Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня

бакалавра
(назва освітньо-кваліфікаційного рівня)

студента *Францев Богдан Вікторович*
(ПІБ)

академічної групи *122-20ск*
(шифр)

спеціальності *122 Комп'ютерні науки*
(код і назва спеціальності)

освітньої програми *Комп'ютерні науки*
(назва освітньої програми)

на тему: Розробка автоматизованої інформаційної системи з продажу книжок.

Керівники	**Прізвище, ініціали**	**Оцінка за шкалою**		**Підпис**
		рейтинговою	**інституційною**	
кваліфікаційної роботи				
розділів:				
спеціальний	*Кабак Л.В.*			
економічний	*проф. Вагонова О.Г.*			
Рецензент	*доц. Каштан В.Ю.*			
Нормоконтролер	*доц. Гуліна І.Г.*			

Дніпро
2023

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:
завідувач кафедри
програмного забезпечення комп'ютерних систем
(повна назва)

_____ М.О. Алексєєв
(підпис) (прізвище, ініціали)

« » _____ 2023 року

ЗАВДАННЯ
на кваліфікаційну роботу

бакалавра
(назва освітньо-кваліфікаційного рівня)

студента *122-20ск* *Францев Б.В.*
(група) (прізвище та ініціали)

тема кваліфікаційної роботи Розробка автоматизованої інформаційної системи з продажу книжок.

затверджена наказом ректора НТУ «ДП» від *16.05.2023* № *350-с*

Розділ	Зміст виконання	Термін виконання
Спеціальний	***На основі матеріалів проєктно-технологічної практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми***	***13.05.2023 р.***
Економічний	***Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки***	***27.05.2023 р.***

Завдання видав _____ *Кабак Л.В.*
(підпис) (посада, прізвище, ініціали)

Завдання прийняв до виконання _____ *Францев Б.В.*
(підпис) (прізвище, ініціали)

Дата видачі завдання: *14.01.2023 р.*

Термін подання кваліфікаційної роботи до ЕК: *3,07.2023 р.*

РЕФЕРАТ

Пояснювальна записка: 77 с., 27 рис., 18 дод., 7 джерел.

Об'єкт розробки: Розробка автоматизованої інформаційної системи з продажу книжок.

Мета дипломного проекту: розробити інтернет-магазин за допомогою якого клієнти зможуть шукати товар за допомогою локальних пошукових систем, створювати замовлення продукту та переглядати існуючі замовлення.

У вступі розглядається аналіз та сучасний стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, наведено обґрунтування актуальності теми та уточнюється постановка завдання.

У першому розділі проведено аналіз предметної області, визначено актуальність завдання та призначення розробки, розроблена постановка завдання, задані вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі виконано аналіз існуючих рішень, обрано вибір мов для розробки, виконано розробку програми та її проєктування, наведено опис алгоритму і структури функціонування системи, визначені вхідні і вихідні дані, наведені характеристики складу параметрів технічних засобів, описаний виклик та завантаження застосунку, описана робота веб-сайту.

В економічному розділі визначено трудомісткість розробленої інформаційної підсистеми, проведений підрахунок вартості роботи по створенню інтернет-магазину та розраховано час на його створення.

Список ключових слів: CSS, HTML, DATABASE, MySQL, phpMyAdmin, OpenServer, PHP, Visual Studio Code.

ABSTRACT

Explanatory note: 77 pages, 27 pics,18 apps, 24 sources

Object of development: Development of an automated information system for the sale of books.

The purpose of the diploma project: to develop a website that will allow customers to search for products using local search system, to create product orders and view existing orders.

The introduction considers the analysis and the current state of the problem, specifies the purpose of the qualification work and the field of its application, provides a justification for the relevance of the topic and clarifies the problem.

In the first section the analysis of the subject area is carried out, the urgency of the task and purpose of development is defined, the statement of the task is developed, requirements to software realization, technologies and software are set.

The second section analyzes the existing solutions, selects the platform for development, performs design and development of the program, describes the algorithm and structure of the system, determines the input and output data, provides the characteristics of the parameters of website, describes the call and application load, describes the website.

In the economic section, the complexity of the developed information subsystem is determined, the cost of work on creating the application is calculated and the time for its creation is calculated.

List of keywords: CSS, HTML, DATABASE, MySQL, phpMyAdmin, OpenServer, PHP, Visual Studio Code.

ЗМІСТ

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ **6**

ВСТУП **7**

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ **9**

1.1. Загальні відомості предметної області 9

1.2. Призначення розробки та область застосування **13**

1.3. Підстава для розробки 13

1.4. Постановка завдання 14

1.5. Вимоги до програми або програмного виробу 14

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ **16**

2.1. Функціональне призначення системи 16

2.2. Опис застосованих математичних методів 16

2.3. Опис використаних технологій та мов програмування 17

2.4. Опис структури системи та алгоритмів її функціонування 21

2.5. Обгрунтування та організація вхідних та вихідних даних програми 28

2.6. Опис роботи розробленої системи 29

РОЗДІЛ 3. ЕКОНОМІЧНА ЧАСТИНА **40**

3.1. Визначення трудомісткості розробки програмного забезпечення 40

3.2. Витрати на створення програмного забезпечення 43

Висновок 44

ВИСНОВКИ **45**

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ **46**

КОД ПРОГРАМИ 47

ВІДГУК 74

РЕЦЕНЗІЯ 75

ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ 76

ПЕРЕЛІК ФАЙЛІВ НА ОПТИЧНОМУ НОСІЇ 77

СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

HTML – HyperText Markup Language(Мова розмітки гіпертексту).

CSS – Cascading Style Sheets(Каскадні таблиці стилів).

PHP – Hypertext Preprocessor (Препроцесор гіпертексту).

HTTP – HyperText Transfer Protocol(Протокол передачі гіпертексту).

MySQL – Structured Query Language (СКБД).

ВСТУП

У наш час, у цифрову епоху, майже всі користуються інтернетом. З кожним днем все менше і менше люди шукають бібліотеки та книжкові магазини, і тому підприємці майже не відкривають нових магазинів, чого не скажеш про інтернет-магазини. Майже кожний в наш час користувався інтернет-магазином, чи то книжковий магазин, чи то магазин побутової техніки.

З одного боку, інтернет-магазини зберігають багато часу клієнтів. Вони не витрачають час на пошук магазину, в якому є саме той продукт який вони шукають, не витрачають час на дорогу, вони просто зайшли на сайт, вписали в пошукову строку назву книжки та обробили замовлення за декілька хвилин, після чого одразу отримали електронний або фізичний формат книжки через декілька днів на пошту або кур'єрською доставкою.

Також інтернет-магазини мають більш ширший вибір та асортимент книжок, можна знайти рідкісні видання, книжки, які буде важко знайти в місцевих магазинах. Клієнту чи підприємцю буде легше порівняти ціну книжок на ринку, подивитися відгуки та рейтинг магазину, тощо.

Маючи книжковий інтернет-магазин підприємцям не треба платити великі суми на оренду приміщення, зменшується кількість робочого персоналу, зручніше керувати процесом та слідкувати за прибутком, мінімальна паперова робота, більше цифрових документів.

З іншого боку, треба знайти спеціаліста або команду таких для розробки дизайну UI/UX інтерфейсу та розробки програмного коду, підтримувати щомісячно сервер та виявляти помилки в коді або в дизайні.

Для клієнта можуть виникнути проблеми в неможливості фізичного огляду, купуючи книгу в інтернет-магазині не можна її оглянути, подивитись якість перетину, паперу або друку, відсутності спонтанних покупок, треба шукати конкретну книжку або заздалегідь знати що шукаєш, можуть бути проблеми з доставкою, іноді доставка може затримуватися по різних причинах, та її ціна може бути дорожче ніж сама книга, появляються проблеми з поверненням, якщо клієнту не сподобалась книга або він бажав більшого і хоче

її повернути, то можуть статися проблеми, тому що деякі інтернет-магазини можуть мати обмеження або складні процедури повернення.

Головною метою цієї дипломної роботи є створення книжкового онлайн-магазину, який буде зрозумілим та зручним для кожного користувача.

Підбиваючи підсумки була обрана тема дипломного проєкту: «Розробка автоматизованої інформаційної системи з продажу книжок.»

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Загальні відомості с предметної галузі

Книжковий інтернет-магазин працює на основі електронної платформи, де клієнти можуть переглядати каталог книжок, зробити замовлення та здійснити оплату. Основний принцип роботи книжкового інтернет-магазину включає наступні етапи:

Онлайн-каталог — книжковий інтернет-магазин має свій веб-сайт, де розміщений онлайн-каталог з вибраними книжками. Клієнти можуть шукати книжки за назвою, автором, жанром, видавництвом або іншими параметрами за допомогою пошукової функції або навігаційних категорій.

Вибір книжки — клієнти переглядають сторінки книжок, де представлені заголовок, автор, опис, обкладинка, ціна та інша інформація. Вони можуть додати книжку до свого кошика для подальшого замовлення.

Оформлення замовлення — коли клієнт закінчує вибір книжок, він переходить до процесу оформлення замовлення. Тут він вказує свої контактні дані, адресу доставки, обирає метод доставки та спосіб оплати.

Оплата — після оформлення замовлення клієнт переходить до етапу оплати. Книжковий інтернет-магазин може приймати різні методи оплати, такі як кредитні або дебетові картки, електронні платіжні системи або готівковий платіж при отриманні (якщо це доступно).

Обробка замовлення — після отримання оплати книжковий інтернет-магазин обробляє замовлення. Це включає підготовку книжок до доставки, упакування та організацію доставки до вказаної адреси.

Доставка — книжковий інтернет-магазин співпрацює з кур'єрськими службами або поштовими операторами для доставки замовлень до клієнтів. Доставка може бути здійснена до дверей або в пункт самовивозу, в залежності від вибраного методу доставки.

Обслуговування післяпродажного сервісу — книжковий інтернет-магазин надає післяпродажне обслуговування, включаючи можливість повернення або обміну товару, вирішення питань щодо замовлень та надання додаткової інформації про книжки.

Для приведення прикладу, я опишу один з перших книжкових інтернет-магазинів, а саме Amazon.

Amazon - це один з найбільших інтернет-магазинів у світі та глобальна електронна комерційна компанія. Заснований в 1994 році Джеффом Безосом як онлайн-книжковий магазин, Amazon швидко розширив свої послуги та асортимент продукції, перетворившись на універсальний інтернет-ритейлер.
Історія Amazon може бути поділена на кілька важливих етапів:

Заснування та книжковий магазин: В липні 1994 року Джефф Безос заснував компанію Amazon під назвою "Cadabra". Початково Amazon був спрямований на продаж книжок онлайн, з метою надати більший вибір книжок, ніж традиційні кам'яні книжкові магазини. У липні 1995 року він змінив назву компанії на Amazon, і перші замовлення книжок були прийняті в той же рік.

Розширення асортименту: Після успіху в книжковому сегменті Amazon почав розширювати свій асортимент продукції. В 1998 році було придбано онлайн-магазин CDNow, що дозволило компанії приступити до продажу музичних CD. Поступово в асортименті Amazon з'явилися електроніка, комп'ютери, одяг, електронні книги, дитячі товари та багато іншого.

Платформа електронної комерції та Amazon Marketplace: У 2000 році Amazon запустив платформу електронної комерції під назвою Amazon Marketplace. Це дало можливість стороннім продавцям продавати свої товари через платформу Amazon, розширюючи асортимент і привертаючи більше покупців. Amazon також розробив власну систему виконання замовлень під назвою Fulfillment by Amazon (FBA), яка дозволяє продавцям зберігати свою продукцію в складах Amazon та використовувати їх для доставки та обробки замовлень.

Розвиток технологій та іновацій: Amazon активно інвестував у розробку технологій та інновації. Вони представили електронну книгу Kindle в 2007 році, що революціонізувало спосіб читання книг. Також були запущені Amazon Prime (сервіс передплати з безкоштовною доставкою), Amazon Echo (персональний голосовий помічник зі вбудованим голосовим асистентом Alexa) та Amazon Web Services (хмарна платформа для бізнесу).

Глобальний лідер електронної комерції: За десятиліття розвитку Amazon став глобальним лідером електронної комерції. Компанія активно розширює свою присутність в різних країнах, запускає нові сервіси та розвивається в галузях, таких як потокове відео, харчування, штучний інтелект та інше. Amazon також придбав інші компанії, включаючи Whole Foods Market, що розширило їхні можливості в сегменті продажу продуктів харчування.

Сьогодні Amazon продовжує займати сильну позицію на світовому ринку електронної комерції та інноваційних технологій, надаючи широкий спектр товарів та послуг для мільйонів клієнтів по всьому світу.

Візуалізація теж одна з головних задач в створенні інтернет-магазину, одні з головних частин програмного забезпечення та веб-додатків для звичайного користувача це:

- Зручний та приємний інтерфейс.
- Якнайбільша візуалізація компонентів системи.
- Легкість в експлуатації.
- Захищеність даних користувача.

Інтерфейс — це один з ключових елементів візуалізації програмного забеспечення, його можна поділити на дві частини, а саме на UI-інтерфейс та UX-інтерфейс.

UI(*або User Interface*) — це інтерфейс користувача і представляє собою спосіб взаємодії між користувачем і комп’ютерною системою, програмою або пристроєм. UI може включати в себе кнопки, меню, форми, вікна, панелі і т.д.

Метою UI є забезпечення зручного та ефективного користування системою. UI включає в себе графічну частину(інтерфейс, що візуально відображається на екрані) і поведінку(реакцію системи на дії користувача).

UX(*або User Experience*) – це інтерфейс користувача, який формується на створенні задоволення, задоволення потреб користувачів та забезпеченні позитивного враження від взаємодії, сприяння зручності використання і підвищення задоволення користувача. Він враховує аспекти взаємодії користувача з продуктом або системою, такі як навігація, організація контенту, доступність, візуальний дизайн, продуктивність та ефективність використання. Також враховуються потреби, цілі та контекст користувача, створюючи оптимальні умови для задоволення його очікувань і досягнення поставлених цілей.

Для створення UI інтерфейсу існують графічні редактори та інструменти, наприклад такі:

- Sketch: Це векторний графічний редактор для дизайну інтерфейсу, який часто використовується для створення макетів, іконок, кнопок та інших елементів інтерфейсу.
- Adobe XD: Це інтегроване середовище для проектування, прототипування та спільної роботи над дизайном інтерфейсу. Він дозволяє створювати інтерактивні прототипи, додавати анімацію та ефекти.
- Figma: Це веб-платформа для дизайну інтерфейсу, яка дозволяє створювати, прототипувати та спільно працювати над проектами. Figma має вбудовані засоби для спілкування, коментування та спільної роботи в реальному часі.
- InVision: Це інструмент для створення прототипів, який дозволяє створювати інтерактивні прототипи з анімацією та переходами між сторінками. Він також надає можливість спільної роботи над проектами та збору зворотного зв'язку від користувачів.

- Axure RP: Це інструмент для прототипування, який дозволяє створювати складні інтерактивні прототипи зі змінною стану, умовами та динамічними ефектами.
- Marvel: Це онлайн-інструмент для створення прототипів, який дозволяє швидко створювати інтерактивні прототипи та спільно працювати над проектами.

1.2 Призначення розробки та область застосування

Темою бакалаврської дипломної роботи виступає: «Розробка автоматизованої інформаційної системи з продажу книжок.» Головною метою роботи буде створення веб-сайту, який представляє собою книжковий інтернет-магазин під назвою **«bibliopolium.com»**.

Головними критеріями розроблювального веб-сайту є:

- ✓ Зручне використання сайту
- ✓ Швидкий пошук продукту
- ✓ Легкість в освоєнні інтерфесу

Система призначена для:

- ✓ Створення середовища для продажу книг
- ✓ Взаємодії з компонентами сайту
- ✓ Обробка заказів користувачів

1.3 Підстава для розробки

Підставою для розробки веб сайту є те, що в кінці навчання студент виконую кваліфікаційну роботу. Тема роботи була заздалегідь узгоджена з керівником проєкту, випускаючою кафедрою та затверджується наказом ректора.

Отже підставою можна вважати:

- ОКХ та ОПП за напрямом підготовки 6.050101 «Комп'ютерні науки»;
- Графік навчального процесу та навчальний план;
- наказ ректора Національного технічного університету «Дніпровська політехніка» № 350-с від 16.05.2023 р;
- завдання на дипломний проект на тему «Розробка автоматизованої інформаційної системи з продажу книжок».

1.4 Постановка завдання

Метою дипломного проєкта є веб-сайту для книжкового магазина під назвою **«bibliopolium.com»**. Він призначений для зручного пошуку книжок та автоматизованою обробкою замовлень.

Веб-сайт повинен реалізувати такі дії:

- Фільтування списків книжок за вибором.
- Пошукова система за назвою книжки.
- Обробляти замовлення клієнтів та зберігати їх в базі даних phpMyAdmin.
- Можливість додавання нової книжки на сайт.

Для виконання проєкту потрібно:

- Проаналізувати дизайн та реалізацію вже існуючих книжкових інтернет-магазинів.
- Спроєктувати архітектуру системи.
- Визначитися з дизайном та інтерфейсом.
- Реалізувати front-end та back-end визначеної системи.

1.5 Вимоги до програми або програмного вибору

1.5.1 Вимоги до функціональних характеристик.

Сторінки веб-сайту повинні забеспечити такі поставленні цілі:

- Додавання нової книжки на сайт
- Пошук книжки по її назві
- Сторінку обробки заказу книжки
- Перегляд куплених книжок на сайті

1.5.2 Вимоги до складу та параметрів технічних засобів

Для комфортного використання веб-сайту необхідно мати нижче перечислені речі:

- Комп’ютерна миша (тачпад)
- Клавіатура.
- Доступ до онлайн мережі.
- 2гб вільного місця на жорсткому диску або на ssd-диску
- Процесор Intel Core i5-11320H з тактовою частотою 3.2-4.5ГГц.
- Від 8Гб оперативної пам’яті або більше.
- Рідкокришталевий монітор з діагоняллю 15.6", або з розширенням екрану 1920x1080, 1080x1920(смартфон з діагоналлю 6-6,5")

Вище наведені характеристики є рекомандованими для використання веб-сайту і були використовані для його розробки. Частота процесору та об’єм оперативної пам’яті грають роль в швидкості обробки даних та роботи сайту.

1.5.3 Вимоги інформаційної та програмної сумісності.

Для коректної роботи сайту необхідно мати на обчислювальній машині останні версії драйверів та системи, які відповідали б таким вимогам:

- ✓ Веб-браузер Google Chrome або Microsoft Edge.
- ✓ Операційна система Microsoft Windows 10 Professional.

Веб-сайт має бути реалізований на мові програмування **PHP** з використанням базою даних **phpMyAdmin**. Для візуалізації було використано CSS, який описує

увесь дизайн та інтерфейс веб-сторінок. Для хостингу веб-сайту було обрано локальний сервер від розробника OSPanel OpenServer версії 5.2.2.0.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНТЕРНЕТ-МАГАЗИНУ

2.1 Функціональне призначення системи

Під час виконання дипломної роботи було розроблено веб-сайт, а саме інтернет магазин для продажу книжок. Головною метою розробки була можливість надання пошуку та покупки книжок користувачем.

Призначення розробленого інтернет-магазину:

- Можливість пошуку книжок за фільтром.
- Можливість пошуку книжок за пошуковим запитом.
- Можливість перегляду всіх книжок інтернет-магазину.
- Можливість обробки замовлень на покупку.
- Можливість перегляду куплених книжок.

Для досягнення всього перерахованого функіоналу інтернет-магазин повинен мати:

- Сумістність системи з HTML5, x64 розрядністю системи.
- Мати операційну систему Windows 10 Home/Proffessional.
- Мати підтримку сучасних браузерів Google Chrome, Microsoft Edge або Firefox.

2.2 Опис застосованих математичних методів

Розв’язувана задача не передбачає застосування математичних методів. При розробці візуалізації та програмної частини не використовувались ніякі математичні методи.

2.3Опис використаних технологій та мов програмування

Для реалізації інтернет-магазину було використано одну могу програмування. Серверна частина була написана за допомогою MySQL на базі phpMyAdmin. Програмна частина була написана на мові PHP. Візуальна частина була написана на CSS.

Опис мови програмування PHP.

PHP (PHP: Hypertext Preprocessor) є мовою програмування загального призначення, яка використовується для розробки веб-додатків та динамічних веб-сторінок. PHP є однією з найпопулярніших мов програмування для веб-розробки.

PHP було спеціально розроблено для роботи з веб-серверами та генерації динамічного контенту на веб-сторінках. Він може бути вбудований безпосередньо в HTML-код і використовується для створення динамічних елементів, таких як форми зворотного зв'язку, обробники форм, системи керування контентом (CMS) та багато іншого.

PHP має велику кількість вбудованих функцій, які спрощують роботу з базами даних, роботу з файлами та обробку рядків. Він також підтримує різні протоколи мережі, такі як HTTP, FTP і SMTP, що дає можливість взаємодіяти з іншими серверами і сервісами через Інтернет.

PHP є відкритим програмним забезпеченням, що означає, що вихідний код PHP безкоштовно доступний для використання та модифікації. Це також означає, що спільнота розробників активно працює над вдосконаленням мови, виправленням помилок і впровадженням нових функцій.

PHP є скриптовою мовою програмування, тому процес його виконання відрізняється від традиційних компільованих мов, таких як C++ чи Java. Основні етапи обробки PHP-коду включають наступне:

- Кроки перед компіляцією (Pre-compilation): Перш за все, PHP-код проходить попередню обробку (pre-processing). У цьому кроці відбувається інтерпретація директив <?php ?> із файлу, розбиття коду на токени та підготовка його до наступних етапів.

- Компіляція (Compilation): У PHP немає окремого етапу компіляції, як у традиційних мовах. Замість цього, PHP-код інтерпретується напряму. Під час виконання PHP-скрипту, вбудований інтерпретатор PHP (наприклад, Zend Engine) поетапно читає і аналізує PHP-код, виконуючи його рядок за рядком.
- Генерація байт-коду: У процесі інтерпретації PHP-коду, інтерпретатор перетворює кожну інструкцію внутрішнього представлення на байт-код (bytecode). Байт-код є низькорівневим представленням PHP-коду, яке може бути виконане вбудованим інтерпретатором.
- Виконання байт-коду: Отриманий байт-код виконується інтерпретатором PHP-скрипту. Інтерпретатор читає та виконує кожну інструкцію байт-коду по черзі, виконуючи необхідні дії (обчислення, звернення до бази даних, генерацію HTML тощо) на основі змісту кожної інструкції.

Цей процес інтерпретації відбувається в реальному часі під час виконання PHP-скрипту на сервері. Кожного разу, коли PHP-скрипт викликається у браузері або через команду командного рядка, ці кроки повторюються для обробки та виконання коду.

Важливо відзначити, що у PHP є певні оптимізації, такі як кешування байт-коду або використання прискорювачів PHP (наприклад, Zend OPcache), що можуть поліпшити продуктивність веб-додатків, зменшуючи час, необхідний для обробки PHP-коду.

Опис MySQL

MySQL є однією з найпопулярніших відкритих реляційних баз даних, яка широко використовується в розробці веб-додатків та інших програмних системах.

Основними рисами та характеристиками MySQL є:

- MySQL базується на реляційній моделі даних, де дані представлені у вигляді таблиць з рядками (записами) та колонками (полями). Це дозволяє зберігати та організовувати дані у структурований спосіб зі зв'язками між таблицями.
- MySQL використовує стандартну мову запитів SQL (Structured Query Language) для взаємодії з базою даних. SQL надає потужні можливості для створення, зміни та вибірки даних, створення таблиць, індексів, зв'язків та інших об'єктів бази даних.
- MySQL відомий своєю високою швидкодією та продуктивністю. Він добре підходить для обробки великих обсягів даних та високонавантажених додатків. MySQL вміє ефективно виконувати операції пошуку, сортування, фільтрації та об'єднання даних.
- MySQL підтримує різні режими реплікації та кластеризації, що дозволяє масштабувати базу даних та забезпечити високу доступність. Реплікація дозволяє створювати копії бази даних для забезпечення резервного копіювання, шкальованості читання та забезпечення вищої доступності.
- MySQL підтримує широкий спектр функцій та можливостей, таких як транзакції, індекси, тригери, хранилища (stored procedures) та інші. Він також підтримує різні типи даних, включаючи числа, рядки, дати, графічні файли та більш складні типи, такі як JSON та географічні дані.
- MySQL надає механізми для захисту даних, такі як автентифікація та авторизація користувачів, шифрування з'єднань та захист від SQL-ін'єкцій та інших атак.
- MySQL має велику та активну спільноту розробників, що надає підтримку, документацію, форуми та інші ресурси для розв'язання питань та проблем, пов'язаних з використанням MySQL.

Загалом, MySQL є потужною та надійною реляційною базою даних, яка широко використовується в розробці різноманітних веб-додатків та систем.

Опис CSS

CSS (Cascading Style Sheets) є мовою стилів, яка використовується для оформлення та візуального оформлення веб-сторінок і документів, написаних у мовах розмітки, таких як HTML або XML.

Основні риси та характеристики мови стилів CSS:

- CSS використовується для розділення оформлення та стилізації веб-сторінок від їх структури та вмісту. Це дозволяє розробникам окремо керувати виглядом і макетом сторінок, що полегшує підтримку та зміну дизайну.
- CSS працює за принципом каскаду, де стилі можуть бути вказані на різних рівнях (внутрішній аркуш стилів, зовнішній аркуш стилів, інлайнові стилі), і вони можуть спадати від одного рівня до іншого. Це дозволяє створювати загальні стилі, які застосовуються до багатьох елементів, а також точніші стилі, що перевизначають загальні налаштування для конкретних елементів.
- CSS використовує селектори, щоб вибирати конкретні елементи на сторінці, до яких будуть застосовуватись стилі. Селектори можуть бути базовими (наприклад, назва тегу) або деталізованими (наприклад, класи, ідентифікатори або розташування елементів).
- CSS має функціональність медіа-запитів, яка дозволяє адаптувати стилі до різних пристроїв та розмірів екранів. Це дозволяє створювати адаптивний дизайн, який забезпечує оптимальний вигляд та досвід користувача на різних пристроях.
- CSS надає широкий набір властивостей, які можуть бути застосовані до вибраних елементів. Наприклад, ви можете визначити кольори, розміри, шрифти, фони, рамки, положення елементів та багато іншого.
- CSS підтримує анімації та переходи, що дозволяють створювати рухомі ефекти та плавні переходи між станами елементів.

– CSS підтримує можливості розширення за допомогою препроцесорів, таких як Sass або Less, які додають додаткові функції та можливості до мови CSS.

CSS є потужним інструментом для створення привабливого та добре організованого вигляду веб-сторінок. Використовуючи CSS, розробники можуть легко контролювати стилізацію та макет своїх проектів.

2.4 Опис структури системи та алгоритмів її функціонування

Інтернет-магазин був побудований на основі MVC підходу.

MVC (Model-View-Controller) є архітектурним шаблоном, який використовується для організації структури та логіки програмного забезпечення. Він розділяє компоненти програми на три основні ролі: Модель (Model), Представлення (View) та Контролер (Controller).

Модель відповідає за управління даними та бізнес-логікою додатку. Вона представляє справжні дані програми, зберігає їх, виконує операції збереження, оновлення, видалення та інші маніпуляції з даними. Модель незалежна від інтерфейсу користувача і представлення, і може бути використана в різних контекстах.

Представлення відповідає за візуальне відображення даних моделі користувачу. Воно забезпечує інтерфейс користувача, через який користувач може взаємодіяти з додатком. Представлення отримує дані від моделі та відображає їх у відповідному форматі (наприклад, HTML для веб-додатків). Воно також може служити для відображення помилок, повідомлень та іншої інформації для користувача.

Контролер обробляє вхідні дані та керує потоком додатку. Він приймає вхідні дані від користувача через представлення, виконує відповідні дії та взаємодіє з моделлю для отримання або оновлення даних. Контролер вирішує, які дії мають бути виконані відповідно до отриманих даних та викликає відповідні методи

моделі. В результаті він може оновлювати стан моделі або відправляти дані до представлення для відображення.

Основна ідея MVC полягає в розділенні обов'язків між цими трьома компонентами, щоб забезпечити більшу модульність, підтримку змін та повторне використання коду. Крім того, це дозволяє забезпечити відокремлення логіки бізнесу від інтерфейсу користувача, що полегшує розвиток, тестування та підтримку програмного забезпечення.

MVC-архітектура (рис 2.4.1) робить розподіл відповідальностей між компонентами, що дозволяє змінювати окремий компонент окремо, не змінюючи інші. Це робить код більше модульним та підтримуваним, що зменшує витрати на розробку та підтримку ПЗ.

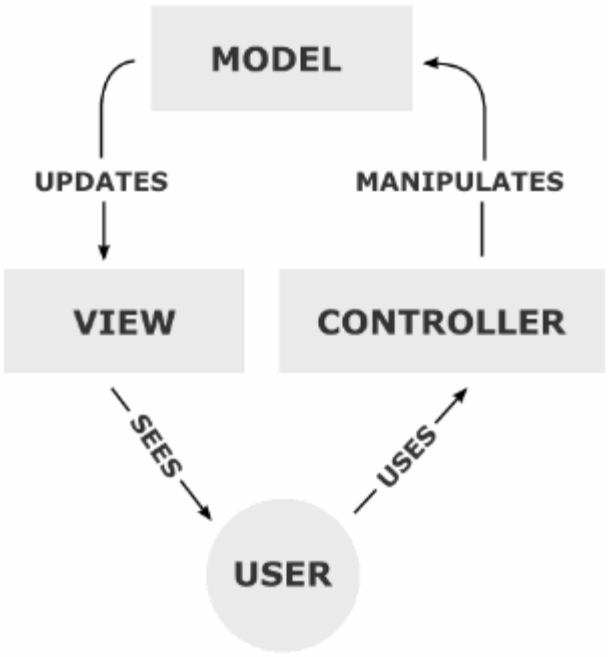

Рис. 2.4.1 – Архітектурний підхід MVC

Розгляд серверної частини проєкту:

Для реалізації серверної частини було використано веб-сервер OpenServer, який надає можливість локально розробляти та тестувати веб-сайти та веб-

додатки на платформі Windows і надає зручне середовище для розробки на PHP, MySQL, Apache та інших технологіях.

OpenServer надає простий інсталятор, який дозволяє встановити веб-сервер з необхідними компонентами, такими як PHP, MySQL та інші, за декілька кроків. OpenServer має можливість встановити його на USB-пристрої та на фізичні носії, такі як SSD-диски та жорсткі диски.

Його функціонал можна побачити на рис. 2.4.2-2.4.5.

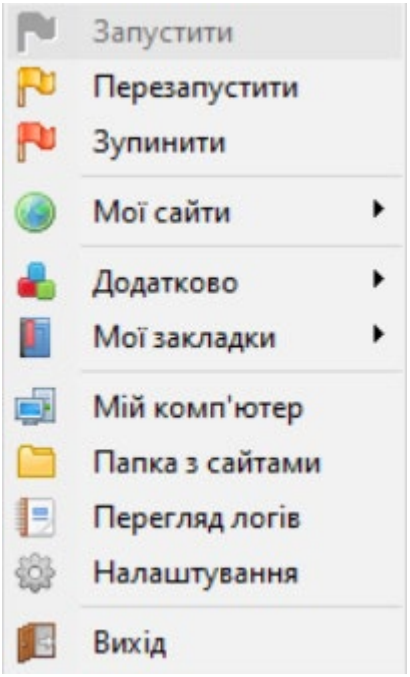

Рис. 2.4.2 – Спливаюче вікно основних компонентів.

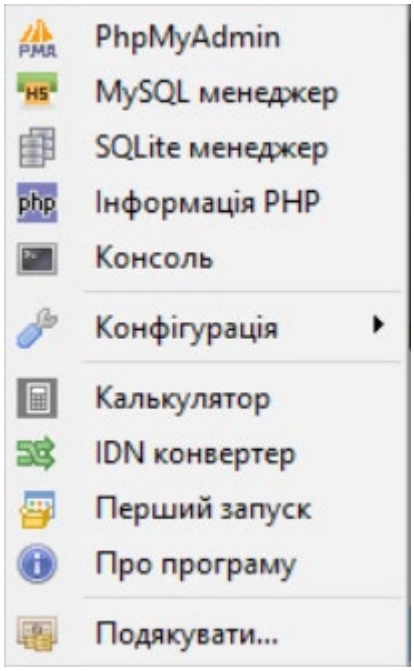

Рис. 2.4.3 – Вкладка «додатково».

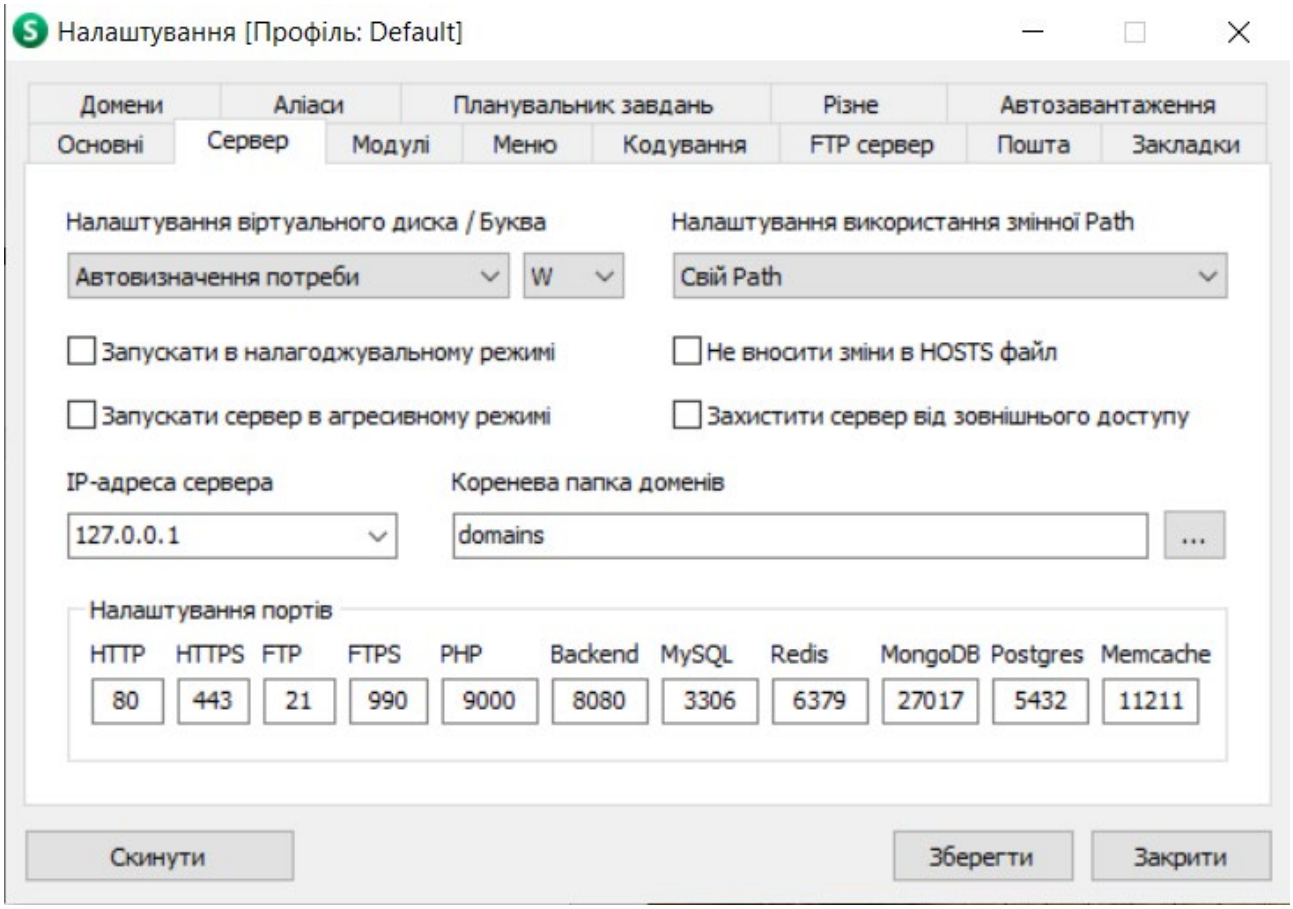

Рис. 2.4.4 – Вкладка «Налаштування».

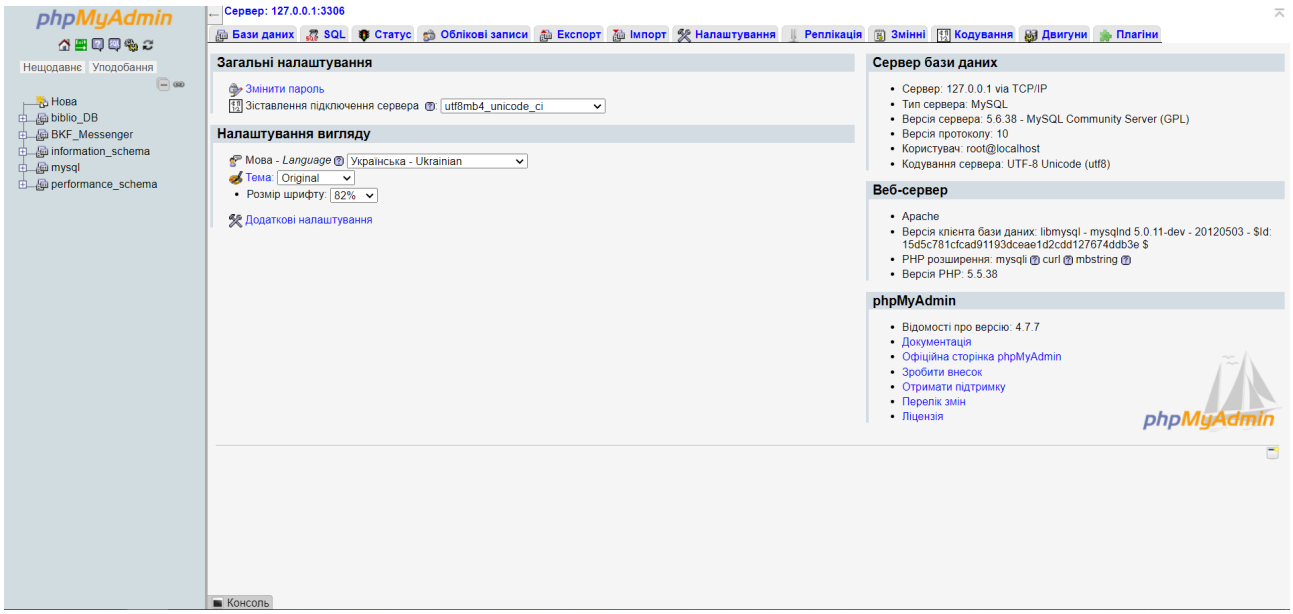

Рис. 2.4.5 – Вікно налаштування таблиць та баз даних на мові MySQL.

MySQL є реляційною базою даних. Реляційна база даних (Relational Database) - це тип бази даних, який зберігає дані у вигляді таблиць зі зв'язками між ними. Вона ґрунтується на реляційній моделі даних, яка була розроблена Едгаром Коддом у 1970-х роках.

Дані у реляційній базі даних організовані у вигляді таблиць. Кожна таблиця має назву та складається з рядків (записів) і стовпців (атрибутів). Кожен стовпець визначає певний тип даних, наприклад, ціле число, рядок, дата тощо.

Реляційна база даних використовує зв'язки між таблицями для встановлення залежностей між даними. Зв'язки можуть бути один-до-одного (one-to-one), один-до-багатьох (one-to-many) або багато-до-багатьох (many-to-many). Вони дозволяють зв'язувати дані з однієї таблиці з даними з інших таблиць, що робить реляційну базу даних гнучкою та потужною.

Ключі використовуються для унікальної ідентифікації записів у таблицях. Основним ключем (Primary Key) є унікальний ідентифікатор кожного запису у таблиці. Зовнішній ключ (Foreign Key) використовується для створення зв'язку між таблицями, посилаючись на основний ключ іншої таблиці.

За допомогою мови запитів, такої як SQL (Structured Query Language), можна виконувати різноманітні операції з даними, такі як вибірка, вставка, оновлення і видалення. Запити дозволяють отримувати потрібну інформацію з бази даних, виконувати фільтрацію, сортування та об'єднання даних за певними умовами.

Реляційні бази даних часто піддаються процесу нормалізації, який допомагає уникнути дублювання даних та забезпечує їх консистентність та ефективне використання. Нормалізація розділяє дані на окремі таблиці, щоб зменшити повторення даних та забезпечити правильність зв'язків між ними.

Реляційні бази даних широко використовуються у багатьох галузях, включаючи веб-розробку, банківські системи, управління запасами та багато інших. Вони надають структурований та організований підхід до зберігання та управління даними, що спрощує роботу з ними та забезпечує консистентність та цілісність інформації.

В реляційній моделі баз даних існує Кілька нормалних форм. Розглянемо основні із них:

- Перша нормальна форма: Вимагає, щоб кожен атрибут у таблиці мав атомарне значення, тобто не містив повторюваних значень або групувався в масиви.

- Друга нормальна форма: Вимагає, щоб кожен неключовий атрибут у таблиці залежав від усього первинного ключа, а не від його окремих частин.
- Третя нормальна форма: Вимагає усунення транзитивних залежностей між неключовими атрибутами у таблиці. Це означає, що якщо атрибут A залежить від атрибута B, а атрибут B залежить від атрибута C, то атрибут A повинен залежати від атрибута C безпосередньо.
- Четверта нормальна форма: Вимагає розкладу таблиці для уникнення множинних залежностей та залежностей між неключовими атрибутами.
- П’ята нормальна форма: Вимагає розкладу таблиці для уникнення залежностей між неключовими атрибутами, що базуються на функціональних залежностях.

Крім цих нормальних форм, існують також інші розширені форми, такі як шоста нормальна форма (6NF) та нормальна форма Дейта-Кордова (DKNF), які застосовуються в деяких специфічних ситуаціях.

Важливо зауважити, що нормалізація бази даних залежить від конкретного контексту і вимог проекту, і може варіюватись в залежності від конкретних сценаріїв використання та потреб користувачів.

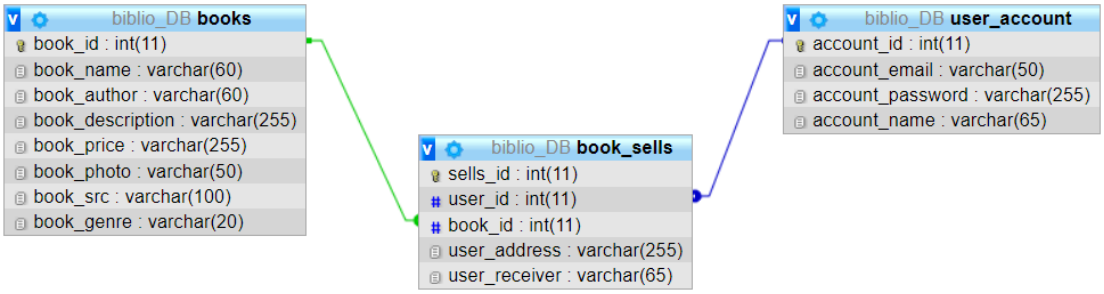

Рис. 2.4.6 – ER діаграма бази даних

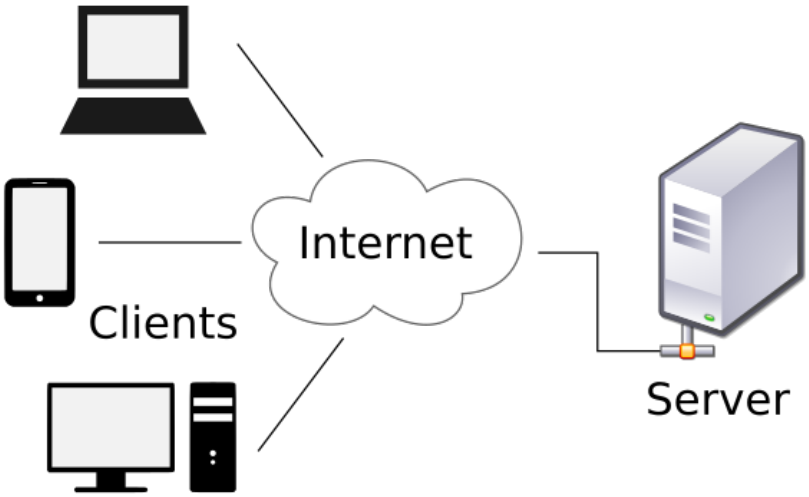

Рис. 2.4.7 – Алгоритм роботи серверної частини

В роботі над проєктом була обрана клієнт-серверна архітектура.

Клієнт-серверна архітектура - це модель розподіленої системи, де задачі та відповідальності розподіляються між клієнтами (користувачами) та серверами. У цій архітектурі клієнт ініціює запити до сервера, який надає відповіді та обробляє запити клієнта.

Клієнт - це кінцевий користувач або програма, яка взаємодіє з сервером для отримання певної послуги або ресурсу. Клієнтська сторона може бути реалізована у вигляді веб-браузера, мобільного додатка, настільної програми або будь-якої іншої програми, яка взаємодіє з сервером.

Сервер - це комп'ютер або система, яка надає послуги або ресурси клієнтам. Сервер зазвичай має вищу потужність та можливості для обробки запитів та забезпечення доступу до даних або послуг. Він слідкує за запитами клієнтів, обробляє їх та надає відповіді. Клієнт та сервер використовує HTTP протокол передачі даних.

HTTP (Hypertext Transfer Protocol) - це протокол передачі гіпертексту, який використовується для комунікації між клієнтами та серверами в Інтернеті. Він є основою для обміну даними між веб-браузерами та веб-серверами, де веб-браузер виступає як клієнт, а веб-сервер - як сервер.

HTTP використовує модель "клієнт-сервер", де клієнт ініціює запит до сервера, а сервер надає відповідь на цей запит. Запит та відповідь складаються з різних повідомлень, які включають заголовки та необов'язкове тіло повідомлення.

Основні методи HTTP-запитів включають:

- GET: Запит на отримання ресурсу з сервера. Використовується для отримання веб-сторінок, зображень, даних тощо.
- POST: Використовується для надсилання даних до сервера, зазвичай для створення нового ресурсу або виконання деякої операції на сервері.
- PUT: Використовується для оновлення або заміни існуючого ресурсу на сервері.
- DELETE: Використовується для видалення ресурсу на сервері.
- PATCH: Використовується для часткового оновлення ресурсу на сервері.

Крім методів запитів, HTTP визначає коди статусу, які повертаються у відповіді сервера для позначення стану запиту (наприклад, 200 OK для успішного запиту, 404 Not Found для недоступного ресурсу).

HTTP забезпечує простий і стандартизований спосіб комунікації між клієнтом та сервером, що дозволяє передавати різноманітні дані, включаючи гіпертекстові документи, зображення, відео, аудіо, API-запити та інші типи ресурсів.

2.5 Обгрунтування та організація вхідних та вихідних даних програми

Веб-сайт отримує вхідні дані із заповнених користувачем форм на сторінках сайту та обробляє їх у PHP коді за методом POST і відправляє їх в базу даних MySQL.

Вихідні дані клієнт бачить після того, як дані були взяті із бази даних за його ідентифікаційним номером або назвою.

2.6 Опис роботи розробленої системи.

2.6.1 Використані технічні засоби.

Для розробки цього проєкту було використано таке обладнання:

- Процесор Intel Core i5-11320H 4.5GHz.
- Оперативна пам’ять SODIM DDR4 16Gb 3200Mhz.
- Вільне місце на SSD-диску 2Gb.
- Відеокартка Nvidia GeForce GTX1650 GDDR6 1780MHz
- Монітор 15.6 дюймів та розширенням 1980x1080.
- Маніпулятор «миша».
- Клавіатура.
- Вихід до глобальної інтернет-мережі.

Дані характеристики були описані як оптимальні і швидкодійні. На характеристиках нижче або вище можуть бути різний час відгуку на сайті або сервері.

2.6.2 Використані програмні засоби.

Веб-сайт реалізований за допомогою мови PHP та каскадної мови CSS.
Необхідні програмні засоби:

- Веб-браузер на базі Google Chrome.
- Операційна система Windows 10.
- Для зберігання даних має бути онлайн або локальний веб-сервер.
- Редактор коду для подальшого редагування сторінок або додавання нових функцій.

2.6.3 Виклик та завантаження програм

Розроблений веб-сайт для його експлуатації потребує веб-браузер з підтримкою HTML5, PHP та JavaScript.

2.6.4 Опис Інтерфейсу користувача

Перше що побаче користувач, це головну сторінку інтернет-магазину (рис2.6.1).

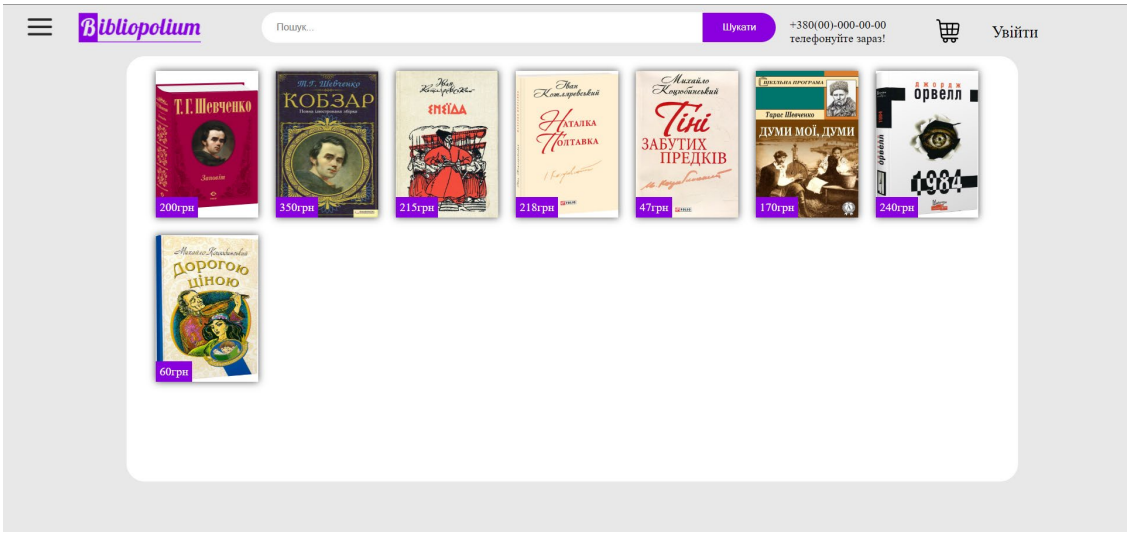

Рис 2.6.1 – Головна сторінка інтернет-магазину.

На головній сторінці користувач бачить так званий “burger-menu”, який знаходиться зліва від логотипу сайту та надає можливість обрати фільтр пошуку(рис 2.6.2), логотип сайту, який дає можливість повернутися на головну сторінку, пошукове поле, в якому буде зручно шукати книги за назвою та їх авторів, номер телефону, по якому можна тримати зворотній зв’язок з власником сайту, список усіх книг з їх ціною та поле входу. «Іконка» корзини має декоративний намір.

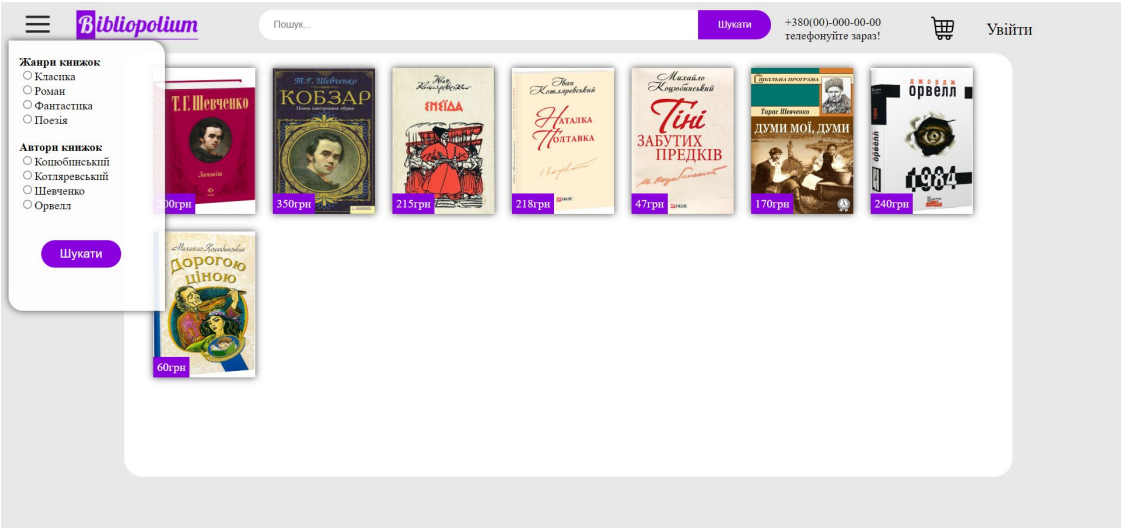

Рис. 2.6.2 – пошуковий фільтр за назвою або автором.

Інтернет-магазин має сіро-білий із ставками фіолетового кольору з закругленими кінцями дизайн. Дизайн було обрано після перегляду інших українських книжкових інтернет-магазинів, таких як yakaboo.ua, book-ye.com.ua, knigoland.com.ua та інші.

Коли користувач знайшов книгу на головній сторінці або за пошуковим запитом, він може подивитись опис книги та замовити її онлайн у фізичному форматі(рис 2.6.3-рис2.6.9).

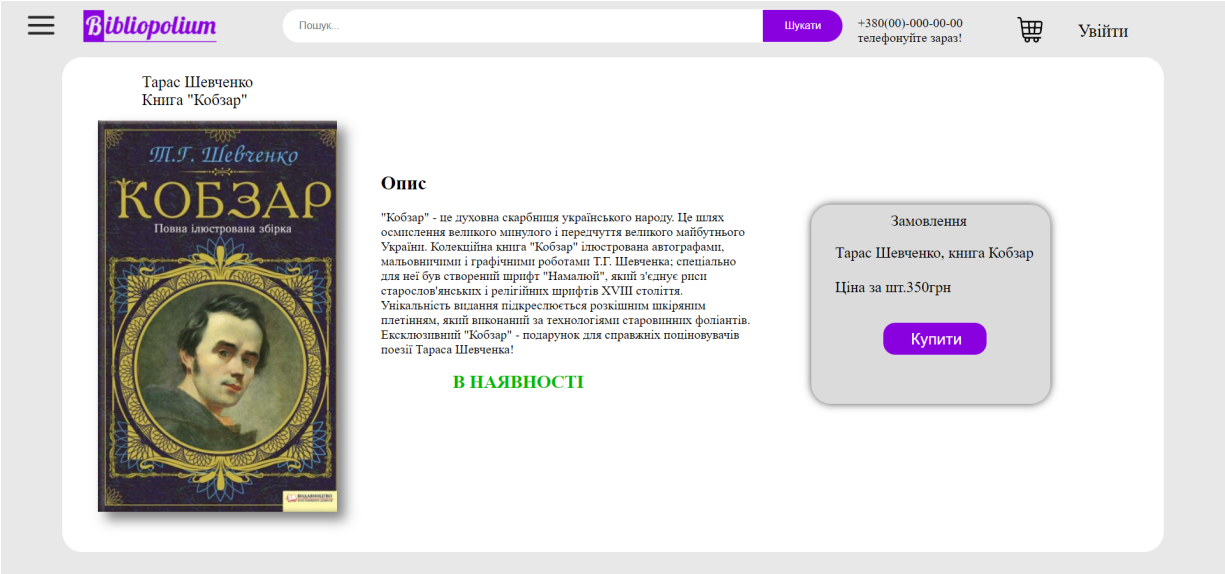

Рис. 2.6.3 – сторінка книги.

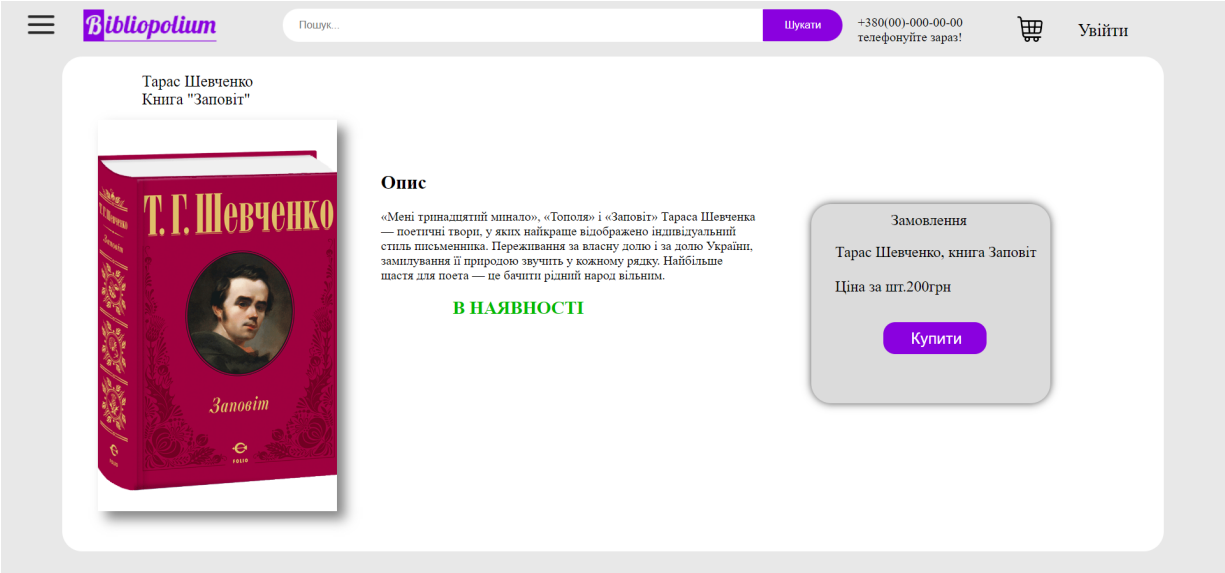

Рис. 2.6.4 – сторінка книги.

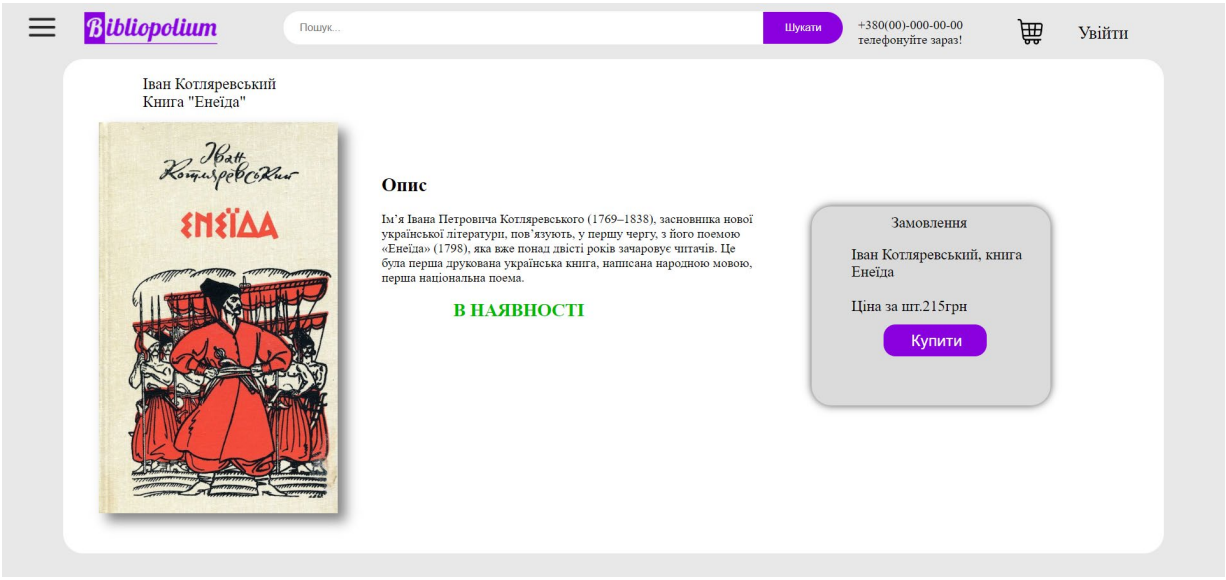

Рис. 2.6.5 – сторінка книги.

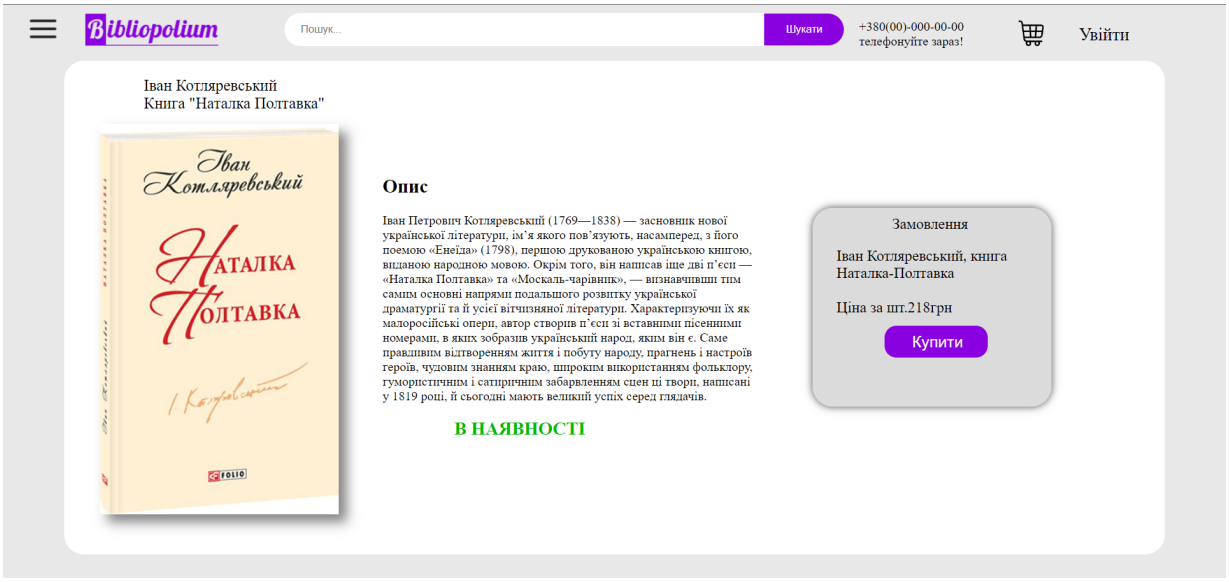

Рис. 2.6.6 – сторінка книги.

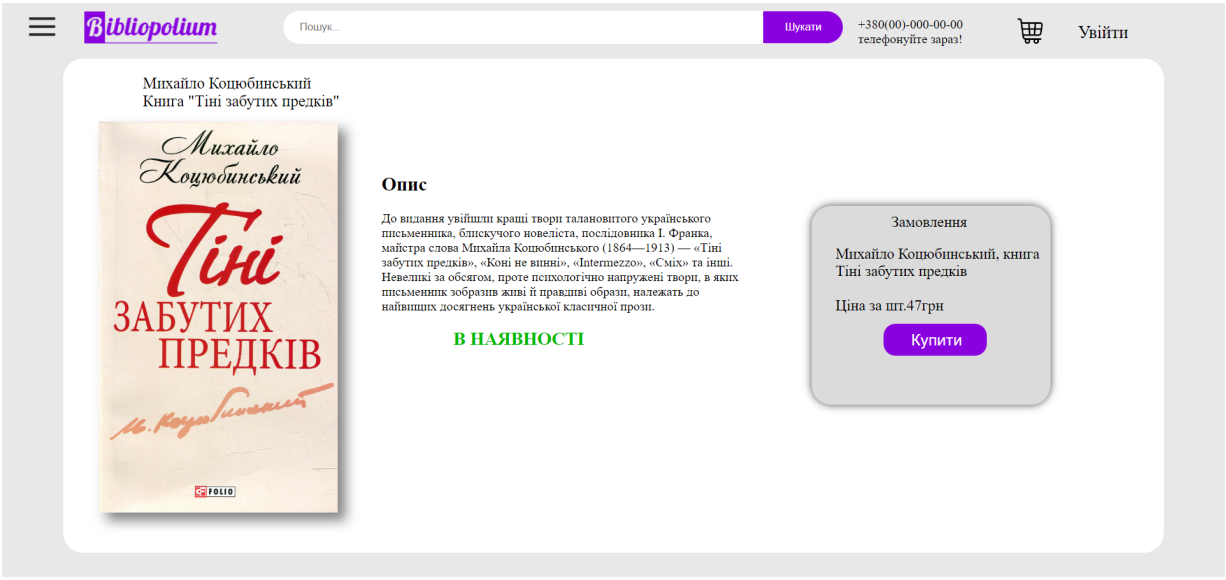

Рис. 2.6.7 – сторінка книги.

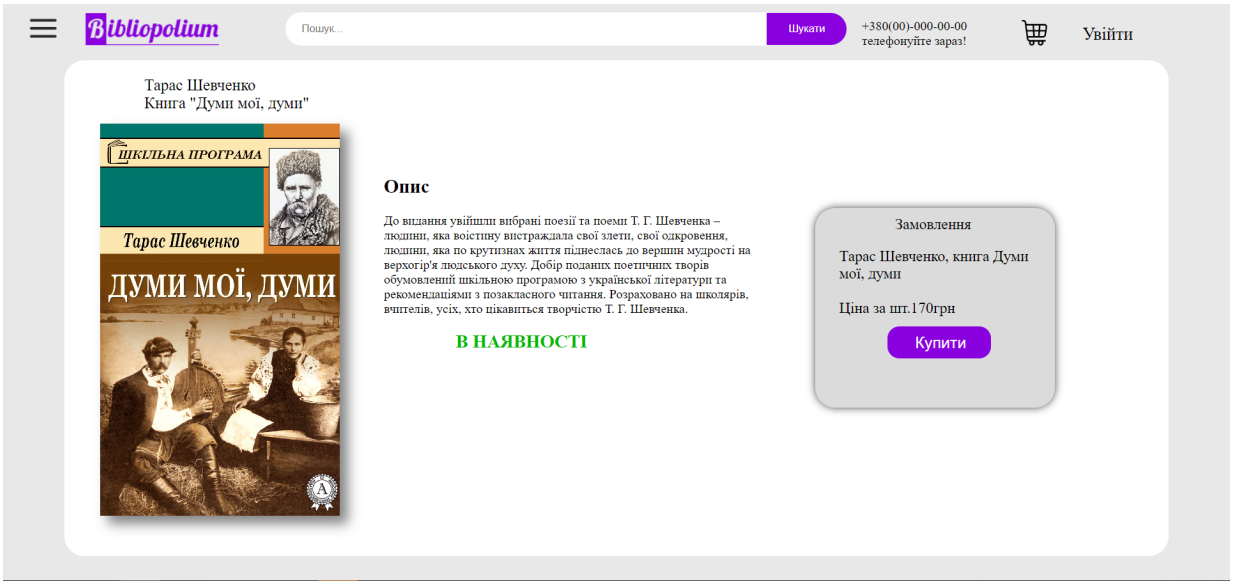

Рис. 2.6.8 – сторінка книги.

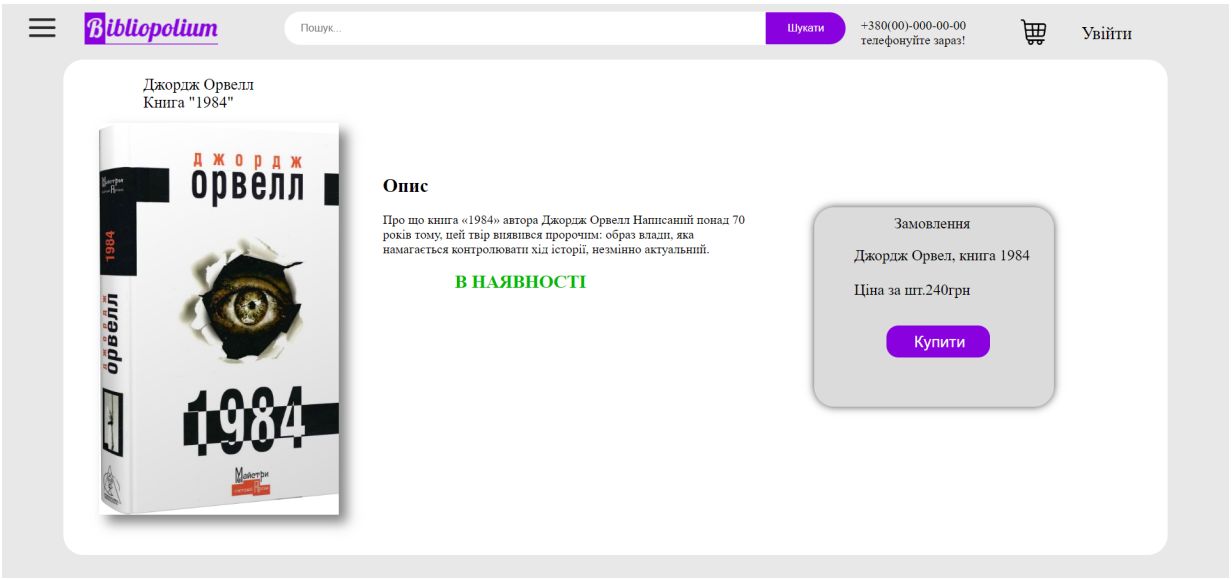

Рис. 2.6.9 – сторінка книги.

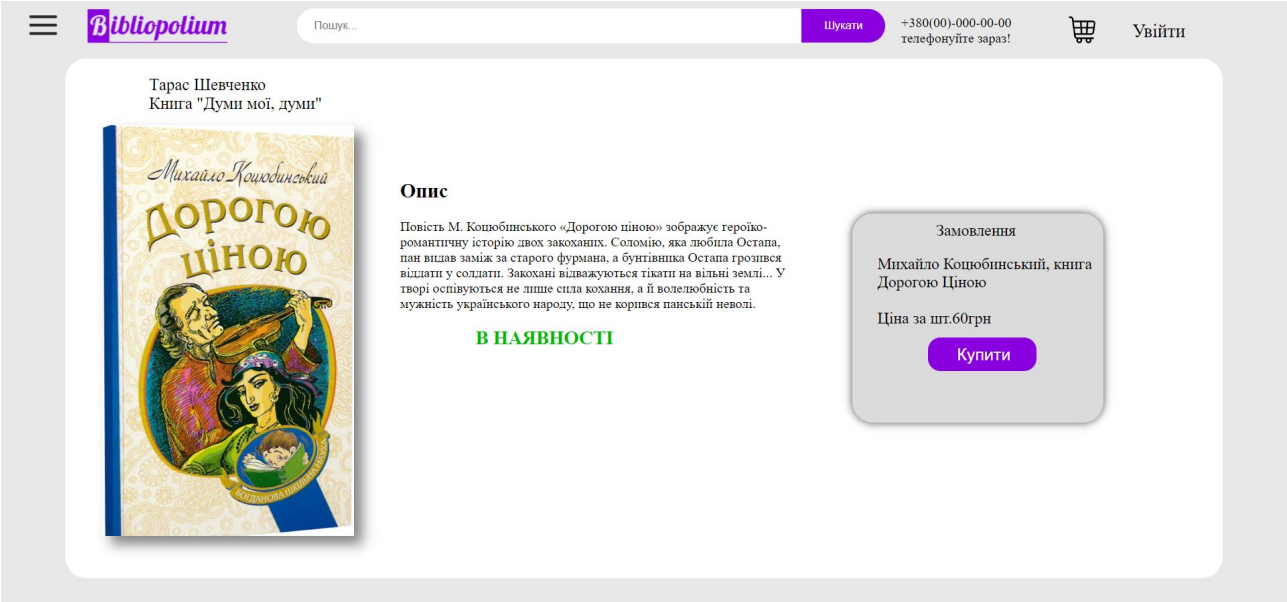

Рис. 2.6.9 – сторінка книги.

Після вибору книги користувач може замовити її при натисканні на кнопку «Купити».Після чого користувача направить до заповненя форми обробки замовлення(рис 2.6.10).

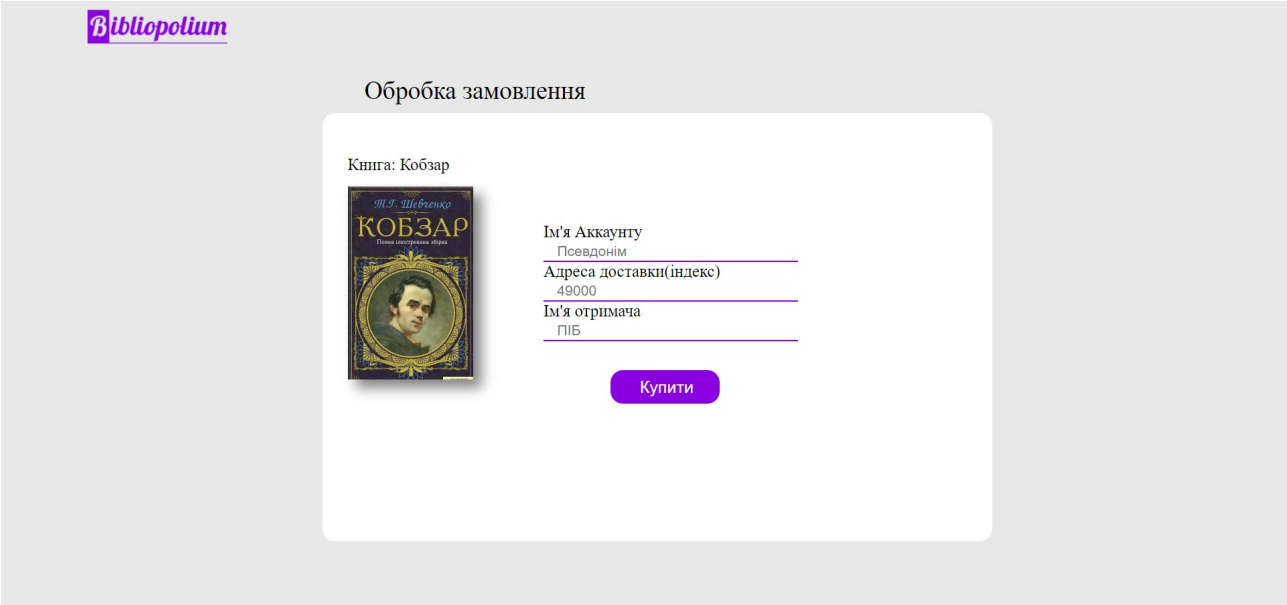

Рис. 2.6.10 – Обробка замовлення.

Якщо користувач ввів не всі дані або не існуюче ім’я аккаунту, тоді при перевірці воно видає такий текст(рис. 2.6.11).

Рис. 2.6.11 – Введені не всі дані або невірне ім’я аккаунту.

Якщо користувач ввів всю дані правильно(рис 2.6.12), тоді він побачить текст іншого характеру(рис 2.6.13).

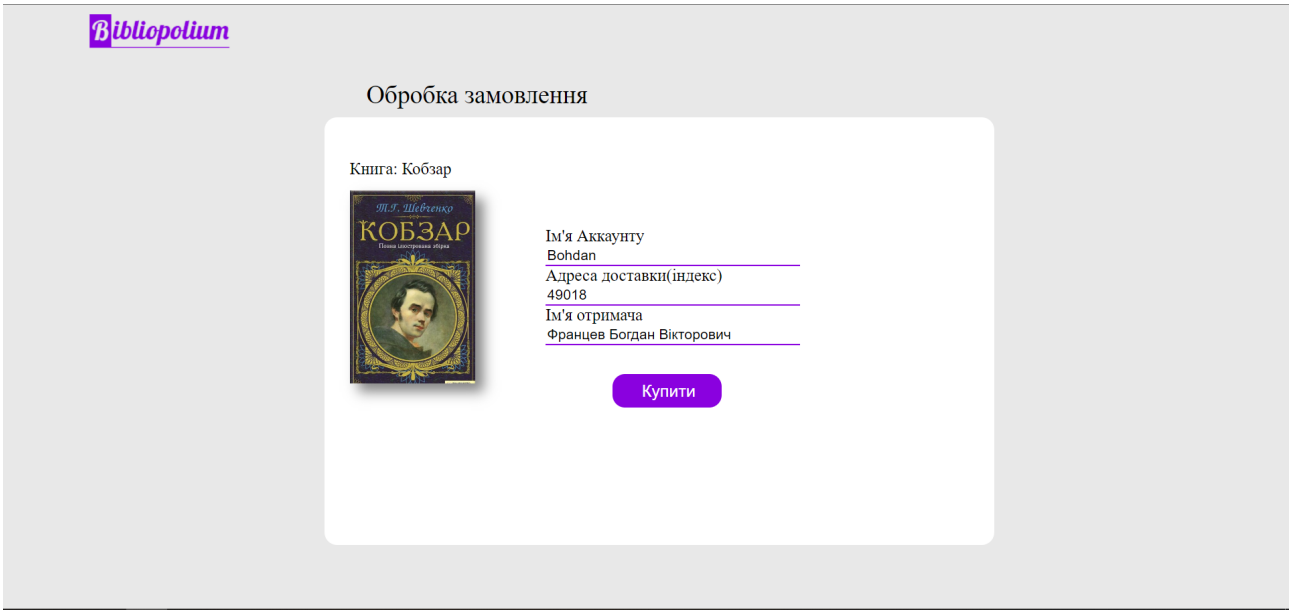

Рис.2.6.12 – Введення даних.

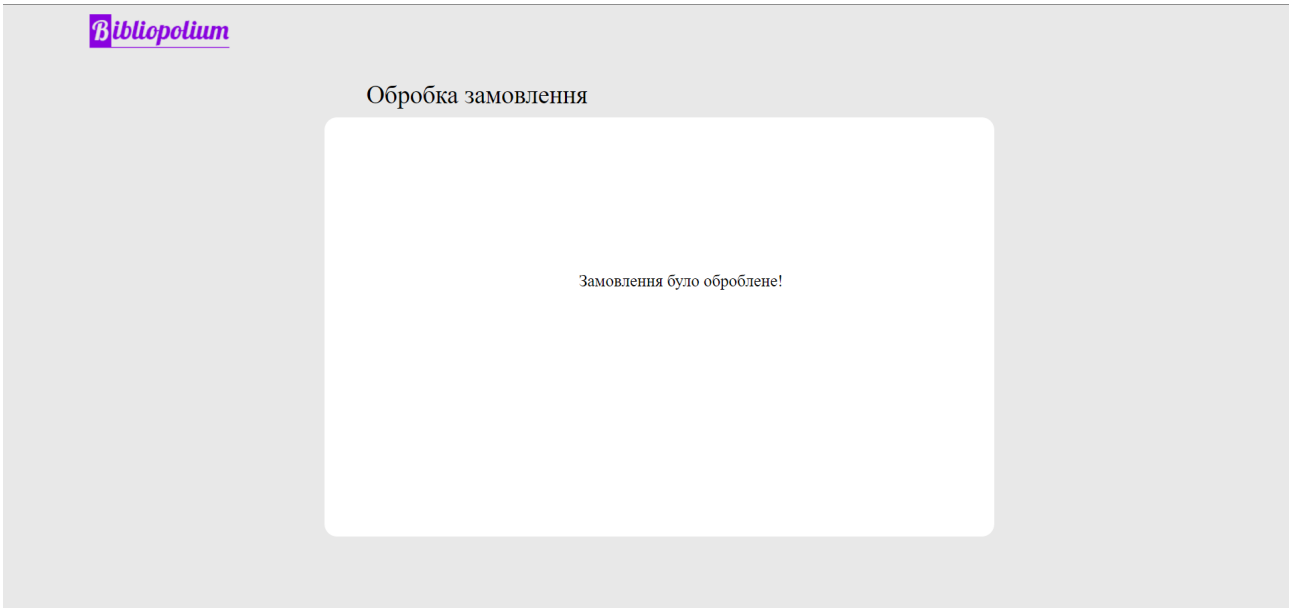

Рис. 2.6.13 – Дані збережено успішно.

Що б перевірити замовлення, користувач може увійти в аккаунт через основну сторінку реєстрації(рис.2.6.14). Якщо користувач ще не зареєструвався, він може зробити це за посиланням(рис.2.6.15).

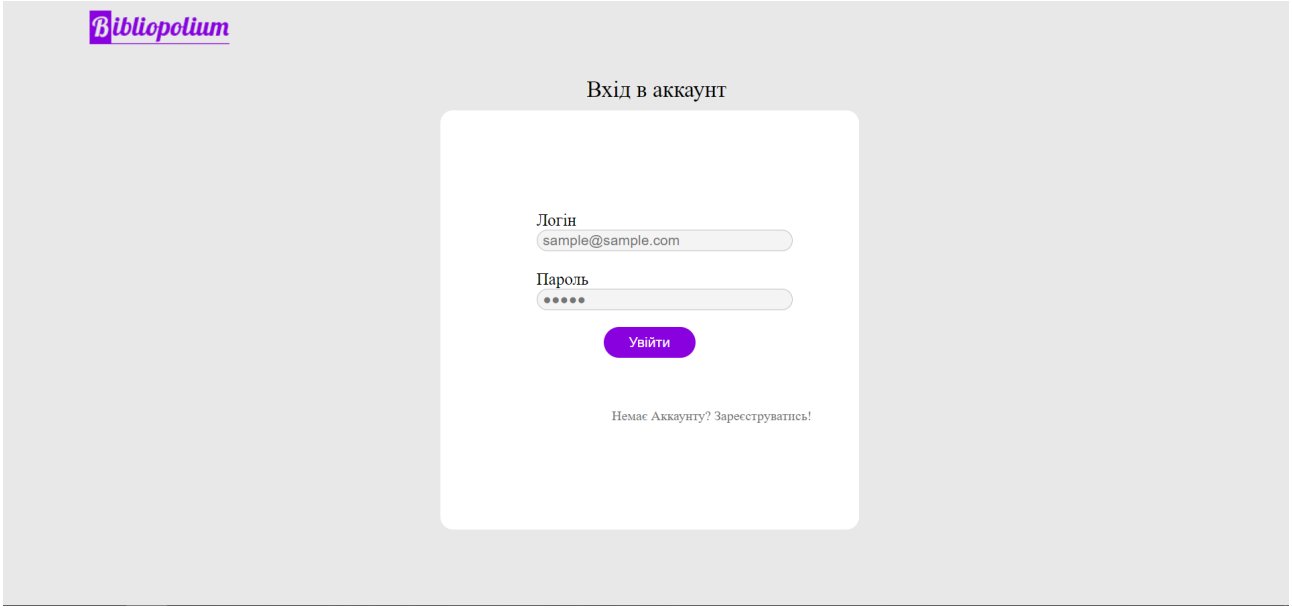

Рис.2.6.14 – Вхід до аккаунту.

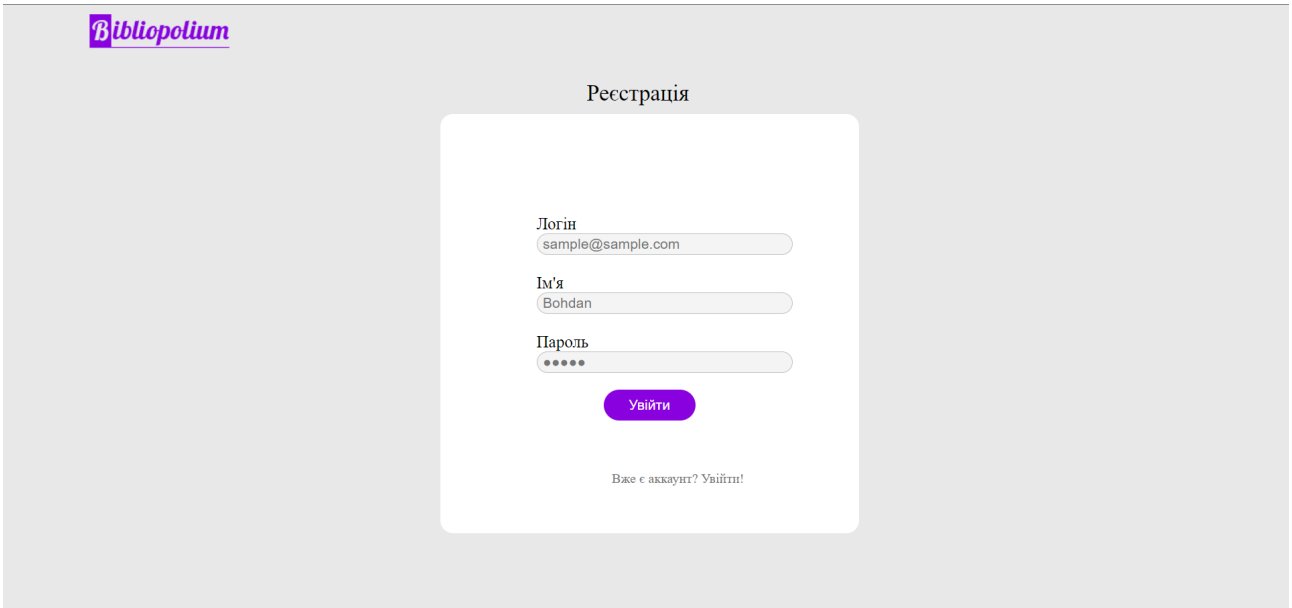

Рис.2.6.15 – Реєстрація на сайті.

У двох випадках зроблена перевірка на пустоту або не правильно введені дані при вході.

Якщо вхід було виконано успішно, тоді користувач побачить свої замовлення(рис.2.6.16). Редагування або видалення замовлень робить вручну розробник проєкту, якщо є така необхідність, користувач в такому випадку має подзвонити по номеру вказаному на головній сторінці.

Рис.2.6.17 – Список куплених книжок.

Демонстрація роботи пошукового фільтру та пошукового поля(Рис2.6.18-2.6.19).

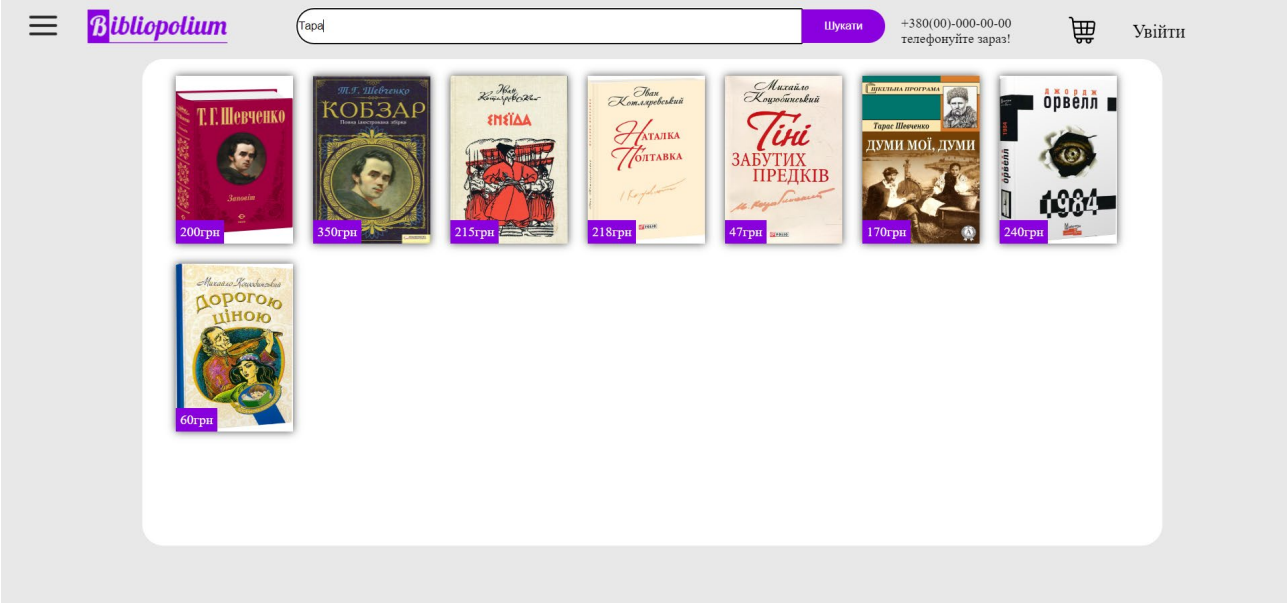

Рис.2.6.18 – Введення пошукового запросу.

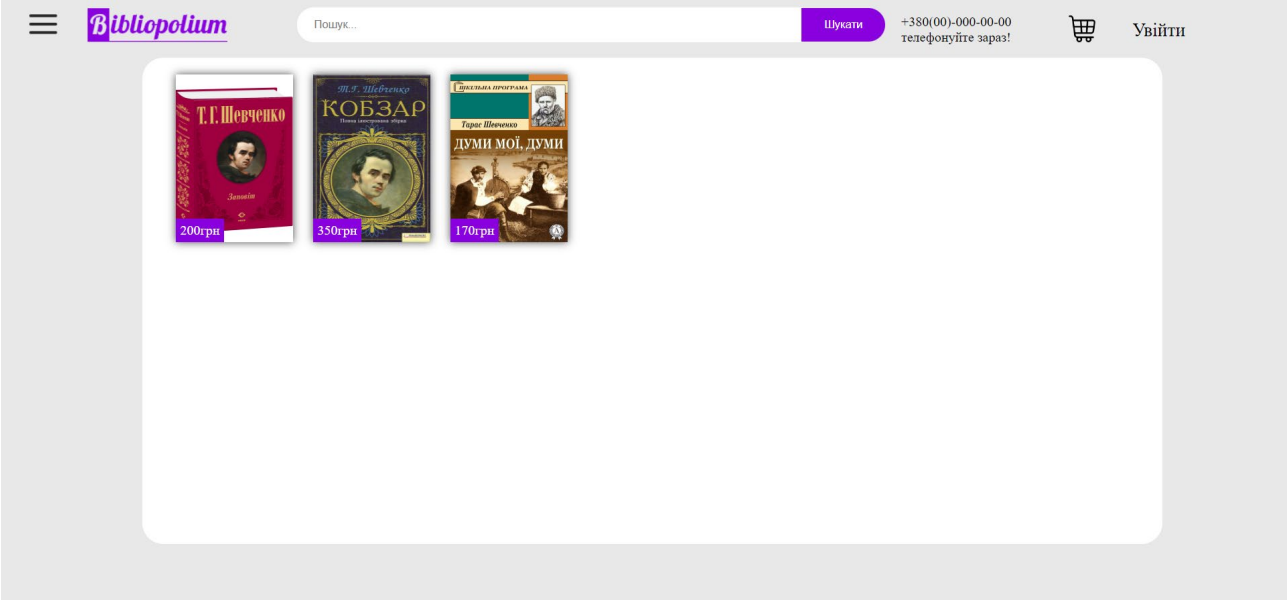

Рис.2.6.19 Виведення знайденої інформації.

Для прикладу роботи фільтру було обрано пункт «Поезія»(рис.2.6.20-рис.2.6.21).

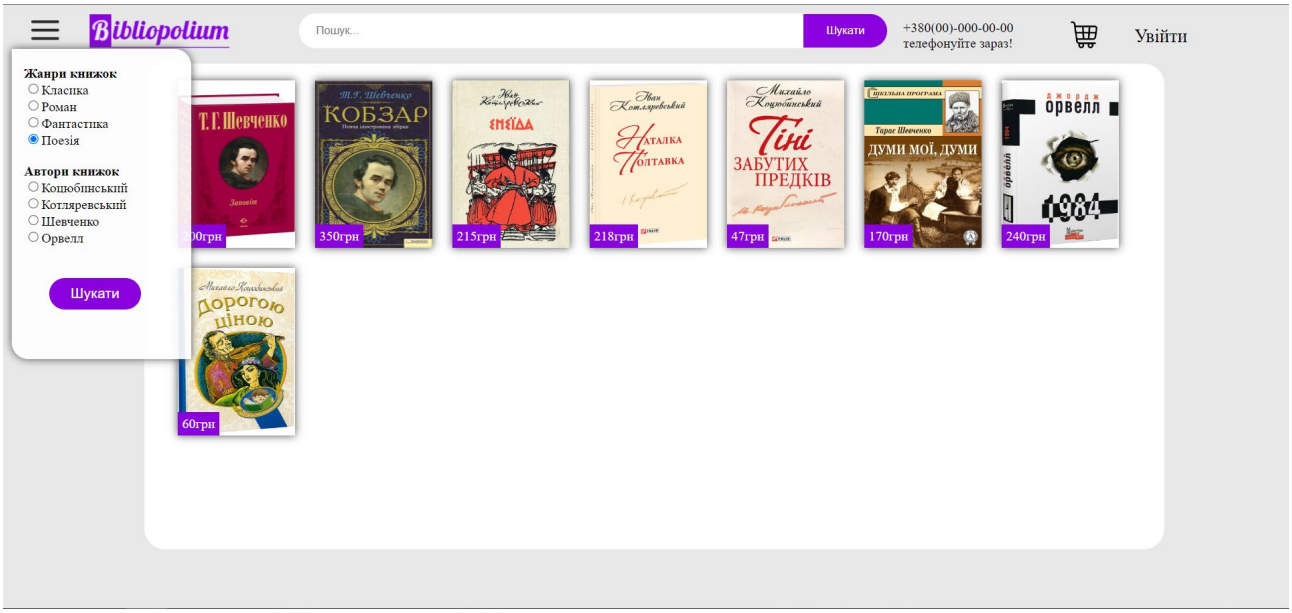

Рис. 2.6.20 – Робота пошукового фільтру.

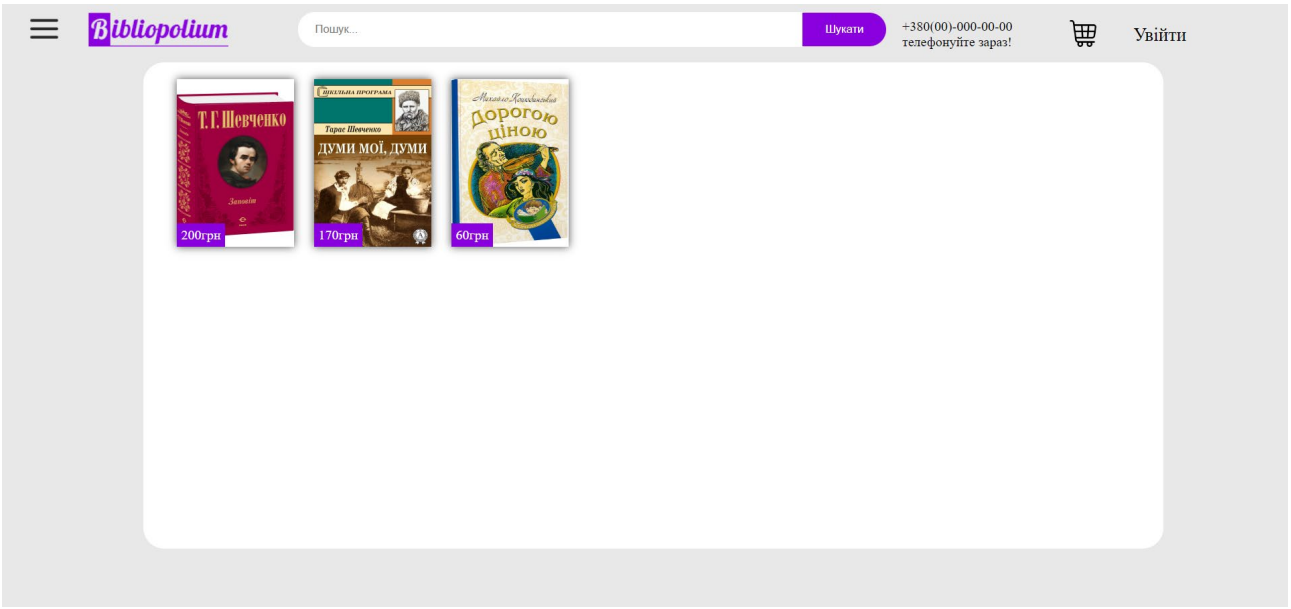

Рис. 2.6.21 – Результат виведення.

РОЗДІЛ 3. ЕКОНОМІЧНА ЧАСТИНА

В економічній частині було розраховано трудомісткість розробки і розрахунок витрат на створення.

3.1 Визначення трудомісткості розробки програмного забезпечення.

Задані дані:

1. Передбачуване число операторів – 100;
2. Коефіцієнт складності програми – 1,7;
3. Коефіцієнт корекції програми в ході її розробки – 0,07;
4. Годинна заробітна плата програміста – 209,43 грн;

Годинна заробітна плата full-stack розробника була вирахувана виходячи з даних «Української спільноти програмістів(DOU)». Середня заробітна плата такого розробника в Україні з досвідом роботи близько року дорівнює 1000 доларів США у місяць. При курсі валют НБУ на кінець червня складає приблизно 36,86 грн за 1 долар США, тому середня плата на місяць дорівнює 36 860 грн. Якщо брати восьмигодинний робочий графік(176 робочих годин на місяць в середньому) середня заробітна плата буде 209,43 грн.

5. Коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,2;
6. Коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 1,2;
7. Вартість машинно-години ЕОМ – 15 грн/год.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{отл} + t_д \text{, людино-годин,} \quad (4.1)$$

де t_o- витрати праці на підготовку й опис поставленої задачі (приймається 50 людино-годин);

t_u - витрати праці на дослідження алгоритму рішення задачі;

t_a- витрати праці на розробку блок-схеми алгоритму;

t_n-витрати праці на програмування по готовій блок-схемі;

$t_{отл}$-витрати праці на налагодження програми на ЕОМ;

$t_д$ - витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у програмногму забезпеченні, яке розробляється.

Умовне число операторів:

$$Q = q \cdot C \cdot (1 + p), \quad (4.2)$$

де q - передбачуване число операторів (100);

C - коефіцієнт складності програми (1,7);

p - коефіцієнт корекції програми в ході її розробки (0,07).

$Q = 100 \cdot 1.7 \cdot (1 + 0{,}07) = 181{,}9$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q \cdot B}{(75..85) \cdot k}, \text{ людино-годин}, \quad (3.3)$$

де B – коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

k – коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності.

$t_u = \frac{181{,}9 \cdot 1{,}2}{(75) \cdot 1{,}3} = 2{,}23$ людино-годин

Витрати праці на розробку алгоритму рішення задачі:

$$t_a = \frac{Q}{(20...25) \cdot k}, \text{ людино-годин} \quad (3.4)$$

$$t_a = \frac{181,9}{20 \cdot 1,3} = 6,99 \text{ людино-годин}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20...25) \cdot k} \text{ людино-годин}, \quad (3.5)$$

$$t_n = \frac{181,9}{25 \cdot 1,3} = 5,59 \text{ людино-годин}$$

Витрати праці на налагодження програми на ЕОМ:

– за умови автономного налагодження одного завдання:

$$t_{отл} = \frac{Q}{(4..5) \cdot k}, \text{ людино-годин} \quad (3.6)$$

$$t_{отл} = \frac{181,9}{5 \cdot 1,3} = 27,9 \text{ людино-годин}$$

– за умови комплексного налагодження завдання:

$$t^k_{отл} = 1,5 \cdot t_{отл}, \text{ людино-годин}, \quad (3.7)$$

$t^k_{отл} = 1,5 \cdot 27,9 = 41,85$ людино-годин.

Витрати праці на підготовку документації:

$$t_д = t_{др} + t_{до}, \text{ людино-годин,} \quad (3.8)$$

де $t_{др}$ – трудомісткість підготовки матеріалів і рукопису;

$$t_{др} = \frac{Q}{15..20 \cdot k}, \text{ людино-годин,} \quad (3.9)$$

$$t_{др} = \frac{181,9}{20 \cdot 1,3} = 6,99 \text{ людино-годин.}$$

$t_{до}$ – трудомісткість редагування, печатки й оформлення документації:

$$t_{до} = 0,75 \cdot t_{др}, \text{ людино-годин,} \quad (3.10)$$

$$t_{до} = 0,75 \cdot 6,99 = 5,24 \text{ людино-годин,}$$

$$t_д = 6,99 + 5,24 = 12,23 \text{ людино-годин.}$$

Тепер розрахуємо трудомісткість ПЗ:

$t = 2,23 + 6,99 + 5,59 + 41,85 + 5,24 + 12,23 = 74,13$ людино-годин.

3.2 Витрати на створення програмного забезпечення

Витрати на створення ПЗ *Кпо* включають витрати на заробітну плату виконавця програми *Зз/п* і витрат машинного часу, необхідного на налагодження програми на ЕОМ:

$$К_{по} = З_{зп} + З_{мв}, \text{ грн.} \quad (3.11)$$

Заробітна плата виконавців визначається за формулою:

$$З_{зп} = t \cdot С_{пр}, \text{ грн,} \quad (3.12)$$

де t – загальна трудомісткість, людино-годин;

$С_{пр}$ – середня годинна заробітна плата програміста, грн/година.

$З_{зп} = 74,13 \cdot 209,43 = 15,525$ грн.

Вартість машинного часу, необхідного для налагодження програми:

$$З_{мв} = t_{отл} \cdot C_{мч}, \text{ грн,} \quad (3.13)$$

де $t_{отл}$ – трудомісткість налагодження програми на ЕОМ, год,

$C_{мч}$ – вартість машино-години ЕОМ, грн/год,

$C_{мч} = 15$, грн/год.

$З_{мв} = 41,85 \cdot 15 = 630$, грн.

Визначені в такий спосіб витрати на створення програмного забезпечення є частиною одноразових капітальних витрат на створення ПЗ:

$К_{по} = 15,525 + 630 = 16\ 155$, грн.

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p}, \text{ міс.,} \quad (3.14)$$

де B_k – число виконавців (приймається 1),

F_p – місячний фонд робочого часу (40 годин на тиждень F_p = 176 годин).

$$Т = \frac{74,13}{1 \cdot 176} = 0,4 \text{ міс.}$$

Висновок: Даний веб-сайт має вартість 16 155грн. Ймовірний час розробки – 0,4 місяця при стандартному 40-годинному робочому тижні і 176-годинному робочому місяці. Термін розробки включає час для дослідження, розробки алгоритму, розв’язання задаі та розробки дизайну сайту. На розробку веб-сайту буде витрачено 74,13 людинно-годин.

ВИСНОВКИ

В дипломній роботі був реалізований інтернет-магазин, спрямований для продажу книжок. Під час розробки було використано мову програмування PHP, каскадна мова CSS, HTML код та база даних MySQL.

Основні функції розробленого інтернет-магазину включають пошук за допомогою фільтру пошуку та пошукового рядка, обробки замовлень, реєстраціїї та входу та перегляд куплених книжок.

В результаті дослідження та розробки було створенно інтернет-магазин, який задовольняє потребам користувачів. Технології, які були використані в дипломному проєкті дозволяють забеспечити ефективну та зручну роботу з веб-сайтом.

Застосування мови PHP допомогло реалізувати багато програмних рішень у цій роботі. Час створення застосунку становить 0,4 місяці, тобто 2 робочої неділі. Вартість проєкта становить 16 155грн.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Документація PHP - https://www.php.net/manual/ru/index.php
2. Документація MySQL - https://dev.mysql.com/doc/
3. Посібник для розробників.MDN - https://developer.mozilla.org/en-US/
4. Документація OpenServer - https://ospanel.io/docs/
5. М.Ю. Швець, Д.С. Заруба, Ю.В. Хохлов Порівняння SQL та NOSQL баз даних 5с.
6. Д. Б. Буй, А.В. Пузікова Теорія нормалізації в реляційних базах даних: сучасний стан 10с.
7. Документація «Як підключати БД в PHP» - https://www.php.net/manual/ru/mysqli.quickstart.statements.php

ДОДАТОК А

КОД ПРОГРАМИ

Зміст файлу main.php:

```
<!DOCTYPE html>
<html>
<head>

   <meta charset = "UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link rel = "stylesheet" href = "style.css">
   <link rel="shortcut icon" href="/images/favicon.png" type="image/png">
   <title>Bibliopolium - інтернет-магазин книжок!</title>

</head>
<body class="background">



   <div class="header"> <!-- header -->
      <!--burger-menu content-->
      <div class="dropdown">
         <div class="dropdown-content" >
            <form method = "POST" action = "http://bibliopolium.com/filter_search.php">
               <strong>Жанри книжок</strong><br>
                  <input type="radio" id="Класика" name="filter_genre" value="Класика" >Класика<br>  <!--
Тіні забутих предків, Наталка-Полтавка -->
                  <input type="radio" id="Роман" name="filter_genre" value="Роман" >Роман  <br>  <!--
Кобзар, Енеїда -->
                  <input type="radio" id="Фантастика" name="filter_genre" value="Фантастика" >Фантастика  <br>
<!-- 1984 -->
                  <input type="radio" id="Поезія" name="filter_genre" value="Поезія" >Поезія  <br>  <!--
Заповіт, Думи мої, думи -->
                  <br>

               <strong>Автори книжок</strong><br>

                  <input type="radio" id="Михайло Коцюбинський" name="filter_author" value="Михайло
Коцюбинський" >Коцюбинський <br>  <!-- Тіні забутих предків -->
```

```
                <input type="radio" id="Іван Котляревський" name="filter_author" value="Іван Котляревський" >Котляревський  <br>  <!-- Енеїда, Наталка-Полтавка -->
                <input type="radio" id="Тарас Шевченко" name="filter_author" value="Тарас Шевченко" >Шевченко <br>  <!-- Кобзар, Заповіт -->
                <input type="radio" id="Джордж Орвел" name="filter_author" value="Джордж Орвел" >Орвелл <br><br>  <!-- 1984 -->

                <input type="submit" value="Шукати" id="filter_button" >
            </form>
        </div>
    </div>

    <!-- burger-menu end-->

    <a href="main.php"><div class="logo"></div></a><!-- logo -->

    <form method = "POST" action = "search.php"> <!-- search begin -->

        <div class="searching">
            <input type="text" id="search" placeholder="     Пошук..." name="search" class="search">
            <input type="submit" id="search_button" value="Шукати"class="search_button">
        </div>

    </form>
    <!-- search end -->

        <div class="phone">+380(00)-000-00-00<br>телефонуйте зараз!</div>
        <a href=""><div class="basket"></div></a>
        <a href="http://bibliopolium.com/authorization.php"><div class="Sign_In">Увійти</div></a>
</div>

<div class="content">
<?php

    $servername = "localhost";
    $username = "root";
    $password = "";
    $dbname = "biblio_DB";

    $conn = new mysqli($servername, $username, $password, $dbname);
```

```
    $sql = "SELECT  book_src, book_photo, book_price FROM books";
        $result = $conn->query($sql);
        if ($result->num_rows > 0) {
            while($row = $result->fetch_assoc()) {


             echo "<a href='".$row['book_src']."' target='blank' ><img class='main_img' src='".$row['book_photo']."'><span class='main_img_price' >".$row['book_price']."</span> </a>";
            }
        }


?>
  </div>

</body>
</html>
```

Зміст файлу style.css

```
.background{
  background-color: rgb(232, 232, 232);
  margin:0px;
  padding:0px;
  font-family:'Times New Roman', Times, serif;
}
/*Header*/
.header{
  width:100%;
  height:50px;
}


@font-face{ /*Uploaded Font*/
  font-family: Logo_font ;
  src: url(fonts/Lobster-Regular.ttf);}
/*.logo{
  font-size:32px;
  font-family: Logo_font;
  font-weight: 550;
  position:absolute;
  margin-left: 50px;
  margin-top:13px;
```

```
}*/
.logo{
    background-image: url(images/logo.png);
    background-size: cover;
    width:220px;
    height:56px;
    position: absolute;
    margin-left:70px;
    margin-top:2px;

}

.searching{
    position:absolute;
    margin-left:23%;
    margin-top:10px;
}
.search{
    position:absolute;
    margin-left:30%;
    border:0px;
    width:600px;
    height:40px;
    border-radius:20px 0px 0px 20px;



}
.search_button{
    background-color: rgb(138, 1, 223);
    color:white;
    position:absolute;
    border-radius:0px 20px 20px 0px;
    border:0px;
    width:100px;
    height:40px;
    margin-left: 603px;
    margin-top:1px;

}

.phone{
    position: absolute;
```

```
    margin-left:70%;
    margin-top:18px;
}
.basket{
    background-image: url(images/basket.png);
    background-size: cover;
    width: 32px;
    height: 32px;
    position: absolute;
    margin-left: 83%;
    margin-top: 20px;
    font-size: 22px;
}
.Sign_In{
    position: absolute;
    margin-left: 88%;
    margin-top: 25px;
    font-size: 22px;
    font-weight: 500;
}

/* burger-menu*/

.dropdown {
    position: absolute;
            color: #000000;
    background-image:url(images/burger_icon.png);
    background-size:cover;
            width:40px;
            height: 40px;
    margin-left:30px;
    margin-top:10px;
    display: inline-block;

}
.dropdown-content {
    display: none;
    position: absolute;
    background-color: rgba(255, 255, 255, 0.9) ;
    width: 200px;
    height:350px;
    margin-top:43px;
```

```
    margin-left:-20px;
    padding-left:15px;
    padding-top:20px;
    border-radius:0px 20px 0px 20px;
    z-index: 10;
    box-shadow:0px 0px 10px rgb(77, 77, 77);


}
.dropdown:hover .dropdown-content {
    display: block;
}


/*          */
/*Ending of Header*/


/*Settings of Filter and Content*/
.filter{
    background-color: rgb(255,255,255);
    width:15%;
    height:580px; /*dont forget to change while get data*/
    border-radius:25px;
    margin-left:0.7%;
    margin-top:20px;
    padding-left:15px;
    padding-top:25px;
    position:absolute;
    font-size: 21px;
    font-family:'Times New Roman', Times, serif;

}
.strong{
    font-size:24px;
    font-weight: 900;
}

.content{
    background-color: rgb(255,255,255);
    width:78%;
    height:580px; /*dont forget to change while get data*/
```

```
    border-radius:25px;
    margin-left:11%;
    margin-top:20px;
    padding-left:20px;
    position:absolute;

}

#filter_button{
    width:110px;
    height:37px;
    border:0px;
    background-color:rgb(138, 1, 223);
    color:white;
    border-radius: 20px 20px 20px 20px;
    font-size:17px;
    margin-left: 30px;
    margin-top:20px;
}

/*settings of div content*/

.main_img{
    margin-left:20px;
    margin-top:20px;
    box-shadow: 0px 0px 10px rgba(0,0,0,0.7);
    width:140px;
    height:200px;
    background-size:cover;
    z-index: 5;


}
.main_img_price{
    background:rgb(138, 1, 223);
    position:absolute;
    padding:5px;
    margin-left:-140px;
    margin-top:192px;
    color:white;
    z-index: 3;
}
```

```
/*settings of any book*/
.book_content{
   background-color: rgb(255,255,255);
   width:90%;
   height:620px; /*dont forget to change while get data*/
   border-radius:25px;
   margin-left:5%;
   margin-top:20px;
   position:absolute;
}

.book_name{
   font-size:20px;
   margin-left:100px;
   margin-top:20px;
}

.book_description{
   position:absolute;
   width:470px;
   margin-top:60px;
   margin-left:400px;
}
.book_price{
   position:absolute;
   margin-left:940px;
   margin-top:120px;
   width:300px;
   height:250px;
   border-radius: 25px;
   background-color:#dbdbdb;
   box-shadow:0px 0px 10px rgb(77, 77, 77);
}

.buy_button{
   background-color:rgb(138, 1, 223);
   width: 130px;
   height:40px;
```

```
    border-radius: 15px;
    padding-left:10px;
    padding-top:2px;
    margin-left:90px;
    margin-top:130px;
    font-size:20px;
    color:#fff;
    position:absolute;
    border:0px;

}

/*Authorization*/
.auth_text{
    width:500px;
    margin-left:34%;
    margin-top:90px;
    position:absolute;
}
.auth_sign-in{
    margin-left:35%;
    font-size:27px;

}
.auth_content{
    margin-top:10px;
    background-color: rgb(255,255,255);
    border-radius:15px;
    width:500px;
    height:500px;
    position:absolute;
    justify-content: center;
}
.auth_form{
    margin-left:23%;
    margin-top:120px;
    font-size:20px;
}
.auth-form-text{
    width:300px;
    background-color:rgb(244, 244, 244);
    border:1px solid #c1c1c1;
```

```
    font-size:17px;
    border-radius:15px;
    height:22px;

}
.auth-form-text:hover{
    background-color:rgba(255, 255, 255, 0.799);
    box-shadow:0px 0px 10px rgb(175, 175, 175);;
}

#auth_submit{
    width:110px;
    height:37px;
    border:0px;
    background-color:rgb(138, 1, 223);
    color:white;
    border-radius: 20px 20px 20px 20px;
    font-size:17px;
    margin-left: 80px;
    margin-top:20px;
}


        /*registration*/
.registration{
     font-size:16px;
     margin-left:90px;
     margin-top:60px;
     position:absolute;
     color:rgb(117, 117, 117);
}

.wrong_registration{
    position: absolute;
    margin-top:42%;
    margin-left:28%;
    text-align: center;
    font-size:20px;
    color:#ff0000;
    text-shadow:0px 0px 10px rgb(37, 34, 34);;
}
```

```
.wrong_registration_decoration{
    color:#000;
    text-decoration:underline;
}

        /*registration end*/

/*Authorization end*/

/* Transaction of buying */

.buy_text{
    width:500px;
    margin-left:25%;
    margin-top:90px;
    position:absolute;
}
.buy_content{
    margin-top:10px;
    background-color: rgb(255,255,255);
    border-radius:15px;
    width:800px;
    height:500px;
    position:absolute;
    justify-content: center;
    font-size:20px;
}

.buy_text_h{
    margin-left:10%;
    font-size:30px;
}
/**/

.book_image{
    background-size: cover;
    position:absolute;
    width:150px;
    height:230px;
    margin-top:15px;
    margin-left:0px;
    box-shadow: 10px 10px 15px rgb(132, 132, 132);
```

```
}
.buy_book_name{
    position: absolute;
    margin-top:40px;
    margin-left:30px;
}

.buy_page_button{
    background-color:rgb(138, 1, 223);
    width: 130px;
    height:40px;
    border-radius: 15px;
    padding-left:10px;
    padding-top:2px;
    margin-left:43%;
    margin-top:37%;
    font-size:20px;
    color:#fff;
    position:absolute;
    border:0px;
}

.buy_inputs{
    position:absolute;
    margin-left:33%;
    margin-top:15%;
}
.buy_inputs_class{
    font-size:17px;
    border:0px;
    border-bottom:2px solid rgb(138, 1, 223);
    width:300px;

}
.buy_decline{
    position: absolute;
    margin-top:23%;
    margin-left:38%;
    text-align: center;
    font-size:20px;
}
/*hyperlink settings*/
```

```
a {
        color: rgb(0, 0, 0) ;
        text-decoration:none;
}

 a:visited {
        color: rgb(0, 0, 0);

        text-decoration:none;
}
 a:active {
        color: rgb(0, 0, 0);
        text-decoration:none;

}
a:hover{
   color:black;

}
/**/
```

Зміст файлу filter_search.php

```
<!DOCTYPE html>
<html>
<head>

   <meta charset = "UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">

   <link rel = "stylesheet" href = "style.css">
   <link rel="shortcut icon" href="/images/favicon.png" type="image/png">
   <title>Bibliopolium - інтернет-магазин книжок!</title>

</head>
<body class="background">

   <div class="header"> <!-- header -->
      <!--burger-menu content-->
      <div class="dropdown">
         <div class="dropdown-content" >
            <form method = "POST" action = "http://bibliopolium.com/filter_search.php">
               <strong>Жанри книжок</strong><br>
```

```
                <input type="radio" id="Класика" name="filter_genre" value="Класика" >Класика<br>  <!-- Тіні забутих предків, Наталка-Полтавка -->
                <input type="radio" id="Роман" name="filter_genre" value="Роман" >Роман  <br>  <!-- Кобзар, Енеїда -->
                <input type="radio" id="Фантастика" name="filter_genre" value="Фантастика" >Фантастика  <br> <!-- 1984 -->
                <input type="radio" id="Поезія" name="filter_genre" value="Поезія" >Поезія  <br>  <!-- Заповіт, Думи мої, думи -->
                <br>

              <strong>Автори книжок</strong><br>

                <input type="radio" id="Михайло Коцюбинський" name="filter_author" value="Михайло Коцюбинський" >Коцюбинський <br>  <!-- Тіні забутих предків -->
                <input type="radio" id="Іван Котляревський" name="filter_author" value="Іван Котляревський" >Котляревський  <br>  <!-- Енеїда, Наталка-Полтавка -->
                <input type="radio" id="Тарас Шевченко" name="filter_author" value="Тарас Шевченко" >Шевченко <br>  <!-- Кобзар, Заповіт -->
                <input type="radio" id="Джордж Орвел" name="filter_author" value="Джордж Орвел" >Орвелл <br><br>  <!-- 1984 -->

                <a href="http://bibliopolium.com/main.php" >Прибрати фільтр</a>
            </form>
        </div>
    </div>

    <!-- burger-menu end-->

    <a href="main.php"><div class="logo"></div></a><!-- logo -->

    <form method = "POST" action = "search.php"> <!-- search begin -->

        <div class="searching">
            <input type="text" id="search" placeholder="    Пошук..." name="search" class="search">
            <input type="submit" id="search_button" value="Шукати"class="search_button">
        </div>

    </form>
    <!-- search end -->

        <div class="phone">+380(00)-000-00-00<br>телефонуйте зараз!</div>
        <a href=""><div class="basket"></div></a>
```

```
        <a href=""><div class="Sign_In">Увійти</div></a>
    </div>

    <div class="content">
    <?php
        $filter_genre  =    $_POST['filter_genre'];
        $filter_author =    $_POST['filter_author'];

        $servername = "localhost";
        $username = "root";
        $password = "";
        $dbname = "biblio_DB";

        $conn = new mysqli($servername, $username, $password, $dbname);

            //when true && true
        if($filter_genre &&  $filter_author){
        $sql = "SELECT  book_src, book_photo, book_price FROM books WHERE book_author='$filter_author' AND book_genre='$filter_genre' ";
            $result = $conn->query($sql);
            if ($result->num_rows > 0) {
                while($row = $result->fetch_assoc()) {

                    echo "<a href='".$row['book_src']."' target='blank' ><img class='main_img' src='".$row['book_photo']."'><span class='main_img_price' >".$row['book_price']."</span> </a>";;

                }
            }
        }
        else{
            // when false && true
            if($filter_author){
                $sql = "SELECT  book_src, book_photo, book_price FROM books WHERE book_author='$filter_author'";
                $result = $conn->query($sql);
                if ($result->num_rows > 0) {
                    while($row = $result->fetch_assoc()) {

                        echo "<a href='".$row['book_src']."' target='blank' ><img class='main_img' src='".$row['book_photo']."'><span class='main_img_price' >".$row['book_price']."</span> </a>";

                    }
                }
```

```
        }
        else{    // when true && false
            if($filter_genre){
              $sql = "SELECT  book_src, book_photo,book_price FROM books WHERE book_genre='$filter_genre'";
              $result = $conn->query($sql);
                if ($result->num_rows > 0) {
                    while($row = $result->fetch_assoc()) {
                        echo "<a href='".$row['book_src']."' target='blank' ><img class='main_img' src='".$row['book_photo']."'><span class='main_img_price' >".$row['book_price']."</span> </a>";
                    }
                }
            }}
    }



    ?>
    </div>

</body>
</html>
```

Зміст файлу search.php

```
<!DOCTYPE html>
<html>
<head>

    <meta charset = "UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel = "stylesheet" href = "style.css">
    <link rel="shortcut icon" href="/images/favicon.png" type="image/png">
    <title>Bibliopolium - інтернет-магазин книжок!</title>

</head>
<body class="background">
    <div class="header"> <!-- header -->
        <!--burger-menu content-->
        <div class="dropdown">
            <div class="dropdown-content" >
```

```
        <form method = "POST" action = "http://bibliopolium.com/filter_search.php">
          <strong>Жанри книжок</strong><br>
            <input type="radio" id="Класика" name="filter_genre" value="Класика" >Класика<br>  <!-- Тіні забутих предків, Наталка-Полтавка -->
            <input type="radio" id="Роман" name="filter_genre" value="Роман" >Роман  <br>  <!-- Кобзар, Енеїда -->
            <input type="radio" id="Фантастика" name="filter_genre" value="Фантастика" >Фантастика  <br> <!-- 1984 -->
            <input type="radio" id="Поезія" name="filter_genre" value="Поезія" >Поезія  <br>  <!-- Заповіт, Думи мої, думи -->
            <br>

          <strong>Автори книжок</strong><br>

            <input type="radio" id="Михайло Коцюбинський" name="filter_author" value="Михайло Коцюбинський" >Коцюбинський <br>  <!-- Тіні забутих предків -->
            <input type="radio" id="Іван Котляревський" name="filter_author" value="Іван Котляревський" >Котляревський  <br>  <!-- Енеїда, Наталка-Полтавка -->
            <input type="radio" id="Тарас Шевченко" name="filter_author" value="Тарас Шевченко" >Шевченко <br>  <!-- Кобзар, Заповіт -->
            <input type="radio" id="Джордж Орвел" name="filter_author" value="Джордж Орвел" >Орвелл <br><br>  <!-- 1984 -->

            <a href="http://bibliopolium.com/main.php" >Прибрати фільтр</a>
        </form>
      </div>
    </div>

    <!-- burger-menu end-->

    <a href="main.php"><div class="logo"></div></a><!-- logo -->

    <form method = "POST" action = "search.php"> <!-- search begin -->

      <div class="searching">
        <input type="text" id="search" placeholder="    Пошук..." name="search" class="search">
        <input type="submit" id="search_button" value="Шукати"class="search_button">
      </div>

    </form>
    <!-- search end -->
```

```
        <div class="phone">+380(00)-000-00-00<br>телефонуйте зараз!</div>
        <a href=""><div class="basket"></div></a>
        <a href=""><div class="Sign_In">Увійти</div></a>
    </div>



    <div class="content">
        <?php
            $book  =    $_POST['search'];

            $servername = "localhost";
            $username = "root";
            $password = "";
            $dbname = "biblio_DB";

            $conn = new mysqli($servername, $username, $password, $dbname);

            if($book){
                $sql = "SELECT  book_src, book_photo, book_price FROM books WHERE book_name LIKE '%$book%' ";
                $result = $conn->query($sql);
                    if ($result->num_rows > 0) {
                        while($row = $result->fetch_assoc()) {

                            echo "<a href='".$row['book_src']."' target='blank' ><img class='main_img' src='".$row['book_photo']."'><span class='main_img_price' >".$row['book_price']."</span> </a>";
                        }
                    }
                    else{
                        $sql = "SELECT  book_src, book_photo, book_price FROM books WHERE book_author LIKE '%$book%' ";
                        $result = $conn->query($sql);
                            if ($result->num_rows > 0) {
                                while($row = $result->fetch_assoc()) {
                                    echo "<a href='".$row['book_src']."' target='blank' ><img class='main_img' src='".$row['book_photo']."'><span class='main_img_price' >".$row['book_price']."</span> </a>";
                                }
                            }
                    }
            }

        ?>
```

```
    </div>

</body>
</html>
```

Зміст файлу registration.php

```
<!DOCTYPE html>
<html>
<head>

    <meta charset = "UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <a href="main.php"><div class="logo"></div></a>

    <link rel = "stylesheet" href = "style.css">
    <link rel="shortcut icon" href="/images/favicon.png" type="image/png">
    <title>Bibliopolium - інтернет-магазин книжок!</title>

</head>

<body class="background">
    <div class="auth_text">
        <span class="auth_sign-in">Реєстрація</span>
        <div class="auth_content">
            <form method = "POST" action = "registration-2.php" class="auth_form" >

                Логін<br>
                <input type="email"  placeholder=" sample@sample.com" name="gmail" class="auth-form-text"><br><br>

                Ім'я<br>
                <input type="text"  placeholder=" Bohdan" name="name" class="auth-form-text"><br><br>

                Пароль<br>
                <input type="password"  placeholder=" ●●●●●" name="password" class="auth-form-text"><br>

                <input type="submit" value="Увійти" id="auth_submit"><br>
                <span class="registration">Вже є аккаунт?<a href="authorization.php" style="color:rgb(117, 117, 117);">
Увійти!</a></span>
            </form>
```

```
    </div>
  </div>

</body>
</html>
```

Зміст файлу registration2.php

```
<!DOCTYPE html>
<html>
<head>

  <meta charset = "UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <a href="main.php"><div class="logo"></div></a>

  <link rel = "stylesheet" href = "style.css">
  <link rel="shortcut icon" href="/images/favicon.png" type="image/png">
  <title>Bibliopolium - інтернет-магазин книжок!</title>

</head>

<body class="background">
  <div class="auth_text">
    <span class="auth_sign-in">Реєстрація</span>
    <div class="auth_content">
      <?php
        $email =    $_POST['gmail'];
        $name =     $_POST['name'];
        $pswd = $_POST['password'];


        if(!$email || !$name || !$pswd){
            echo "<div class='wrong_registration'>Ви ввели не всі дані!<a href='http://bibliopolium.com/registration.php' class='wrong_registration_decoration' > <br> Повернутися до реєстрації.</a></div> ";
        }
        else{

            $servername = "localhost";
```

```
                $username = "root";
                $password = "";
                $dbname = "biblio_DB";

                $conn = new mysqli($servername, $username, $password, $dbname);

                $insert = "INSERT INTO user_account (account_email, account_password, account_name) VALUES ('$email','$pswd','$name')";

                if (mysqli_query($conn, $insert)){

                  $sql = "SELECT  account_email, account_password, account_name FROM user_account WHERE account_email ='".$email."'";

                  $result = $conn->query($sql);
                  if ($result->num_rows > 0) {
                    while($row = $result->fetch_assoc()) {

                      echo "Ви зареєструвались!";
                    }
                  }
                }
            }
        ?>
    </div>
  </div>

</body>
</html>
```

Зміст файлу sign-in.php

```
<!DOCTYPE html>
<html>
<head>

  <meta charset = "UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <a href="main.php"><div class="logo"></div></a>
```

```
    <link rel = "stylesheet" href = "style.css">
    <link rel="shortcut icon" href="/images/favicon.png" type="image/png">
    <title>Bibliopolium - інтернет-магазин книжок!</title>

</head>

<body class="background">
    <div class="auth_text">
        <span class="auth_sign-in">Доброго дня!</span>
        <div class="auth_content">
            <?php
                $email =    $_POST['gmail'];
                $pswd = $_POST['password'];



                if(!$email || !$pswd){
                    echo "<div class='wrong_registration'>Ви ввели не всі дані!<a href='http://bibliopolium.com/authorization.php' class='wrong_registration_decoration' > <br> Повернутися до авторизації.</a></div> ";
                }
                else{

                    $servername = "localhost";
                    $username = "root";
                    $password = "";
                    $dbname = "biblio_DB";

                    $conn = new mysqli($servername, $username, $password, $dbname);

                      $sql = "SELECT * FROM user_account WHERE account_email ='$email' AND account_password='$pswd'";
                      $result = $conn->query($sql);

                      if ($result->num_rows > 0) {

                        while($row = $result->fetch_assoc()) {
                          $user_id = $row['account_id'];

                          $sql2 ="SELECT * FROM book_sells WHERE user_id='$user_id'";
                          $result = $conn->query($sql2);

                          if ($result->num_rows > 0) {
```

```
                        while($row2 = $result->fetch_assoc()) {
                            //$book_id= $row2['book_id'];
                            $sql3 ="SELECT * FROM books WHERE book_id='".$row2['book_id']."'";
                            $result = $conn->query($sql3);

                            if ($result->num_rows > 0) {
                                while($row3 = $result->fetch_assoc()) {
                                    $book_name = $row3['book_name'];
                                    $book_price = $row3['book_price'];

                                    echo "<div style='margin-left:25%;margin-top:20px;' >Куплена книга $book_name за ціною $book_price</div>";

                                }
                            }
                        }

                    }
                }
                else{
                    echo" Ви ввели невірні дані або аккаунту не існує.<br><a href='http://bibliopolium.com/authorization.php' >Повернутися до входу</a>";
                }

            }
        ?>
    </div>
</div>

</body>
</html>
```

Зміст файлу authorization.php

```
<!DOCTYPE html>
<html>
<head>

  <meta charset = "UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <a href="main.php"><div class="logo"></div></a>

    <link rel = "stylesheet" href = "style.css">
    <link rel="shortcut icon" href="/images/favicon.png" type="image/png">
    <title>Bibliopolium - інтернет-магазин книжок!</title>

</head>

<body class="background">
    <div class="auth_text">
        <span class="auth_sign-in">Вхід в аккаунт</span>
        <div class="auth_content">
            <form method = "POST" action = "sign-in.php" class="auth_form" >

                Логін<br>
                <input type="email"  placeholder=" sample@sample.com" name="gmail" class="auth-form-text"><br><br>

                Пароль<br>
                <input type="password"  placeholder=" ●●●●●" name="password" class="auth-form-text"><br>

                <input type="submit" value="Увійти" id="auth_submit"><br>
                <span class="registration">Немає Аккаунту?<a href="registration.php" style="color:rgb(117, 117, 117);">
Зареєструватись!</a></span>
            </form>
        </div>
    </div>

</body>
</html>
```

Зміст файлу book_1984.php

Програмний код цього файлу демонструє шаблон створення нових книжок, з використанням даних БД, а саме зображення книги, опису книги, назви книги, автора книги та її ціну. Подібних файлів як цей було створено ще 9.

```
<!DOCTYPE html>
<html>
<head>
```

```
    <meta charset = "UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel = "stylesheet" href = "style.css">
    <link rel="shortcut icon" href="/images/favicon.png" type="image/png">
    <title>Bibliopolium - інтернет-магазин книжок!</title>

</head>
<body class="background">

    <div class="header"> <!-- header -->
        <!--burger-menu content-->
        <div class="dropdown">
            <div class="dropdown-content" >
                <form method = "POST" action = "http://bibliopolium.com/filter_search.php">
                    <strong>Жанри книжок</strong><br>
                        <input type="radio" id="Класика" name="filter_genre" value="Класика" >Класика<br>              <!--
Тіні забутих предків, Наталка-Полтавка -->
                        <input type="radio" id="Роман" name="filter_genre" value="Роман" >Роман  <br>                  <!--
Кобзар, Енеїда -->
                        <input type="radio" id="Фантастика" name="filter_genre" value="Фантастика" >Фантастика  <br>
<!-- 1984 -->
                        <input type="radio" id="Поезія" name="filter_genre" value="Поезія" >Поезія          <br>          <!--
Заповіт, Думи мої, думи -->
                        <br>

                    <strong>Автори книжок</strong><br>

                        <input type="radio" id="Михайло Коцюбинський" name="filter_author" value="Михайло
Коцюбинський" >Коцюбинський <br>     <!-- Тіні забутих предків -->
                        <input type="radio" id="Іван Котляревський" name="filter_author" value="Іван Котляревський"
>Котляревський  <br>  <!-- Енеїда, Наталка-Полтавка -->
                        <input type="radio" id="Тарас Шевченко" name="filter_author" value="Тарас Шевченко" >Шевченко
<br>   <!-- Кобзар, Заповіт -->
                        <input type="radio" id="Джордж Орвел" name="filter_author" value="Джордж Орвел" >Орвелл
<br><br>  <!-- 1984 -->

                        <input type="submit" value="Шукати" id="filter_button" >
                </form>
            </div>
        </div>
```

```
    <!-- burger-menu end-->

    <a href="main.php"><div class="logo"></div></a><!-- logo -->

    <form method = "POST" action = "search.php"> <!-- search begin -->

        <div class="searching">
            <input type="text" id="search" placeholder="     Пошук..." name="search" class="search">
            <input type="submit" id="search_button" value="Шукати"class="search_button">
        </div>

    </form>
    <!-- search end -->

        <div class="phone">+380(00)-000-00-00<br>телефонуйте зараз!</div>
        <a href=""><div class="basket"></div></a>
        <a href=""><div class="Sign_In">Увійти</div></a>
</div>
<div class="book_content">
    <div class="book_name">
        Джордж Орвелл<br>
        Книга "1984"
    </div>
    <div style="
            background-image:url(images/1984.jpg);
            background-size: cover;
            position:absolute;
            width:300px;
            height:490px;
            margin-top:15px;
            margin-left:45px;
            box-shadow: 10px 10px 15px rgb(132, 132, 132);"></div>
    <div class="book_description">
    <h2>Опис</h2>
        Про що книга «1984» автора Джордж Орвелл
        Написаний понад 70 років тому, цей твір виявився
        пророчим: образ влади, яка намагається контролювати хід історії,
        незмінно актуальний.<br><br>
        <span style="color:rgb(0, 186, 0); font-size:23px;margin-left:90px; font-weight: 800; " >В НАЯВНОСТІ</span>
    </div>
    <div class="book_price">
```

```
        <div style="margin-left:100px; margin-top:10px; position:absolute; font-size:19px;" >Замовлення</div>
        <?php
        $servername = "localhost";
        $username = "root";
        $password = "";
        $dbname = "biblio_DB";

        $conn = new mysqli($servername, $username, $password, $dbname);

        $sql = "SELECT  * FROM books WHERE book_name = '1984'";
        $result = $conn->query($sql);

        if ($result->num_rows > 0) {
           // output data of each row
           while($row = $result->fetch_assoc()) {
             echo "<div style='margin-left:50px; margin-top:50px; position:absolute;font-size:19px;'>".$row["book_author"].", книга " . $row["book_name"]." <br><br> Ціна за шт.". $row["book_price"]. " </div><br>";

              echo "
              <form method = 'POST' action = 'book_buy.php'>
                 <input type='hidden' name='book_id' id='book_id' value='".$row['book_id']."'>
                 <input type='submit' class='buy_button' value='Купити'>

              </form>";
           }
        }
        ?>
    </div>
    </div>

</body>
</html>
```

ДОДАТОК Б

ВІДГУК

на кваліфікаційну роботу бакалавра
на тему:
«Розробка автоматизованої інформаційної системи з продажу книжок»
студента групи 122-20ск Францева Богдана Вікторовича

ДОДАТОК В

РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра
на тему:
«Розробка автоматизованої інформаційної системи з продажу книжок»
студента групи 122-20ск Францева Богдана Вікторовича

ДОДАТОК Г

ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ

ДОДАТОК Д

ПЕРЕЛІК ФАЙЛІВ НА ОПТИЧНОМУ НОСІЇ

Ім’я файла	**Опис**
Пояснювальні документи	
Диплом.docx	Пояснювальна записка до дипломного проекту. Документ Word.
Диплом.pdf	Пояснювальна записка до дипломного проекту в форматі PDF
Програма	
bibliopolium.com.rar	Архів. Містить коди сторінок веб-сайту.
biblio_DB.sql	Імпортована БД MySQL
Презентація	
Презентація.ppt	Презентація дипломного проекту