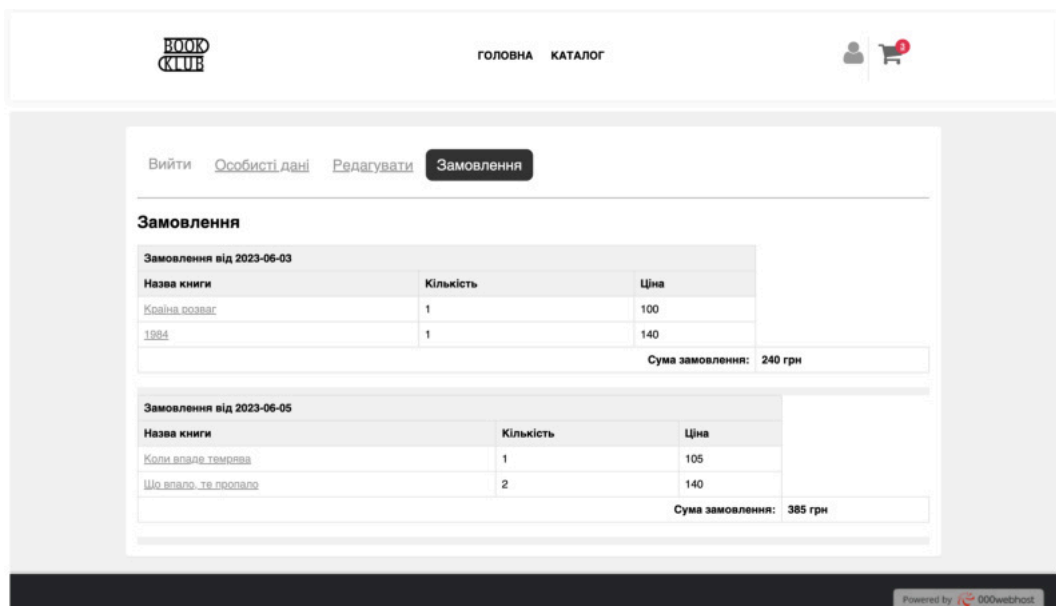


Рисунок 3.22 – Історія замовлень

У нижній частині сторінки розташований «підвал» сайту, що має посилання на форму зворотнього зв'язку, інформаційну сторінку про умови доставки та оплати, формою підписки на розсилку новин на пошту та контактну інформацію магазину.

На рисунку 3.23 зображено сторінку з формою зворотнього зв'язку.

BOOK CLUB

ГОЛОВНА КАТАЛОГ

· ЛИСТ BOOK KLUB ·

ПІБ _____ e-mail _____

+380 _____

Як Ви бажаєте отримати зворотню відповідь?

Поділіться _____

SEND MESSAGE

Powered by 00webhost

ДОСТАВКА ТА ОПЛАТА
СКАРГИ І ПРОПОЗИЦІЇ
ОФОРМИТИ ПІДПИСКУ НА НОВИНИ

Підписатися

КОНТАКТНА ІНФОРМАЦІЯ

Адреса магазину: м. Суми, вулиця Миколи Сумська, 2

Графік: з 11:00 до 21:00, без вихідних

+38 (096) 43 28 909

anastasiia.vasylevska@student.sumdu.edu.ua

Рисунок 3.23 – Сторінка з формою зворотнього зв'язку

Каталог сайту має можливість сортування книг та фільтрації за категоріями, авторами та ціною (рис. 3.24).

BOOK CLUB

ГОЛОВНА КАТАЛОГ

X Категорія: Романи X Категорія: Містика X Автор: Стивен Кінг X Автор: Маргарет Мітчелл

СОРТУВАТИ
Назва книги

ВИДАЛИТИ ВСІ ФІЛЬТРИ

КАТЕГОРІЇ

- Абетки
- Казки
- Містика
- Медична література
- Романи
- Фантастика

АВТОР

- Діокан Роулінг
- Стивен Кінг
- Маргарет Мітчелл
- Катерина Трофимова
- Марія Морозенко
- Адам Кукарський
- Бен Лінч
- Джордж Орвелл

ЦІНА

Від _____

До 99999

Що впало, те пропало
Стивен Кінг
140 грн

Звіяні вітром
Маргарет Мітчелл
120 грн

Керрі
Стивен Кінг
110 грн

Коли впаде темрява
Стивен Кінг
105 грн

Країна розваг
Стивен Кінг
100 грн

Powered by 00webhost

ДОСТАВКА ТА ОПЛАТА
СКАРГИ І ПРОПОЗИЦІЇ
ОФОРМИТИ ПІДПИСКУ НА НОВИНИ

Підписатися

КОНТАКТНА ІНФОРМАЦІЯ

Адреса магазину: м. Суми, вулиця Миколи Сумська, 2

Графік: з 11:00 до 21:00, без вихідних

+38 (096) 43 28 909

anastasiia.vasylevska@student.sumdu.edu.ua

Рисунок 3.24 – Сторінка каталогу

При наведенні на книгу з'являється кнопка що дозволяє одразу додати книгу до кошику (рис. 3.25).

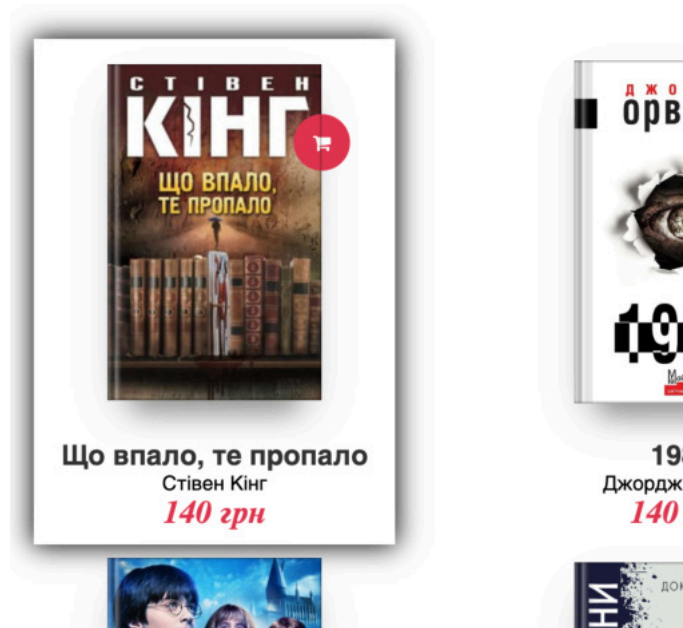


Рисунок 3.25 – Активна книга

Після натискання на зону наведення книги організовано перенаправлення на сторінку з деталями книги де додатково є опис книги та можливість додати її до кошику (рис. 3.26).

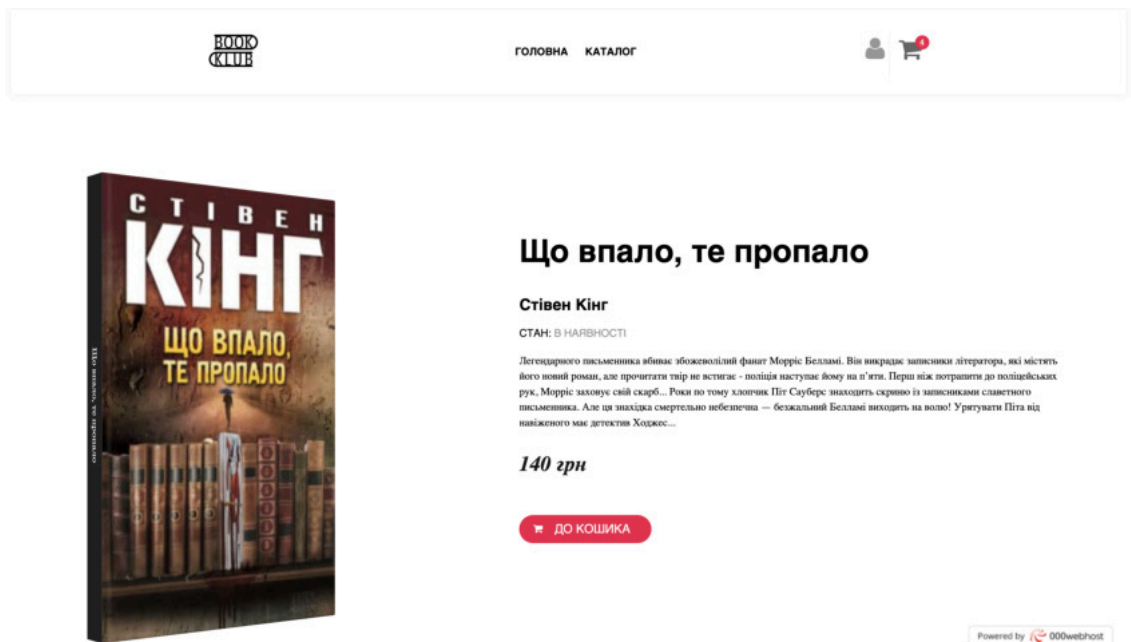


Рисунок 3.26 – Деталі книги

3.3 Тестування веб-застосунку

«Ручне» (manual) тестування є ресурсовитратним через потрібну кількість людей та рутинність задач, тому було розроблено автоматизовані тест-кейси для двох функцій – авторизація та оформлення замовлення.

Даний автоматизований тест було розроблено з використанням мови програмування Python та бібліотеки Selenium. Python являє собою доволі потужну мову програмування, яка є простою для вивчення та розуміння, а це безпосередньо впливатиме на можливість масштабування тестових сценаріїв. Якщо казати про Selenium, то даний інструмент зарекомендував себе як один з найкращих, що дозволяє автоматизувати тестування web-додатків. Він має обширний функціонал, який дозволяє написати сценарії, що покрийуть функціонал сайту по максимуму.

Для того, щоб використати ці інструменти, достатньо зробити декілька кроків:

1. Встановити Python, бажано версії 3.10. Завантажити його можна на офіційному сайті <https://www.python.org/>.

2. За бажанням, завантажити також IDE для більш зручної розробки нашого тестового сценарію, наприклад, PyCharm. За допомогою цієї IDE, при створенні нового проєкту автоматично буде створено нове віртуальне середовище.

3. Коли вже наявне активоване віртуальне середовище, потрібно інстальювати Selenium за допомогою лише однієї команди: `pip install selenium`.

4. В директорію проєкту потрібно помістити WebDriver, за допомогою якого відбуватиметься взаємодія із сайтом. WebDriver це спрощений браузер, який працює у парі з Selenium. Завантажити його також можна на офіційному сайті <https://chromedriver.chromium.org/downloads>.

Перші рядки коду сценарію виглядатимуть наступним чином:

```
import time
import os
from faker import Faker
from selenium import webdriver
```

```

import unittest
from selenium.webdriver.common.by import By
# Бібліотека, яка дозволить генерувати випадкові слова або
речення
fake = Faker()
# Набір констант, які будуть використовуватися у нашому
тестовому сценарії або сценаріях
USERNAME = "lolalol188888888@gmail.com"
PASSWORD = "11"
SITE_URL = "http://diploma-
avasylevska.000webhostapp.com/index.php"
ACTIONS_DELAY = 2
PHONE_NUMBER = "+380664328920"
TEST_NAME = fake.word()
TEST_DESCRIPTION = fake.sentence()
TEST_ADDRESS = fake.sentence()
# Ініціалізація класу, який міститиме у собі setup-функцію, в
якій відбуватиметься ініціалізація нашого webdriver та teardown,
яка виконуватиметься по завершенню виконання сценаріїв
class BlogTest(unittest.TestCase):
# setup функція, в якій ініціалізується наш WebDriver
def setUp(self):
self.browser =
webdriver.Chrome(f'{os.getcwd()}//chromedriver.exe')

```

Код сценарію, який тестуватиме процес авторизації користувача:

```

def test_sign_in_scenario(self):
# Заходимо на сайт
self.browser.get(SITE_URL)
time.sleep(ACTIONS_DELAY)
print("Кликаємо на іконку юзера")
self.browser.find_element(by=By.CSS_SELECTOR,
value='.fa-user').click()
time.sleep(ACTIONS_DELAY)

```

```

print("Вводимо логін")
self.browser.find_element(by=By.NAME,
value='loguseremail').send_keys(USERNAME)
time.sleep(ACTIONS_DELAY)
print("Вводимо пароль")
self.browser.find_element(by=By.ID, value='password-
input').send_keys(PASSWORD)
print("Кликаємо на кнопку 'Увійти'")
self.browser.find_element(by=By.XPATH,
value="//button[contains(., 'Увійти')]").click()
print("Вхід успішний!")

```

У даному сценарії відбувається пошук елементів за певними атрибутами та їх значенням (наприклад, пошук іконки користувача з CSS-селектором, який має значення «.fa-user»). Після цього пошуку, WebDriver повинен або натиснути на елемент за допомогою методу `click()`, або надіслати до нього певні значення за допомогою методу `send_keys()`.

Також між набором дій використовується `time.sleep(ACTIONS_DELAY)`, де `ACTIONS_DELAY` – це константа зі значенням 2, що вже була реалізована на початку нашого коду. Використання такої функції не є випадковим, адже можуть бути такі обставини:

Selenium WebDriver працює настільки швидко, що після натискання на один з елементів може ще не завантажитись інший, а отже його не буде знайдено і тестовий сценарій поверне помилку.

Сайт може мати певний захист від автоматизованого тестування, за рахунок чого тестові сценарії також можуть повертати помилки. Задля запобігання спрацювання такого захисту, дії сповільнено таким чином, ніби ці дії виконує простий користувач.

Вихідний код сценаріїв представлено у додатку Г. Було покрито дві основні функції – авторизація користувача та оформлення ним замовлення (перехід у «Каталог», додавання у «Кошик» двох книг та заповнення форми замовлення).

ВИСНОВКИ

У процесі виконання бакалаврської кваліфікаційної роботи було створено інформаційне та програмне забезпечення веб-орієнтованої платформи книгарні. Метою проекту було створити сайт-інтернет-каталог для моніторингу наявності та актуальних цін на книжкову продукцію книгарні, а також для замовлення товарів. Основним завданням було створення комфортного середовища для перегляду інформації про товари та підприємство.

У процесі розробки було реалізовано кілька етапів. Перший етап включав аналіз вимог, де визначалися потреби та вимоги до функціональності та інтерфейсу користувача платформи. Другий етап полягав у проектуванні системи, включаючи архітектуру, дизайн інтерфейсу користувача та бази даних. На третьому етапі було розроблено програмне забезпечення, використовуючи інструменти HTML, CSS, PHP, JavaScript, jQuery, AJAX та базу даних MySQL. Четвертий етап включав створення адміністративної панелі, що дозволила керувати вмістом каталогу. На п'ятому етапі проводилося тестування, яке включало як ручне, так і автоматизоване тестування з використанням мови програмування Python та інструменту Selenium.

Після успішного завершення тестування, система була розгорнута на сервері та запущена в експлуатацію. Користувачі змогли почати використовувати створений інтернет-каталог для перегляду інформації про книги, перевірки наявності та замовлення товарів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ЗАКОН УКРАЇНИ Про Національну програму інформатизації (ст.1) м. Київ, 4 лютого 1998 року № 74/98-ВР [Електронний ресурс] – режим доступу: <https://zakon.rada.gov.ua/laws/show/74/98-%D0%B2%D1%80#Text>
2. Економічна інформатика: Конспект лекцій у трьох частинах. – Частина 2: Обробка економічної інформації за допомогою електронної таблиці Excel / Укладач А. В. Булашенко. – Суми: Вид-во СумДУ, 2008. – С. 97
3. Інтернет-магазин «bookinstein» [Електронний ресурс] – режим доступу: <https://bookinstein.ua/ua>
4. Інтернет-магазин «yakaboo» [Електронний ресурс] – режим доступу: <https://www.yakaboo.ua/>
5. Стаття про AJAX в енциклопедії «Вікіпедія» [Електронний ресурс] – режим доступу: <https://uk.wikipedia.org/wiki/AJAX>
6. Що таке jQuery? [Електронний ресурс] – режим доступу: <https://jquery.com/>
7. About phpMyAdmin [Електронний ресурс] – режим доступу: <https://www.phpmyadmin.net/>
8. Стаття про PhpStorm в енциклопедії «Вікіпедія» [Електронний ресурс] – режим доступу: <https://uk.wikipedia.org/wiki/PhpStorm>.
9. Стаття про Atom в енциклопедії «Вікіпедія» [Електронний ресурс] – режим доступу: [https://en.wikipedia.org/wiki/Atom_\(text_editor\)](https://en.wikipedia.org/wiki/Atom_(text_editor)).
10. Стаття про Notepad++ в енциклопедії «Вікіпедія» [Електронний ресурс] – режим доступу: <https://uk.wikipedia.org/wiki/Notepad%2B%2B>
11. Adams C., Edwards J. The Art & Science of JavaScript: SitePoint Pty. Ltd., 2008. — 258 p. — ISBN10: 0980285844.
12. Bartlett J. Programming for Absolute Beginners: Using the JavaScript Programming Language: Apress, 2023. — 333 p. — eBook ISBN 978-1-4842-8751-4.
13. Sedliak M., Anbiah R.R.J., Bhattarai R. PHP Ajax Cookbook: Packt Publishing, 2011. - 332 p. ISBN-13: 978-1849513081

14. Imsirovic Ajdin. Bootstrap layouts in depth: Building Amazing Layouts. Book 2 :Leanpub, 2021. — 264 p.
15. Krause Jörg. Introducing Bootstrap 4: Create Powerful Web Applications Using Bootstrap 4.5 :2nd Edition. — Apress Media LLC, 2020. — 374 p. — ISBN13: (electronic): 978-1-4842-6203-0.
16. What is HTML? | HyperText Markup Language explained [Электронный ресурс] – режим доступа: <https://bit.ly/2Yw2V84>
17. What is CSS, How Does It Work and What is It Used For? – Skillcrush [Электронный ресурс] – режим доступа: <https://bit.ly/3e3ePg1>
18. Sriparasa Sai Srinivas. JavaScript and JSON Essentials : Packt Publishing, 2018. — 226 p.
19. Dowden Martine, Dowden Michael. Architecting CSS: The Programmer’s Guide to Effective Style Sheets : Apress, 2020. — 246 p. — ISBN13: (electronic): 978-1-4842-5750-0.
20. Felke-Morris Terry. Web Development and Design Foundations with HTML5 : 9th ed. — Pearson, 2018. — 720 p. — ISBN: 9780134801148.

ДОДАТКИ

Додаток А. Програмний код адміністративної панелі

admin.php

```

<?php
session_start();
ob_start();
if (empty($_SESSION['email']) || $_SESSION['email'] !==
'admin@admin') {
    $host = $_SERVER['HTTP_HOST'];
    $extra = 'index.php';
    header("Location: http://$host/$extra");
}
?>
<!doctype html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Адмін панель || Book KLUB</title>
        <link rel="stylesheet" href="../css/admin.css">
        <script src="../js/main.js"></script>

    </head>
    <body>
        <a href="logout.php" ><i class="menu-item fa fa-sign-
out"><div class="menu-text">Вийти</div></i></a>
        <?php require 'BDconect.php'?>
        <div id="beau-go-top" ><i class="fa fa-angle-double-up"></i></div>
        <script type="text/javascript" src="js/go-top.js"></script>
        <div class="box tovar">
            <div class="box-head">
                <h2>Додавання товару</h2>
            </div>
            <form method="post" action="admin_add.php"
enctype="multipart/form-data">
                <div class="form">
                    <p> <span class="req">Максимальна кількість символів
255</span>

                        <label>Назва книги</label>
                        <input type="text" name="title" class="field size1"
/>

                    </p>
                    <p> <label>Автор книги</label>
                        <select class="field size1" name="avt">
                            <?php

                                $result =mysqli_query($link, "SELECT * FROM avtor");
                                while ($myrow =mysqli_fetch_array($result, MYSQLI_ASSOC)){
                                    echo "<option>";
                                    foreach ($myrow as $buf){echo "$buf "; }
                                    echo "</option>";
                                }

                            ?>

                        </select> </p>

```

```

        <p>
            <label for="book_image">Виберіть зображення для завантаження
            (PNG, JPG, JPEG) розміром менше 10МБ</label>
            <input type="file" id="book_image" name="book_image"
            class="field size1 " accept="image/jpg, image/jpeg, image/png"
            onchange="checkFileSize(this)" multiple>

            <script>
                function checkFileSize(input) {
                    const fileSize = input.files[0].size; // Отримуємо розмір першого
                    вибраного файлу
                    const maxSize = 10 * 1024 * 1024; // Максимальний розмір - 10МБ
                    (10 * 1024 * 1024 байтів)

                    if (fileSize > maxSize) {
                        alert("Файл занадто великий. Максимальний розмір - 10МБ. Можете
                        скористатись сайтом https://www.resizerpixel.com/uk для зменшення
                        розміру");
                        input.value = ''; // Очищаємо значення вибраного файлу
                    }
                }
            </script>
        </p>

        <p>
            <label>Категорія</label>
            <select class="field size1" name="subcategory">
                <?php

                $result =mysqli_query($link, "SELECT subcategory FROM categories");
                while ($myrow =mysqli_fetch_array($result, MYSQLI_ASSOC)){
                echo "<option>";
                foreach ($myrow as $buf)
                {echo "$buf ";
                }
                echo "</option>";
                }
                ?>

                </select></p>
            <p><label>Ціна <span>(копійки вводити через
            крапку.)</span></label>
            <input type="text" name="price" class="field size1"
            />
            </p>
            <p> <label>Опис книги </label>
            <textarea class="field size1" name="description"
            rows="10" cols="30"></textarea>
            </p>
            </div>
            <div class="buttons">
                <input type="submit" class="button" value="Додати"
            />
            </div>
        </form>

        </div>
        <div class="box_in tovar1">
        <div class="box">
            <div class="box-head">

```

```

        <h2>Додавання автора</h2>
    </div>
    <form method="post" action="admin_add_avtor.php">
        <div class="form">
            <p>
                <label>Ім'я</label>
                <input type="text" name="fname" class="field size1"
/>
            </p>
            <p>
                <label>Прізвище</label>
                <input type="text" name="lname" class="field size1"
/>
            </p>
            <div class="buttons">
                <input type="submit" class="button" value="Додати"
/>
            </div>
        </div>
    </form>
</div>
<div class="box">
    <div class="box-head">
        <h2>Додавання категорії</h2>
    </div>
    <form method="post"
action="../php/admin_add_category.php">
        <div class="form">
            <p>
                <label>Категорія <span>(Нехудожня
література)</span></label>
                <input type="text" name="category" class="field
size1" />
            </p>
            <p>
                <label>Похідна від неї <span>(Медична
література)</span></label>
                <input type="text" name="subcategory" class="field
size1" />
            </p>
            <div class="buttons">
                <input type="submit" class="button" value="Додати"
/>
            </div>
        </div>
    </form>
</div>
</div>
<br> <br> <br>
<div class="box box2">
    <div class="box-head">
        <h2 class="left">База</h2>
    </div>
    <div class="table">
        <table width="100%" border="0" cellspacing="0"
cellpadding="0">

```

```

        <tr>
            <th width="13"></th>
            <th>Назва</th>
            <th>Автор</th>
            <th>Категорія</th>
            <th>Ціна</th>
            <th width="110" class="ac"></th>
        </tr>

<?php
$result = mysqli_query ($link, "SELECT *
FROM books, avtor, categories
WHERE `books`.`id_avtor` = `avtor`.`id_avtor`
AND `books`.`subcategory` = `categories`.`subcategory`");
$myrow = mysqli_fetch_array ($result);
$i=0;
do {
    $i++;
    ?>

        <tr>
            <td><input type="checkbox" class="checkbox" /></td>
            <td><h3><?php echo $myrow['title']; ?></h3></td>
            <td><?php echo $myrow['fname']."
$. $myrow['lname']; ?></td>
            <td><?php echo $myrow['subcategory']; ?></td>
            <td><?php echo $myrow['price']; ?></td>
            <td><a href="delete.php?id_book=<?php echo
$myrow['id_book']; ?>" class="ico del">Видалили</a><a class="ico edit"
onclick="document.getElementById('<?php echo
$myrow['id_book']; ?>').style.display = 'block';"
>Редагувати</a></td>
        </tr>
        <div id="<?php echo $myrow['id_book']; ?>" class="modal">
        <div class="Forms">
        <span class="close" onclick="document.getElementById('<?php
echo $myrow['id_book']; ?>').style.display = 'none';">✕</span>
        <div class="box ">
        <div class="box-head">
        <h2>Редагування товару</h2>
        </div>
        <form method="post" action="edit.php?id_book=<?php echo
$myrow['id_book']; ?>" enctype="multipart/form-data">
        <div class="form">
        <p> <span class="req">Максимальна кількість символів
255</span>
        <label>Назва книги</label>
        <input type="text" name="title" class="field size1"
value="<?php echo $myrow['title']; ?>" />
        </p>
        <p> <label>Автор книги</label>
        <select class="field size1" name="avt"/>
        <?php

        $res =mysqli_query($link, "SELECT * FROM avtor");
        while($buf = $res->fetch_array()){
            if($buf['id_avtor'] == $myrow['id_avtor']){
                echo "<option selected>".$buf['id_avtor']."
$. $buf['fname']." ".$buf['lname']."</option>";

```

```

    }
    else{
        echo "<option >".$buf['id_avtor']."
".$buf['fname']." ".$buf['lname']."</option>";
    }
}
?>

</select> </p>
<p>
<label>Назва фотографії книги</label>
<label for="book_image"> <span>Виберіть зображення
для завантаження (PNG, JPG, JPEG) розміром менше 10МБ</span></label>
<input type="file" id="book_image" name="book_image"
class="field size1 " accept="image/jpg, image/jpeg, image/png"
onchange="checkFileSize(this)" multiple>
</p>
<p> <label>Категорія</label>
<select class="field size1" name="subcategory" />
<?php

$res =mysqli_query($link, "SELECT subcategory FROM categories");
while($buf = $res->fetch_array()){
    if($buf['subcategory'] == $myrow['subcategory']){
        echo "<option selected
>".$buf['subcategory']."</option>";
    }
    else{
        echo "<option>".$buf['subcategory']."</option>";
    }
}
?>

</select></p>
<p><label>Ціна <span>(копійки вводити через
крапку.)</span></label>
<input type="text" name="price" class="field size1"
value="<?php echo $myrow['price']; ?>" />
</p>
<p> <label>Опис книги </label>
<textarea class="field size1" name="description"
rows="10" cols="30"> <?php echo $myrow['description']; ?></textarea>
</p>
</div>
<div class="buttons">
<input type="submit" class="button"
value="Редагувати" />
</div>
</form>
</div>
</div>
</body>
</html>

```

admin_add.php

```

<?php
require 'BDconnect.php';
$t = $_POST['avt'];
$id_avtor = preg_replace("/[\D]/", "", $t);
$title=$_POST['title'];
$book_image=$_FILES['book_image']['name'];
$price=$_POST['price'];
$subcategory=$_POST['subcategory'];
$description=$_POST['description'];
$host = 'files.000webhost.com';
$user = 'diploma-avasylevska';
$pass = 'diploma-avasylevska';
$connect = ftp_connect($host, 21);
$result = ftp_login($connect, $user, $pass);
$file = $_FILES['book_image']['name'];

if(move_uploaded_file($_FILES['book_image']['tmp_name'],
$_SERVER['DOCUMENT_ROOT']."/img/book/{$_FILES['book_image']['name']}")){
} else echo('Файл картинки відсутній');
$res= mysqli_query($link, "INSERT INTO books( title, id_avtor,
book_image, price, subcategory, description)
VALUES
('$title', '$id_avtor', '$book_image', '$price', '$subcategory', '$descriptio
n');");
$host = $_SERVER['HTTP_HOST'];
$extra = 'php/admin.php';
header("Location: http://$host/$extra");
?>

```

admin_add_category.php

```

<?php
require 'BDconnect.php';
$subcategory=$_POST['subcategory'];
$category=$_POST['category'];
$result= mysqli_query($link, "INSERT INTO
`categories`(`subcategory`, `category`) VALUES
('$subcategory', '$category');");
$host = $_SERVER['HTTP_HOST'];
$extra = 'php/admin.php';
header("Location: http://$host/$extra");
?>

```

admin_add_avtor.php

```

<?php
require 'BDconnect.php';
$fname=$_POST['fname'];
$lname=$_POST['lname'];
$result= mysqli_query($link, "INSERT INTO `avtor`(`fname`,
`lname`) VALUES ('$fname', '$lname');");
$host = $_SERVER['HTTP_HOST'];
$extra = 'php/admin.php';
header("Location: http://$host/$extra");
?>

```


delete.php

```

<?php
require 'BDconect.php';
$del = $_GET['id_book'];
$result= mysqli_query($link, "DELETE FROM books WHERE id_book =
'$del'");
$host = $_SERVER['HTTP_HOST'];
$extra = 'php/admin.php';
header("Location: http://$host/$extra");
?>

```

edit.php

```

<?php
require 'BDconect.php';
$del = $_GET['id_book'];
$t = $_POST['avt'];
$id_avtor = preg_replace("/[\D]/", "", $t);
$title=$_POST['title'];
$book_image=$_FILES['book_image']['name'];
$price=$_POST['price'];
$subcategory=$_POST['subcategory'];
$description=$_POST['description'];
if ($_FILES['book_image']['error'] === UPLOAD_ERR_OK) {
    $result= mysqli_query($link, "UPDATE books SET `title`='$title',
`id_avtor`='$id_avtor', `book_image`='$book_image', `price`='$price',
`subcategory`='$subcategory', `description`='$description'
WHERE `id_book`='$del'");
    if
(file_exists($_SERVER['DOCUMENT_ROOT']."/img/book/{$_FILES['book_image']
['name']}")) {
        $host = $_SERVER['HTTP_HOST'];
        $extra = 'php/admin.php';
        header("Location: http://$host/$extra");
    }
    else
        if(move_uploaded_file($_FILES['book_image']['tmp_name'],
$_SERVER['DOCUMENT_ROOT']."/img/book/{$_FILES['book_image']
['name']}")){
            $host = $_SERVER['HTTP_HOST'];
            $extra = 'php/admin.php';
            header("Location: http://$host/$extra");
        }
        else{
            echo('Щось пішло не так');
            echo('<a href="php/admin.php">Назад до адмін панелі</a>');
        }
    }
    else {
        $result= mysqli_query($link, "UPDATE books SET `title`='$title',
`id_avtor`='$id_avtor', `price`='$price', `subcategory`='$subcategory',
`description`='$description'
WHERE `id_book`='$del'");
        $host = $_SERVER['HTTP_HOST'];
        $extra = 'php/admin.php';
        header("Location: http://$host/$extra");
    }
?>

```

Додаток Б. Програмний код сторінки «Каталог»

catalog.php

```

<!doctype html><html>
<head>
    <meta charset="utf-8">
    <title>Каталог || Book KLUB</title>
    <link rel="stylesheet" href="css/index.css">
    <link rel="stylesheet" href="css/catalog.css">
    <script src="https://code.jquery.com/jquery-
3.7.0.js"></script>
    <script type="text/javascript" src="js/main.js"></script>
    <script>
        var activeFilters = {
            category: [],
            author: [],
            priceFrom: "",
            priceTo: ""
        };
        $(document).ready(function () {
            $("#sortForm select").change(function () {
                var form = $("#sortForm");
                var url = form.attr("action");
                var formData = form.serialize();
                $.ajax({
                    type: "POST",
                    url: url,
                    data: formData,
                    success: function (response) {
                        $("#bookList").html(response);
                    }
                });
            });
            $("#priceForm input").keyup(function () {
                var form = $("#sortForm");
                var url = form.attr("action");
                var formData = form.serialize();
                $.ajax({
                    type: "POST",
                    url: url,
                    data: formData,
                    success: function (response) {
                        $("#bookList").html(response);
                    }
                });
            });
            function filterByCategory(category) {
                var form = $("#sortForm");
                var url = form.attr("action");
                var index = activeFilters.category.indexOf(category);
                if (index > -1) {
                    activeFilters.category.splice(index, 1);
                }
                else {
                    activeFilters.category.push(category);
                }
            }
        });
    </script>

```

```

form.find("input[name='category']").val(JSON.stringify(activeFilters.category));

    $.ajax({
        type: "POST",
        url: url,
        data: form.serialize(),
        success: function (response) {
            $("#bookList").html(response);
            // Оновити вміст елемента activeFilters
            updateActiveFilters();
        }
    });
}

function filterByAuthor(author) {
    var form = $("#sortForm");
    var url = form.attr("action");
    // Оновити поле категорії в activeFilters
    // Перевірити, чи автор вже обраний
    var index = activeFilters.author.indexOf(author);
    if (index > -1) {
        activeFilters.author.splice(index, 1);
    }
    else {
        activeFilters.author.push(author);
    }
}

form.find("input[name='author']").val(JSON.stringify(activeFilters.author));

    $.ajax({
        type: "POST",
        url: url,
        data: form.serialize(),
        success: function (response) {
            $("#bookList").html(response);
            // Оновити вміст елемента activeFilters
            updateActiveFilters();
        }
    });
}

function updateActiveFilters() {
    var filters = "";
    if (activeFilters.category.length !== 0) {
        activeFilters.category.forEach(function (value) {
            filters += "<span class='filter' data-filter-type='category' data-filter-value='" + value + "'>Категорія: " + value + "</span> ";
        });
    }
    if (activeFilters.author.length !== 0) {
        activeFilters.author.forEach(function (value) {
            filters += "<span class='filter' data-filter-type='author' data-filter-value='" + value + "'>Автор: " + value + "</span> ";
        });
    }
    $("#activeFilters").html(filters);
}

function removeALL() {
    activeFilters = {

```

```

        priceFrom: null,
        priceTo: null,
        author: [],
        category: []
    };
    updateActiveFilters();
    submitFormWithFilters();
}
function removeFilter(filterType, filterValue) {
    activeFilters[filterType] =
activeFilters[filterType].filter(function (value) {
    return value !== filterValue;
});
    updateActiveFilters();
    submitFormWithFilters();
}
$(document).on("click", ".filter", function () {
    var filterType = $(this).data("filter-type");
    var filterValue = $(this).data("filter-value");
    removeFilter(filterType, filterValue);
});
function submitFormWithFilters() {
    var form = $("#sortForm");
    var url = form.attr("action");
    form.find("input[name='price_from']").val(activeFilters.priceFrom);
    form.find("input[name='price_to']").val(activeFilters.priceTo);
    form.find("input[name='author']").val(activeFilters.author.join(",
"));
    form.find("input[name='category']").val(activeFilters.category.join(
    ", "));
    form.find("select[name='sort_by']").val("name");
    $.ajax({
        type: "POST",
        url: url,
        data: form.serialize(),
        success: function (response) {
            $("#bookList").html(response);
        }
    });
}
</script>
</head>
<body>
<?php require 'php/BDconect.php'?>
<?php require 'header.php'?>
    <div class="shop-main-area">
        <div class="book-container">
            <form id="sortForm" action="sort_books.php"
method="post">
                <div class="filter-menu">
                    <div class="sidebar-content">
                        <h3 style="cursor: pointer;"><a style="cursor:
pointer;" onclick="removeALL()">ВИДАЛИТИ ВСІ ФІЛЬТРИ</a></h3>
                        <div class="section-title">
                            <h2>Категорії</h2>
                        </div>
                        <div class="sidebar-category-list">
                            <input name="category" type="hidden">

```

```

        <?php
            $result = mysqli_query($link, "SELECT
subcategory FROM categories");
            echo "<ul>";
            while ($buf = $result->fetch_array()) {
                $category = $buf['subcategory'];
                echo "<li><a
onclick=\"filterByCategory('$category')\">$category</a></li>";
            }
            echo "</ul>";
        ?>
    </div>
</div>
<div class="sidebar-content">
    <div class="section-title"><h2>Автор</h2></div>
    <div class="sidebar-category-list">
        <input name="author" type="hidden">
        <?php
            $result = mysqli_query($link, "SELECT
fname, lname FROM avtor");
            echo "<ul>";
            while ($buf = $result->fetch_array()) {
                $author = $buf['fname'] . " " .
$buf['lname'];
                echo "<li><a
onclick=\"filterByAuthor('$author')\">$author</a></li>";
            }
            echo "</ul>";
        ?>
    </div>
</div>
<div class="sidebar-content">
    <div class="section-title"><h2>Ціна</h2></div>
    <div class="sidebar-category-list"
id="priceForm">
        ВІД <input onkeyup="this.value =
this.value.replace(/[^\d]/g, '');" name="price_from" type="number"
min="0" placeholder="0"><br><br>
        ДО <input onkeyup="this.value =
this.value.replace(/[^\d]/g, '');" name="price_to" type="number" min="0"
placeholder="9999999999">
    </div>
</div>
<div class="catalog-container">
    <div id="activeFilters"></div>
    <div class="shop-item-filter">
    <div class="filter-by">
        <h3>СОРТУВАТИ</h3>
        <select name="sort_by">
            <option value="name">Назва
книги</option>
            <option value="priceA">Ціна (за
спаданням)</option>
            <option value="priceZ">Ціна (за
зростанням)</option>
        </select>
    </div>
</div>

```

```

        </div>
        <div class="box" id="bookList">
            <?php include 'sort_books.php'; ?>
        </div>
    </div>
</form>
</div>
</div>
</body>
<?php require 'footer.php'?>
</html>

```

sort_books.php

```

<!doctype html><html>
<head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="css/index.css">
    <link rel="stylesheet" href="css/catalog.css">
</head>
<?php require 'php/BDconnect.php'?>
<?php
$sortBy = $_POST['sort_by'] ?? 'name';
$priceFrom = $_POST['price_from'];
$priceTo = $_POST['price_to'];
$authors = $_POST['author'] ?? [];
$categories = $_POST['category'] ?? [];
$sql = "SELECT * FROM books
        INNER JOIN avtor ON books.id_avtor = avtor.id_avtor
        INNER JOIN categories ON books.subcategory =
categories.subcategory";
$conditions = array();
$authorConditions = array();
$categoryConditions = array();
if (!empty($authors)) {
    $authors = str_replace(array('[', ']', '"'), '', $authors);
    $authors = str_replace(' ', ' ', trim($authors, ' '));
    $authors = '"' . str_replace(' ', ' ', $authors) . '"';
    $authorConditions[] = "CONCAT(avtor.fname, ' ', avtor.lname) IN
($authors)";
    $conditions[] = "(" . implode(" OR ", $authorConditions) . ")";
}
if (!empty($categories)) {
    $categories = str_replace(array('[', ']', '"'), '',
$categories);
    $categories = str_replace(' ', ' ', trim($categories, ' '));
    $categories = '"' . str_replace(' ', ' ', $categories) . '"';
    $categoryConditions[] = "books.subcategory IN ($categories)";
    $conditions[] = "(" . implode(" OR ", $categoryConditions) .
")";}
if (!empty($priceFrom) && !empty($priceTo)) {
    $priceFrom = intval($priceFrom);
    $priceTo = intval($priceTo);
    $conditions[] = "books.price BETWEEN $priceFrom AND $priceTo";
} elseif (empty($priceFrom) && !empty($priceTo)) {
    $priceTo = intval($priceTo);
    $conditions[] = "books.price <= $priceTo";
} elseif (!empty($priceFrom) && empty($priceTo)) {
    $priceFrom = intval($priceFrom);

```

```

        $conditions[] = "books.price >= $priceFrom"; }
    if (!empty($conditions)) { $sql .= " WHERE " . implode(" and ",
$conditions); }
    switch ($sortBy) {
        case 'priceA':
            $sql .= " ORDER BY books.price DESC";
            break;
        case 'priceZ':
            $sql .= " ORDER BY books.price ASC";
            break;
        case 'name':
            $sql .= " ORDER BY books.title ASC";
            break;
        default:
            $sql .= " ORDER BY books.title ASC";
            break;
    }
    $result = mysqli_query($link, $sql);
    if (!$result) {
        die('Помилка виконання запиту: ' . mysqli_error($link));
    }
    if (mysqli_num_rows($result) > 0) {
        while ($myrow = mysqli_fetch_assoc($result)) {
            ?>
            <div id="activeFilters"></div>
            <div class="but item_box">
                <div class="book-item">
                    <a href="deta1.php?title=<?php echo $myrow['title']; ?>">
                        <div class="book-image">
                            ">
                        </div>
                    </a>
                    <div class="book-actions">
                        <button class="button add_item" data-id="<?php echo
$myrow['id_book']; ?>" data-title="<?php echo $myrow['title']; ?>" data-
price="<?php echo $myrow['price']; ?>" data-img="<?php echo
"img/book/" . $myrow['book_image']; ?>"></button>
                    </div>
                </div>
                <div class="book-info">
                    <span class="book-name item_title"><a
href="deta1.php?title=<?php echo $myrow['title']; ?>"><?php echo
$myrow['title']; ?></a></span><br>
                    <span class=" " "><?php echo $myrow['fname'] . "
" . $myrow['lname']; ; ?></a></span><br>
                    <span class="book-price item_price"><b><?php echo
$myrow['price'] ?> грн</b></span>
                </div>
            </div>
            <?php
        }
    }
    ?>
</html>

```

Додаток В. Програмний код обробника події кнопки «вгору»

Кнопка, що розташована в `header.php` і автоматично підключена до всіх сторінок

```
<div id="beau-go-top" ><i class="fa fa-angle-double-up"></i></div>
go-top.js
const goTopBtn = document.getElementById("beau-go-top");
// Функція, яка перевіряє, чи прокручена сторінка достатньо далеко
вниз, щоб показувати кнопку "вгору"
function trackScroll() {
    const scrolled = window.pageYOffset;
    const coords = document.documentElement.clientHeight;

    if (scrolled > coords) {
        goTopBtn.classList.remove("hidden");
    }
    if (scrolled < coords) {
        goTopBtn.classList.add("hidden");
    }
    if (scrolled === 0) {
        goTopBtn.classList.remove("hidden");
    }
}
// Викликаємо функцію при прокручуванні сторінки
window.addEventListener("scroll", trackScroll);
// Функція, яка повертає сторінку вгору при натисканні на кнопку
"вгору"
function backToTop() {
    if (window.pageYOffset > 0) {
        window.scrollTo(0, -80);
        setTimeout(backToTop, 10);
    }
}
// Додаємо обробник події для кнопки "вгору"
goTopBtn.addEventListener("click", backToTop);
```


Додаток Г. Тестування веб-застосунку

```

import time
import os
from faker import Faker
from selenium import webdriver
import unittest
from selenium.webdriver.common.by import By

fake = Faker()

USERNAME = "lolalol188888888@gmail.com"
PASSWORD = "11"
SITE_URL = "http://diploma-avasylevska.000webhostapp.com/index.php"
ACTIONS_DELAY = 2
PHONE_NUMBER = "+380664328920"
TEST_NAME = fake.word()
TEST_DESCRIPTION = fake.sentence()
TEST_ADDRESS = fake.sentence()

class BlogTest(unittest.TestCase):
    def setUp(self):
        self.browser =
webdriver.Chrome(f'{os.getcwd()}//chromedriver.exe')

    def test_sign_in_scenario(self):
        # Заходимо на наш сайт
        self.browser.get(SITE_URL)
        time.sleep(ACTIONS_DELAY)
        print("Клікаємо на іконку юзера")
        self.browser.find_element(by=By.CSS_SELECTOR, value='.fa-
user').click()
        time.sleep(ACTIONS_DELAY)

        print("Вводимо логін")
        self.browser.find_element(by=By.NAME,
value='loguseremail').send_keys(USERNAME)
        time.sleep(ACTIONS_DELAY)
        print("Вводимо пароль")

```

```

        self.browser.find_element(by=By.ID, value='password-
input').send_keys(PASSWORD)
        print("Клікаємо на кнопку 'Увійти'")
        self.browser.find_element(by=By.XPATH,
value="//button[contains(., 'Увійти')]").click()
        print("Вхід успішний!")
    def test_order_book_scenario(self):
        # Заходимо на наш сайт
        self.browser.get(SITE_URL)
        time.sleep(ACTIONS_DELAY)
        print("Клікаємо на іконку юзера")
        self.browser.find_element(by=By.CSS_SELECTOR, value='.fa-
user').click()
        time.sleep(ACTIONS_DELAY)
        print("Вводимо логін")
        self.browser.find_element(by=By.NAME,
value='loguseremail').send_keys(USERNAME)
        time.sleep(ACTIONS_DELAY)
        print("Вводимо пароль")
        self.browser.find_element(by=By.ID, value='password-
input').send_keys(PASSWORD)
        print("Клікаємо на кнопку 'Увійти'")
        self.browser.find_element(by=By.XPATH,
value="//button[contains(., 'Увійти')]").click()
        print("Вхід успішний!")
        time.sleep(ACTIONS_DELAY)
        print("Переходимо до нашого каталогу")
        self.browser.find_element(by=By.XPATH,
value="//a[contains(text(), 'Каталог')]").click()
        time.sleep(ACTIONS_DELAY * 3)

        print("Обираємо перші дві книги у каталозі та додаємо їх до
кошика")

        self.browser.get("http://diploma-
avasylevska.000webhostapp.com/detal.php?title=%20%D0%A9%D0%BE%20%D0%B2%D
0%BF%D0%B0%D0%BB%D0%BE,%20%D1%82%D0%B5%20%D0%BF%D1%80%D0%BE%D0%BF%D0%B0%
D0%BB%D0%BE")
        time.sleep(ACTIONS_DELAY)

```

```

        self.browser.find_element(by=By.CSS_SELECTOR,
value="button.button:nth-child(1)").click()
        time.sleep(ACTIONS_DELAY)
        self.browser.get("http://diploma-
avasylevska.000webhostapp.com/detal.php?title=1984")
        time.sleep(ACTIONS_DELAY)
        self.browser.find_element(by=By.CSS_SELECTOR,
value="button.button:nth-child(1)").click()
        time.sleep(ACTIONS_DELAY)
        print("Клікаємо на іконку з нашим кошиком")
        self.browser.find_element(by=By.CSS_SELECTOR, value=".fa-
shopping-cart").click()
        time.sleep(ACTIONS_DELAY)
        print(f"Вводимо такі дані у поля для вводу: {TEST_NAME=}, "
              f"{USERNAME=}, "
              f"{PHONE_NUMBER=}, "
              f"{TEST_ADDRESS=}, "
              f"{TEST_DESCRIPTION=}")
        self.browser.find_element(by=By.NAME,
value="user_name").send_keys(TEST_NAME)
        self.browser.find_element(by=By.ID,
value="email_input").send_keys(USERNAME)
        self.browser.find_element(by=By.ID,
value="telephone_input").send_keys(PHONE_NUMBER)
        self.browser.find_element(by=By.NAME,
value="user_address").send_keys(TEST_ADDRESS)
        self.browser.find_element(by=By.NAME,
value="user_comment").send_keys(TEST_DESCRIPTION)
        print("Замовлення зроблено успішно!")

    def tearDown(self):
        self.browser.quit()

if __name__ == '__main__':
    unittest.main()

```