

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 Чернявського Ігоря Сергійовича
(ПІБ)

академічної групи 121-20ск-1
(шифр)

спеціальності 121 Інженерія програмного забезпечення
(код і назва спеціальності)

освітньої програми Інженерія програмного забезпечення
(назва освітньої програми)

на тему: Розробка веб-додатку для інтернет-магазину
косметики ручної роботи з використанням бібліотеки React

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	доц. Спиринцев В.В.			
розділів:				
спеціальний	доц. Спиринцев В.В.			
економічний	доц. Касьяненко Л.В.			
Рецензент				
Нормоконтролер	ст. викладач Мартиненко А.А.			

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 121-20ск-1 Чернявського Ігоря Сергійовича
(група) (прізвище та ініціали)

тема кваліфікаційної роботи Розробка веб-додатку для
інтернет-магазину косметики ручної роботи з використанням
бібліотеки React

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

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

Завдання видав

(підпис)

доц. Спирінцев В.В

(посада, прізвище, ініціали)

Завдання прийняв до виконання

(підпис)

Чернявський І.С

(прізвище, ініціали)

Дата видачі завдання: 14.01.2023 р.

Термін подання кваліфікаційної роботи до ЕК: 11.06.2023 р.

РЕФЕРАТ

Пояснювальна записка: 79 с., 35 рис., 7 табл. 3 дод., 21 джерело.

Об'єкт розробки: веб-орієнтована інформаційна система для автоматизації діяльності підприємства в сфері електронної комерції.

Мета кваліфікаційної роботи: розробка програмного забезпечення веб-орієнтованої інформаційної системи для автоматизації діяльності компанії з реалізації продажу косметики ручної роботи в сфері електронної комерції.

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 проаналізовано предметну галузь, визначено актуальність завдання та призначення розробки, сформульовано постановку завдання, зазначено вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі проаналізовані наявні рішення, обрано платформу для розробки, виконано проектування і розробку веб-орієнтованої інформаційної системи, описана робота системи, алгоритм і структура його функціонування, а також виклик та завантаження додатку, визначено вхідні і вихідні дані, охарактеризовано склад параметрів технічних засобів.

В економічному розділі визначено трудомісткість розробленої інформаційної системи, проведений підрахунок вартості роботи по створенню додатку та розраховано час на його створення.

Практичне значення полягає у створенні веб-орієнтованої інформаційної системи, яка надає компанії зручну та ефективну платформу для продажу товарів косметики ручної роботи через електронну комерцію. Система допомагає автоматизувати процеси управління та маркетингу, що дозволяє підвищити лояльність клієнтів та збільшити прибуток компанії.

Актуальність розробки інформаційної системи для автоматизації діяльності компанії в сфері електронної комерції не викликає сумніву, оскільки електронна комерція стабільно зростає, а підприємствам потрібні конкурентні переваги для ефективного функціонування в цифровій економіці та досягнення успіху на ринку.

Список ключових слів: ІНФОРМАЦІЙНА СИСТЕМА, ІНТЕРНЕТ-МАГАЗИН, БАЗА ДАНИХ, БРАУЗЕР, SQL, MYSQL, HTML, CSS, SCSS JAVASCRIPT, REACT, NODE.

ABSTRACT

Explanatory note: 79 pp., 35 fig., 7 tables, 3 extra, 21 sources.

The object of development: web-oriented information system for automating the activities of an enterprise in the field of e-commerce.

The purpose of the diploma project: software development web-based information system to automate the company's activities in the field of e-commerce.

The introduction considers the analysis and the current state of the problem, specifies the purpose of the qualification work and the scope of its application, substantiates the relevance of the topic and clarifies the formulation of the problem.

In the first chapter, the subject area is analyzed, the relevance of the task and the purpose of development are determined, the statement of the problem is formulated, the requirements for software implementation, technologies and software are indicated.

In the second section, the available solutions are analyzed, a platform for development is selected, the design and development of a web-oriented information system is carried out, the operation of the system, the algorithm and structure of its functioning, as well as the call and loading of the application are described, the input and output data are determined, the composition of the parameters of the technical means.

In the economic section, the labor intensity of the developed information system is determined, the cost of work on creating an application is calculated and the time for its creation is calculated.

The practical significance is to create a web-based information system that provides a convenient and efficient platform for the company to sell handmade cosmetics products through e-commerce. The system helps to automate management and marketing processes, which allows to increase customer loyalty and increase the company's profit.

The relevance of the development of an information system for the automation of the company's activities in the field of e-commerce does not arise, since e-commerce is growing steadily, and enterprises need competitive advantages to function effectively in the digital economy and achieve success in the market.

List of keywords: INFORMATION SYSTEM, ONLINE STORE, DATABASE, BROWSER, SQL, MYSQL, HTML, CSS, SCSS JAVASCRIPT, REACT, NODE.

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	7
ВСТУП	8
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ	10
1.1. Загальні відомості з предметної галузі	10
1.1.1. Особливості електронної комерції.....	10
1.2. Призначення розробки та галузь застосування	14
1.3. Підстава для розробки	17
1.4. Постановка завдання	17
1.5. Вимоги до програми або програмного виробу	19
1.5.1. Вимоги до функціональних характеристик	19
1.5.2. Вимоги до інформаційної безпеки	20
1.5.3. Вимоги до складу та параметрів технічних засобів	21
1.5.4. Вимоги до інформаційної та програмної сумісності	21
РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ	22
2.1. Функціональне призначення програми	22
2.2. Опис застосованих математичних методів	23
2.3. Опис використаної архітектури та шаблонів проектування	23
2.4. Опис використаних технологій та мов програмування	26
2.5. Опис структури програми та алгоритмів її функціонування	35
2.6. Обґрунтування та організація вхідних та вихідних даних програми.....	46
2.7. Опис розробленого програмного продукту.....	47
2.7.1. Використані технічні засоби.....	47
2.7.2. Використані програмні засоби	48
2.7.3. Виклик та завантаження програми	50
2.7.4. Опис інтерфейсу користувача	51

РОЗДІЛ 3 ЕКОНОМІЧНА ЧАСТИНА	67
3.1. Визначення трудомісткості та вартості розробки програмного продукту ..	67
3.2. Розрахунок витрат на створення програми.....	71
ВИСНОВКИ	74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	75
ДОДАТОК А КОД ПРОГРАМИ	77
ДОДАТОК Б ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ	78
ДОДАТОК В ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ.....	79

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- ПЗ - Програмне забезпечення
- ІС - Інформаційна система
- ОС - Операційна система
- БД - База даних
- СУБД - Система управління базами даних
- CMS - Система управління контентом
- CSS - Cascading Style Sheets
- SQL - Structured Query Language
- DB- Database
- JS - JavaScript
- HTML - Hyper Text Markup Language

ВСТУП

Впровадження електронної комерції у сучасній бізнес-діяльності стає все більш важливим для підприємств у зв'язку зі стрімким розвитком інформаційних технологій та зміною підходів у взаємодії між суб'єктами ринку. Онлайн-продажі дозволяють компаніям проникати на нові ринки, привертати більше клієнтів і підвищувати ефективність своєї діяльності.

У цьому контексті, актуальність розробки інформаційної системи для автоматизації діяльності підприємства в сфері електронної комерції стає незаперечною. Системи, які забезпечують інтернет-магазини, є важливим інструментом для залучення та обслуговування клієнтів у цифровому просторі.

Об'єктом дослідження даної кваліфікаційної роботи є розробка веб-орієнтованої інформаційної системи, спрямованої на автоматизацію діяльності підприємства в сфері електронної комерції. Основною метою цієї роботи є розробка програмного забезпечення для інформаційної системи, що охоплює маркетингову діяльність підприємства. Ця система спроектована з метою підвищення ефективності роботи підприємства шляхом автоматизації процесів у сфері електронної комерції.

Розроблена інформаційна система сприятиме зручній та ефективній реалізації товарів косметики ручної роботи. Це дозволить підприємству ефективно виконувати функції маркетингу, управління та продажу товарів в онлайн-середовищі. Косметика ручної роботи має особливу цінність та привабливість для клієнтів, а розроблена система надасть зручну платформу для її просування та продажу в електронній комерції.

Для досягнення поставленої мети в рамках кваліфікаційної роботи передбачається вирішення наступних задач:

– Розгляд особливостей електронної комерції в контексті розвитку національної економіки. Дослідження поточного стану ринку електронної комерції, трендів та перспектив розвитку цієї сфери.

– Формування основних вимог до розробленого програмного продукту, зокрема вимог щодо функціональних характеристик, інформаційної безпеки, складу та параметрів технічних засобів, інформаційної та програмної сумісності, а також змісту інформаційного наповнення системи.

– Проектування програмного продукту: визначити функціональне призначення програми, описати застосовані математичні методи, архітектуру та шаблони проектування. Вказати використані технології та мови програмування. Розробити структуру програми та алгоритми її функціонування.

– Обґрунтувати та організувати вхідні та вихідні дані програми, які забезпечують зручну інтерактивну взаємодію з користувачами.

– Розробка веб-додатку для інтернет-магазину косметики ручної роботи з використанням бібліотеки React. Реалізація основного функціоналу системи, забезпечення зручного інтерфейсу для користувачів та можливості безперервного замовлення та оплати товарів.

– Розрахунок трудомісткості та витрат на розробку програмного продукту. Оцінити затрати часу, фінансові витрати та інші ресурси, необхідні для розробки програмного продукту, з урахуванням обсягу робіт, складності проекту та вимог до продукту.

Практичне значення даної кваліфікаційної роботи полягає у створенні веб-орієнтованої інформаційної системи, яка забезпечує зручність та ефективність процесів у сфері електронної комерції. Розроблений веб-додаток для інтернет-магазину косметики ручної роботи з використанням бібліотеки React дозволить покращити взаємодію з клієнтами, спростити процес замовлення та покупки товарів, а також підвищити рівень задоволеності користувачів.

Таким чином, дана кваліфікаційна робота спрямована на вивчення особливостей електронної комерції, а також розробку інформаційної системи для автоматизації діяльності підприємства в цій сфері. Очікується, що результати дослідження та розробки дадуть позитивний вплив на роботу підприємства і сприятимуть підвищенню його ефективності.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Загальні відомості з предметної галузі

1.1.1. Особливості електронної комерції

В сучасній економіці, де зростає конкуренція між підприємствами, виникає потреба у швидкій реакції на зміни на ринку товарів і послуг, а також в ефективному здійсненні комерційних операцій. Один з перспективних напрямків в бізнесі в умовах жорсткої конкуренції - електронна комерція, що охоплює всі види торгівлі через Інтернет.

Важливою складовою електронної комерції є веб-сайти [1]. Вони є потужним інструментом для розвитку бізнесу, оскільки надають підприємствам можливість присутності в онлайн-просторі та взаємодії зі своїми клієнтами в цифровому середовищі. Основні особливості електронної комерції через веб-сайти включають:

- Глобальний охоплюючий ринок: електронна комерція через веб-сайти не має прив'язки до географічного місця розташування підприємства. Це дозволяє досягти широкої аудиторії клієнтів на всій планеті, залежно від галузі бізнесу.

- Зручність для клієнтів: веб-сайти надають зручну платформу для клієнтів, де вони можуть ознайомитися з асортиментом товарів, порівняти ціни, знайти необхідну інформацію та зробити покупку в зручний для них час. Вони можуть також скористатися різними платіжними методами, що спрощує процес оплати.

- Збільшення ефективності: електронна комерція дозволяє автоматизувати багато процесів, що пришвидшує виконання замовлень та операцій. Інтеграція з системами управління запасами, фінансовими системами

та іншими допомагає зменшити ризики помилок та збільшити ефективність бізнесу.

– Персоналізація та маркетингові можливості: веб-сайти дозволяють підприємствам збирати дані про своїх клієнтів і аналізувати їх поведінку. Це дає можливість персоналізувати пропозиції, рекомендації та акції, що покращує спілкування з клієнтами і підвищує шанси на здійснення продажів.

– Зниження витрат: електронна комерція через веб-сайти дозволяє підприємствам зменшити витрати на оренду приміщень, утримання персоналу та інфраструктури. Крім того, ефективне використання онлайн-реклами може допомогти залучити клієнтів з меншими витратами порівняно з традиційними методами реклами.

– Аналітика та звітність: веб-сайти дозволяють збирати велику кількість даних про клієнтів, їхню поведінку та переваги. Це надає можливість проводити аналіз, розуміти потреби ринку та здійснювати своєчасні корективи в стратегії бізнесу.

Електронна комерція через веб-сайти стала невід'ємною складовою сучасного бізнесу, забезпечуючи компаніям можливість просування, розширення клієнтської бази та підвищення ефективності операцій. З розвитком технологій та зміною споживацьких пристрастей, важливість електронної комерції та веб-сайтів у бізнесі продовжує зростати, створюючи нові можливості для підприємств у цифровому світі [2].

Один з прикладів успішних інтернет-магазинів у сфері косметики є міжнародний роздрібний магазин Sephora. Його веб-сайт пропонує широкий асортимент косметичних товарів із відомих брендів. Зручний інтерфейс сайту, детальні описи продуктів, відгуки клієнтів та можливість замовлення товарів онлайн сприяють зручній та ефективній покупці. Інтернет-магазин зображено на рис. 1.1 .

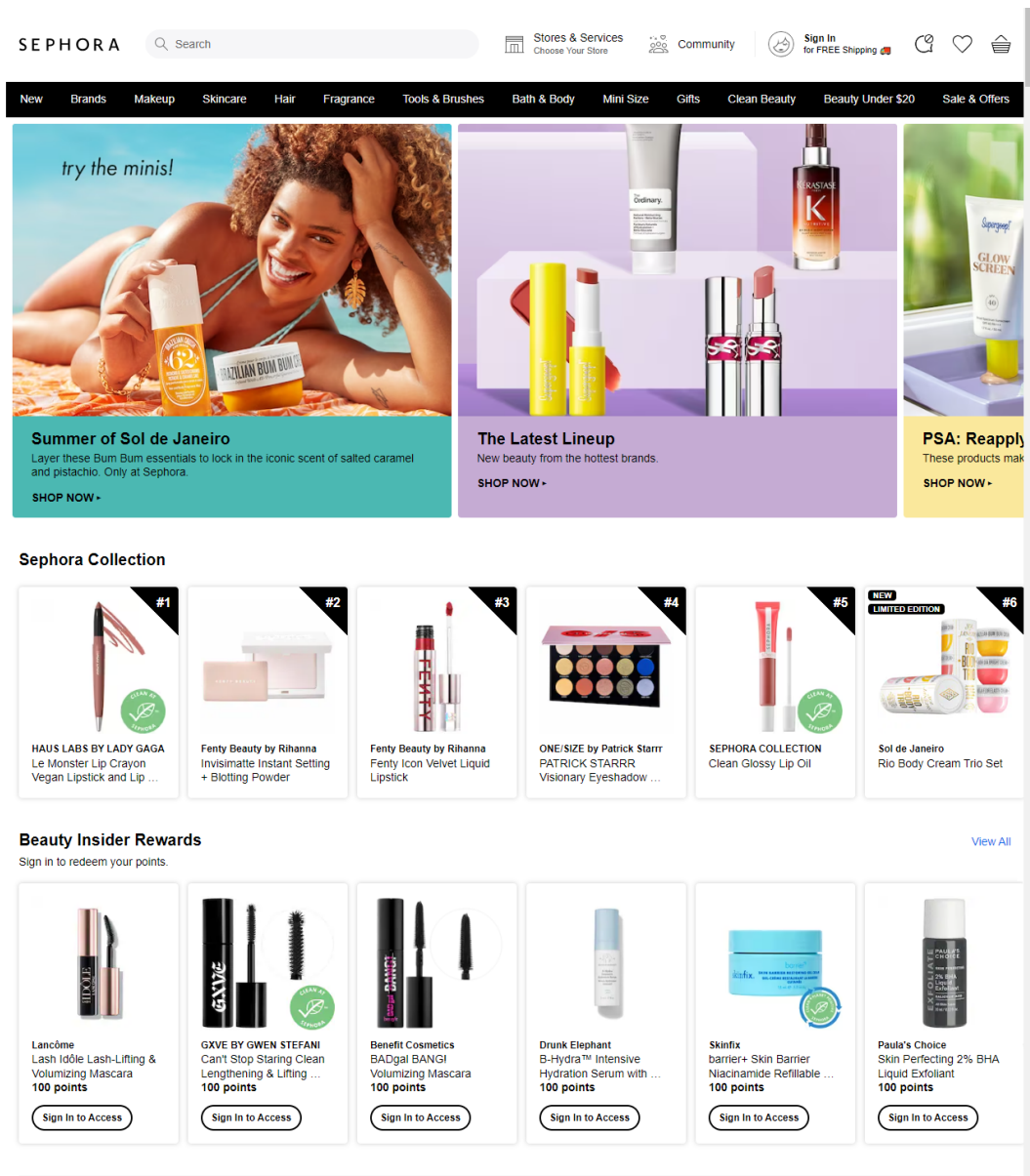


Рис. 1.1 Головна сторінка інтернет-магазину Sephora

Інший приклад - бренд косметики Glossier, який фокусується на підкресленні природної краси. Веб-сайт Glossier відрізняється своєю мінімалістичною дизайном та надає клієнтам можливість ознайомитися з продуктами, переглядати фотографії, читати відгуки та замовляти косметику онлайн. Головну сторінку інтернет-магазину зображено на рис. 1.2 .

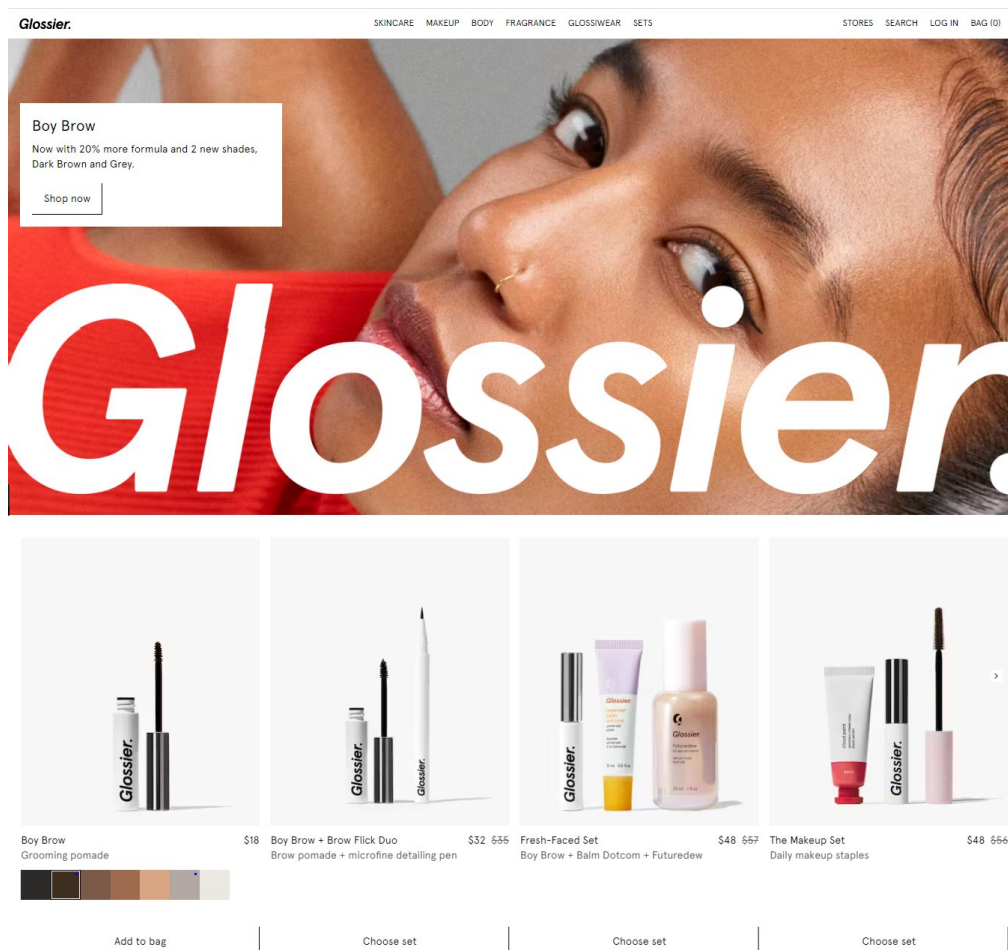


Рис. 1.2 Головна сторінка інтернет-магазину Glossier

Ці приклади ілюструють різноманітність підходів до веб-сайтів в косметичній галузі і демонструють, як електронна комерція через веб-сайти може допомогти підприємствам залучати та обслуговувати клієнтів у цифровому просторі.

Аналізуючи існуючі додатки та функціонал сучасних магазинів, можна виокремити деякі аспекти, які є ключовими для успішної реалізації веб-додатку. Одним з таких аспектів є мобільна сумісність, оскільки все більше користувачів здійснюють покупки з мобільних пристроїв. Крім того, важливо мати зручний інтерфейс, який дозволяє легко знаходити та ознайомлюватися з продуктами, а також здійснювати швидке та безпечне оформлення замовлень.

Проте, варто враховувати, що не всі функції та можливості, що доступні на існуючих веб-сайтах, можуть бути необхідні для кожного підприємства.

Важливо аналізувати переваги та недоліки існуючих додатків та веб-сайтів та пристосовувати їх до конкретних потреб підприємства, забезпечуючи оптимальну функціональність та зручність для клієнтів.

Один з недоліків, якого можна знайти на цих веб-сайтах, це незадовільна швидкість завантаження сторінок. Це може стати серйозною перешкодою для користувачів, які очікують швидкого та безперебійного досвіду під час покупок.

Причини такого недоліку можуть бути різними. Наприклад, сайт може містити велику кількість важких зображень, відео або анімації, які збільшують час завантаження сторінок. Також, недостатньо оптимізований код або неправильне використання технологій можуть вплинути на продуктивність сайту.

Це є недоліком, оскільки довгий час завантаження може викликати незадоволення користувачів, а також знижувати ймовірність успішного завершення покупки. Люди часто мають обмежений час та вимагають швидкості від інтернет-магазину. Через це даний недолік повинен бути вирішено як один із важливих, тому що це може призвести до втрати потенційних клієнтів, низької конверсії та загального негативного враження від веб-сайту.

1.2. Призначення розробки та галузь застосування

В якості об'єкта для впровадження розроблюваної веб-орієнтованої інформаційної системи для автоматизації діяльності підприємства в сфері електронної комерції розглядається створення веб-сайту магазину «BioBeauty», спрямованого на продаж косметики ручної роботи.

Основні вимоги, що ставляться до сучасного інтернет-магазину, включають такі аспекти:

– мобільну сумісність: веб-сайт повинен бути оптимізований для відображення на різних пристроях, таких як смартфони та планшети;

- система повинна мати можливість відстежувати наявність товарів на складі та автоматично оновлювати інформацію про наявність товару на веб-сайті;

- веб-сайт має мати зручний та інтуїтивно зрозумілий каталог, який дозволяє клієнтам швидко знайти потрібний товар;

- веб-сайт повинен мати можливість оформлення замовлення з вказанням варіантів доставки та розрахунку вартості;

- забезпечення конфіденційності та безпеки даних клієнтів є важливою складовою електронної комерції.

Після аналізування вищевказаних вимог, було зроблено висновок, що в проекті обов'язково буде реалізовано такі елементи:

- вітрина інтернет-магазину, що складатиметься з сукупності сторінок з детальним описом та фотографіями продуктів;

- функція пошуку, яка дозволяє користувачам швидко знайти потрібний товар за ключовими словами, категоріями, характеристиками тощо;

- корзина покупок, де клієнти можуть збирати та переглядати вибрані товари перед оформленням замовлення;

- система оформлення замовлення, яка дозволяє клієнтам вказати кількість товарів, обрати варіанти доставки, розрахунку вартості та ввести контактну інформацію;

- інтеграція з соціальними медіа, щоб клієнти могли легко ділитися продуктами та акціями магазину зі своїми друзями та отримувати зворотний зв'язок через ці платформи;

- адаптивний дизайн, що забезпечує зручний перегляд та навігацію на веб-сайті з різних пристроїв, включаючи комп'ютери, планшети та мобільні телефони [3].

Головна сторінка є ключовою в інтерфейсі користувача, тому її дизайн та функціональність мають бути добре продуманими, щоб забезпечити

користувача достатньою кількістю необхідної інформації, яка відповідає консультації в реальному магазині.

Прототип головної сторінки було створено за допомогою сервісу Figma. На ньому продемонстровано зовнішній вигляд головної сторінки інтернет магазину. Прототип головної сторінки зображено на рис. 1.3 .

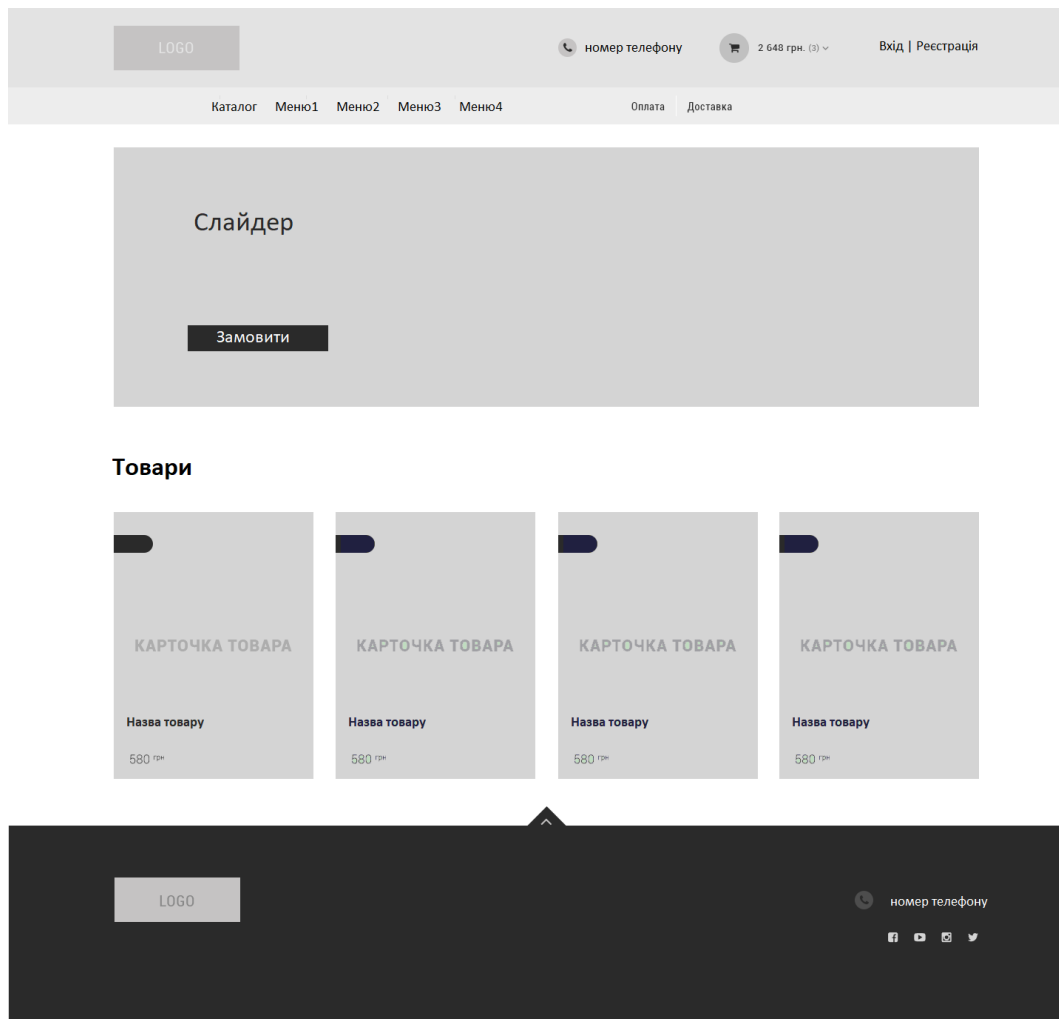


Рис. 1.3 Прототип головної сторінки

Головна сторінка містить:

- логотип, який представлений у формі стилізованого тексту, що відображає бренд магазину;
- система навігації, яка дозволяє користувачеві легко переходити між різними розділами і категоріями товарів;

- список акцій та спеціальних пропозицій, щоб заохотити користувачів до покупок;

- «підвал» та «футер», де розміщені додаткові посилання, контактна інформація та інші корисні елементи.

Розроблена інформаційна система має наступні цілі:

- інтегрувати елементи прямого маркетингу та традиційної торгівлі, створюючи потужну комерційну платформу;

- забезпечити зручний і простий процес пошуку, перегляду та придбання товарів;

- гарантувати безпеку та захист конфіденційної інформації клієнтів під час здійснення транзакцій та передачі даних;

- забезпечити можливість зворотного зв'язку та підтримки клієнтів для вирішення запитань і вирішення проблем.

1.3. Підстава для розробки

Підставами для розробки (виконання кваліфікаційної роботи) є:

- ОПП за спеціальністю 121 «Інженерія програмного забезпечення»;
- навчальний план та графік навчального процесу;
- наказ ректора Національного технічного університету «Дніпровська політехніка» № 350-с від 16.05.2023 р;

- завдання на кваліфікаційну роботу на тему «Розробка веб-додатку для інтернет-магазину косметики ручної роботи з використанням бібліотеки React».

1.4. Постановка завдання

В даній кваліфікаційній роботі розглядається створення інформаційної веб-орієнтованої системи для автоматизації діяльності підприємства в сфері електронної комерції.

Інформаційна система створюється для надання широкого спектру послуг

і інформації. Аудиторію інформаційної системи становлять люди у віці від 16 до 70 років.

Тематикою інформаційної системи є реалізація продуктів і послуг, пов'язаних з косметикою ручної роботи.

Інформаційна система повинна задовольняти наступним основним вимогам: висока швидкість завантаження сторінок; максимальна зручність роботи із системою для користувача; оптимізація сторінок під пошукові системи; легкість сприйняття інформації; простота й повнота управління змістом.

Створення й розробка інформаційної системи повинна включати наступні етапи:

- визначення цілей та завдань інформаційної системи, а також затвердження технічного завдання для її розробки;
- аналіз вимог та потреб користувачів, вивчення ринку та конкурентних рішень для визначення оптимальних функціональних можливостей системи;
- проектування архітектури інформаційної системи, розробка схеми бази даних та інтерфейсу користувача;
- розробка програмного забезпечення, включаючи написання коду, тестування та виправлення помилок;
- інтеграція різних компонентів системи, забезпечення взаємодії між ними та тестування їх спільної роботи;
- впровадження інформаційної системи на робочій платформі, підготовка до її експлуатації;
- після впровадження та супроводження системи, виявлення та усунення можливих проблем та помилок, а також надання користувачам необхідної підтримки.

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

Для досягнення поставленої в роботі мети в інформаційній системі, що розробляється, необхідно забезпечити наступні функціональні характеристики:

– Час відгуку на запит: система повинна бути добре оптимізованою і забезпечувати швидкий час відгуку на запити користувачів. Запити на перегляд товарів, додавання до кошика, оплату та інші мають оброблятися миттєво або у максимально короткий час.

– Надійність і безперебійна робота: веб-сайт має працювати стабільно та безперебійно протягом тривалого періоду часу. Необхідно забезпечити, щоб веб-сайт міг працювати без перерви протягом не менше 5 днів підряд, уникати відмов і аварій.

– Масштабованість: система повинна бути здатна працювати ефективно навіть при великому навантаженні та великій кількості одночасних користувачів. Масштабованість дозволить забезпечити стабільну роботу навіть у пікові періоди навантаження.

– Захист персональних даних: система повинна забезпечити надійний захист персональних даних користувачів, таких як адреси електронної пошти, номери телефонів та інше.

– Зручний інтерфейс: веб-сайт повинен мати інтуїтивно зрозумілий та зручний інтерфейс для користувачів. Легка навігація, зрозумілі іконки та елементи керування допоможуть забезпечити комфортний досвід покупок.

– Функціонал каталогу товарів: система повинна мати повноцінний та функціональний каталог товарів з можливістю швидкого пошуку, фільтрації та сортування. Користувачам має бути легко знаходити потрібні товари та ознайомлюватися з їхніми характеристиками.

– Зручний процес оформлення замовлення: система повинна забезпечувати зручний та простий процес оформлення замовлення з

можливістю вибору варіантів доставки та розрахунку вартості. Покупці повинні мати можливість легко виконувати всі необхідні кроки для завершення покупки.

1.5.2. Вимоги до інформаційної безпеки

Під інформаційною безпекою розуміють стан захищеності систем обробки і зберігання даних, при якому забезпечено конфіденційність, доступність і цілісність інформації, використання й розвиток в інтересах громадян або комплекс заходів, спрямованих на забезпечення захищеності інформації особи, суспільства і держави від несанкціонованого доступу, використання, оприлюднення, руйнування, внесення змін, ознайомлення, перевірки запису чи знищення. Забезпечення інформаційної безпеки передбачає використання різних заходів, які спрямовані на запобігання загрозам та забезпечення захищеності інформації від різноманітних атак та порушень безпеки [4].

Можливі види загроз та протистояння їм виникненню [4]:

– Несанкціонований доступ до клієнтських даних: для захисту від цього виду загрози необхідно використовувати механізми автентифікації та авторизації, щоб забезпечити тільки обмеженому колу авторизованих користувачів доступ до конфіденційної інформації.

– Вразливості програмного забезпечення: важливо проводити регулярне оновлення програмного забезпечення, щоб усунути виявлені вразливості. Використання захисного програмного забезпечення, такого як фаєрволи та антивіруси, також допоможе запобігти вторгненням.

– Витік інформації: застосування шифрування даних під час їх передачі, зберігання даних в захищених середовищах та використання політик доступу до інформації допоможуть уникнути витоку конфіденційної інформації.

1.5.3. Вимоги до складу та параметрів технічних засобів

Для нормального функціонування веб-орієнтованої інформаційної системи, повинні виконуватися певні вимоги до технічних засобів.

Для клієнтської частини:

- процесор з тактовою частотою не менше 2.4 ГГц;
- оперативна пам'ять не менше 4 GB;
- 5 Гб вільного місця на жорсткому диску.

Для серверної частини:

- процесор з тактовою частотою не менш 2.4 ГГц;
- оперативна пам'ять не менше 4 GB;
- 10 Гб вільного місця на жорсткому диску.

Наведені вище технічні характеристики є рекомендованими, тобто при наявності технічних засобів не нижче зазначених, розроблений програмний виріб буде функціонувати відповідно до вимог щодо надійності, швидкості обробки даних і безпеки, висунутими замовником.

1.5.4. Вимоги до інформаційної та програмної сумісності

Для нормального функціонування програми необхідно, щоб програмне забезпечення обчислювальної машини, на якій буде функціонувати веб-орієнтована підсистема, відповідало наступним вимогам:

- операційна система Unix, Linux, Microsoft Windows 7/8/10/11;
- браузер Інтернет (Microsoft Internet Explorer, MozillaFireFox, Opera, Google Chrome);

Веб-орієнтована інформаційна система має бути реалізовано на мові програмування HTML, SCSS, JavaScript для верстання веб-сторінок, бібліотеки React для написання клієнтських сценаріїв, бібліотеки Node та Express для серверної частини, з використанням СУБД MySQL.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

2.1. Функціональне призначення програми

В результаті виконання даної кваліфікаційної роботи буде створено додаток, який буде гнучко працювати в різних середовищах. Це означає, що додаток буде запускатися в браузері та зможе працювати незалежно від операційної системи та пристрою, на якому він використовується.

Кінцевий продукт матиме наступний функціонал:

- реєстрація та аутентифікація користувачів: додаток забезпечить можливість реєстрації нових користувачів та входу в систему за допомогою облікових записів;
- каталог товарів: додаток надаватиме зручний спосіб перегляду та пошуку товарів, представлених в магазині;
- можливість сортування та пошуку товарів;
- додавання товарів до кошика: користувачі матимуть можливість додавати товари до віртуального кошика для подальшого оформлення замовлення;
- додавання товарів в обране: користувачі матимуть можливість додавати товари в обране задля подальшого збереження товарів які сподобались;
- оформлення замовлення: додаток дозволить зручно оформляти замовлення, включаючи вказування адреси доставки та обраного способу оплати;
- зворотний зв'язок та підтримка клієнтів: додаток надасть можливість звертатися до підтримки клієнтів для вирішення питань та вирішення проблем;
- надання віддаленого доступу до застосунку через веб-браузер через девайс користувача.

Веб-застосунок повинен надавати необхідну користувачам інформацію в зручному вигляді, мати зрозумілий та зручний інтерфейс, а також сприяти взаємодії продавця з клієнтом який використовує ПК так і з клієнтом що використовує телефон.

2.2. Опис застосованих математичних методів

У розробленій системі використовується лише математичний метод складання загальної суми до сплати в кошику, або під час оформлення замовлення.

2.3. Опис використаної архітектури та шаблонів проектування

Веб-додаток працює за архітектурою клієнт-сервер. Серверна сторона програми реалізована з використанням веб-фреймворку Express.js, який забезпечує швидку та ефективну обробку HTTP-запитів від клієнтів. Express.js базується на платформі Node.js і надає зручну основу для розробки веб-додатків.

Express.js - це швидкий та ефективний веб-фреймворк, побудований на основі платформи Node.js. Він надає розробникам зручні інструменти для створення веб-додатків, особливо зосереджених на обробці HTTP-запитів.

Одна з ключових особливостей Express.js - це його мінімалістичний підхід, що дозволяє розробникам вибирати лише ті функціональні компоненти, які необхідні для конкретного проекту. Він надає гнучкість і свободу вибору, що дозволяє розробникам будувати веб-додатки за власними потребами [5].

Для зберігання та керування даними програма використовує базу даних MySQL. MySQL є однією з найпоширеніших та надійних реляційних баз даних, яка надає широкий набір функціональних можливостей для ефективної роботи з даними.

MySQL дозволяє створювати таблиці, що використовуються для організації даних в структурований спосіб. Кожна таблиця має свою структуру, включаючи назви колонок (полів) та їх типи даних. За допомогою SQL-запитів (Structured Query Language), яка є мовою запитів для реляційних баз даних, можна виконувати різноманітні операції з даними [6].

Операції CRUD (створення, читання, оновлення, видалення) є основними операціями, які можна виконувати з даними в базі MySQL. Використовуючи SQL-запити, можна створювати нові записи в таблицях (створення), отримувати дані з таблиць (читання), змінювати вже існуючі записи (оновлення) та видаляти непотрібні записи (видалення). Ці операції дозволяють ефективно керувати даними та забезпечувати їх актуальність та цілісність [7].

MySQL також підтримує виконання запитів, які дозволяють отримувати потрібну інформацію з бази даних. Запити можуть бути простими, які повертають всі записи з певної таблиці, або складнішими, які виконують об'єднання (JOIN) таблиць, фільтрування результатів за певними умовами (WHERE), агрегування (GROUP BY) та інші операції для отримання потрібних даних.

MySQL також надає можливості для управління зв'язками даних, які дозволяють пов'язувати дані між різними таблицями. Це забезпечує ефективну організацію даних і можливість виконувати складні запити, які пов'язують дані з різних таблиць [6].

Вибір бази даних MySQL для зберігання даних у хмарному сервісі Free Sql Database забезпечує надійність та масштабованість. Хмарний сервіс дозволяє зберігати дані в безпечному та доступному середовищі, а також підтримує можливість масштабування ресурсів бази даних відповідно до зростання потреб програми. Таким чином, база даних MySQL спільно з хмарним сервісом надають потужну та надійну інфраструктуру для зберігання та керування даними у програмному додатку.

Для адміністрування бази даних MySQL використовується інструмент phpMyAdmin, який надає зручний інтерфейс для створення та зміни таблиць,

виконання запитів, керування даними та оптимізації продуктивності бази даних.

phpMyAdmin — вебдодаток з відкритим кодом, написаний мовою PHP із графічним веб-інтерфейсом для адміністрування бази даних MySQL або MariaDB. Середовище phpMyAdmin зображено на рис. 2.1 . phpMyAdmin дозволяє через браузер здійснювати адміністрування сервера MySQL, запускати запити SQL, переглядати та редагувати вміст таблиць баз даних. Ця програма користується великою популярністю у веброзробників, оскільки дозволяє керувати базою даних MySQL без вводу SQL команд через дружній інтерфейс і з будь-якого комп'ютера під'єданого до інтернету без необхідності встановлення додаткового програмного забезпечення [8].

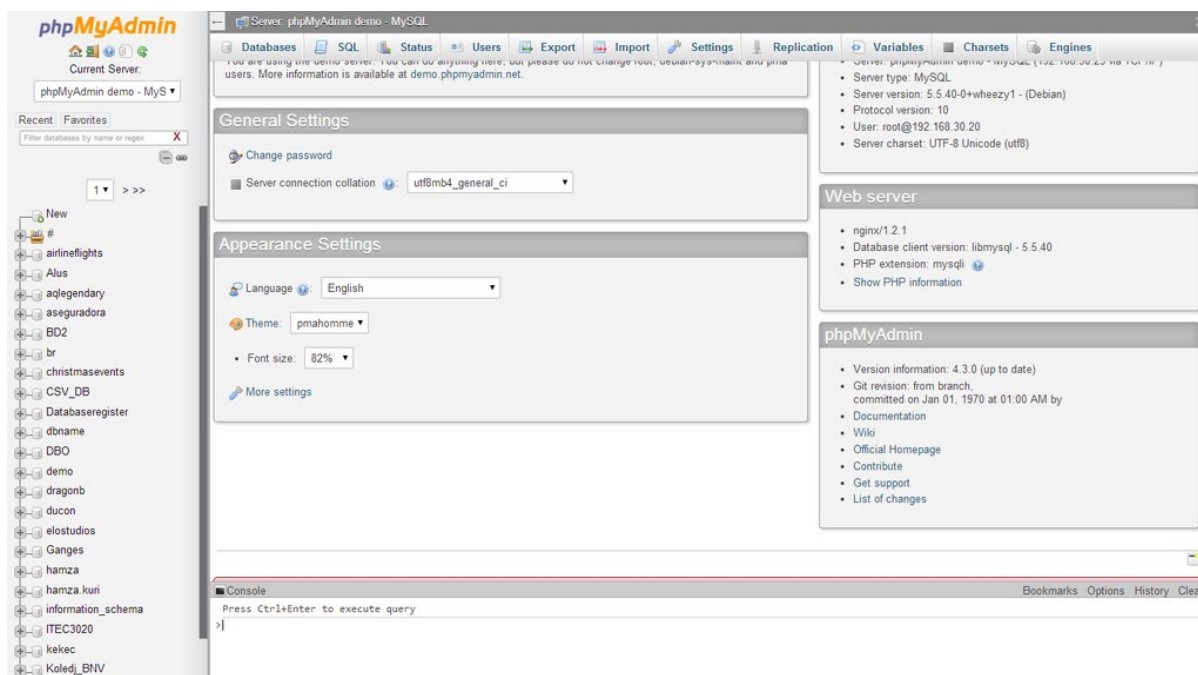


Рис. 2.1 Зовнішній вигляд середовища phpMyAdmin

Архітектура з використанням Express.js, MySQL та phpMyAdmin дозволяє забезпечити швидку та надійну комунікацію між клієнтами та сервером, а також зручне та ефективне зберігання та керування даними. Express.js надає потужні можливості для обробки запитів та маршрутизації програми, в той час

як MySQL та phpMyAdmin забезпечують потрібні функції для роботи з базою даних. Ця архітектура є популярною та надійною для розробки та управління веб-додатками. Схема роботи архітектури зображено на рис. 2.2 .

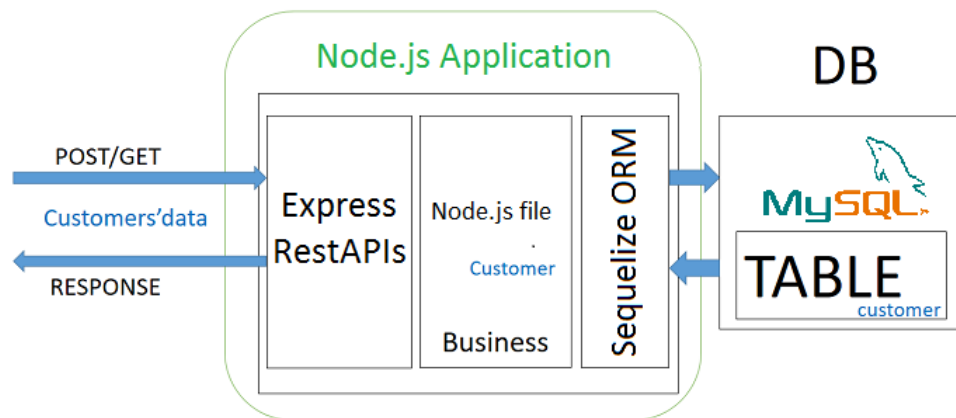


Рис. 2.2 Схема роботи архітектури

2.4. Опис використаних технологій та мов програмування

Додаток було розроблено з використанням мови програмування JavaScript разом із бібліотекою React.JS для створення інтерфейсу користувача. Мова розмітки HTML була використана для структурування та відображення веб-сторінок, надаючи інструкції браузеру щодо того, як відтворити вміст. HTML є найпоширенішою мовою розмітки завдяки своїй популярності та стандартизації. Він використовує теги, укладені в кутові дужки, щоб визначити початок і кінець кожного елемента, а браузери інтерпретують ці теги для відображення веб-сторінки. Однак проблеми сумісності можуть виникнути в різних системах, версіях і браузерах, що призведе до неправильного відтворення, якщо в коді HTML є помилки [9].

Додаток використовує HTML5, п'яту версію HTML, випущену в 2017 році, метою якої було досягнення незалежності від платформи та узгодженості під час відтворення на різних пристроях. У HTML5 представлені функції для

обробки мультимедійних вимог, наприклад вбудовування відео та аудіо безпосередньо в код. Він також покращив перевірку форм, дозволивши визначати типи даних у полях введення за допомогою самого HTML замість того, щоб покладатися виключно на JavaScript. Існують і інші відомі мови розмітки, такі як XHTML і XML, але вони мають певні обмеження та рідше використовуються порівняно з HTML [9].

Для керування візуальним представленням документа використовуються каскадні таблиці стилів . CSS — це мова, що складається з ієрархічних правил, які визначають, як елементи мають відображатися на екрані. Це дозволяє відокремити зміст і дизайн, полегшуючи керування та зміну зовнішнього вигляду веб-сторінок. Хоча CSS забезпечує гнучкість у стилях, оновлення кількох документів під час зміни дизайну може стати громіздким, особливо для великих веб-сайтів [10].

Щоб спростити код CSS і підвищити його абстракцію, був використаний модуль SASS (Syntactically Awesome Style Sheets). SASS надає два синтаксиси: Sass і SCSS. Синтаксис Sass пропускає фігурні дужки та використовує відступи для вкладених елементів, усуваючи потребу в крапках з комою після кожної властивості. Однак синтаксис SCSS, який дуже нагадує CSS, був обраний для розробки через його звичність і читабельність серед більшості розробників. SCSS розширює CSS, додаючи такі функції, як константи та функції, що дозволяє краще підтримувати узгодженість стилів у великому наборі стилів. Константи дозволяють визначати значення, які можна повторно використовувати в стилях, а функції розширюють можливості блоків атрибутів стилю [11]. Порівняння синтаксису CSS та SCSS наведено на рис. 2.3 .

Таким чином, додаток створено з використанням JavaScript і React JS, а HTML виступає як мова розмітки для структурування та відображення вмісту. CSS використовувався для стилізації, а модуль SASS, зокрема синтаксис SCSS, використовувався для покращення зручності обслуговування та організації коду CSS за допомогою таких функцій, як константи та функції.

CSS	SCSS
<pre> 1 section { 2 height: 100px; 3 width: 100px; 4 } 5 6 section .class-one { 7 height: 50px; 8 width: 50px; 9 } 10 11 section .class-one .button { 12 color: #b6b1b1; 13 } </pre>	<pre> 1 section { 2 height: 100px; 3 width: 100px; 4 } 5 6 .class-one { 7 height: 50px; 8 width: 50px; 9 } 10 11 .button { 12 color: #b6b1b1; 13 } </pre>

Рис. 2.3 Порівняння CSS та SCSS

JavaScript - мова програмування, що дозволяє реалізувати ряд складних рішень в web-документах. Вона допомагає зробити сторінки сайту більш інтерактивними, обробляє дії користувачів сайту. Це об'єктно-орієнтована клієнтська мова, яка підтримується додатками, що працюють з дизайном сайту. JavaScript став ще більш популярним в середовищі девелоперів, коли з'явилася AJAX-технологія, що призвело до нового етапу в розробці сайтів [12].

JavaScript широко використовується на більшості веб-сайтів і вважається однією з найпопулярніших мов програмування. Оскільки інтерпретатори JavaScript вбудовані в сучасні браузері на різних пристроях, його використання стало повсюдним. Останні тенденції свідчать про те, що кількість JavaScript-розробників в Україні, згідно з опитуванням порталу DOU (рис. 2.4), стабільно зростає. Це можна пояснити безперервною еволюцією мови програмування, яка сприяла зростанню її популярності серед розробників. Цього року найпопулярнішою мовою залишається JavaScript - 19% розробників пишуть нею комерційні проєкти [13].

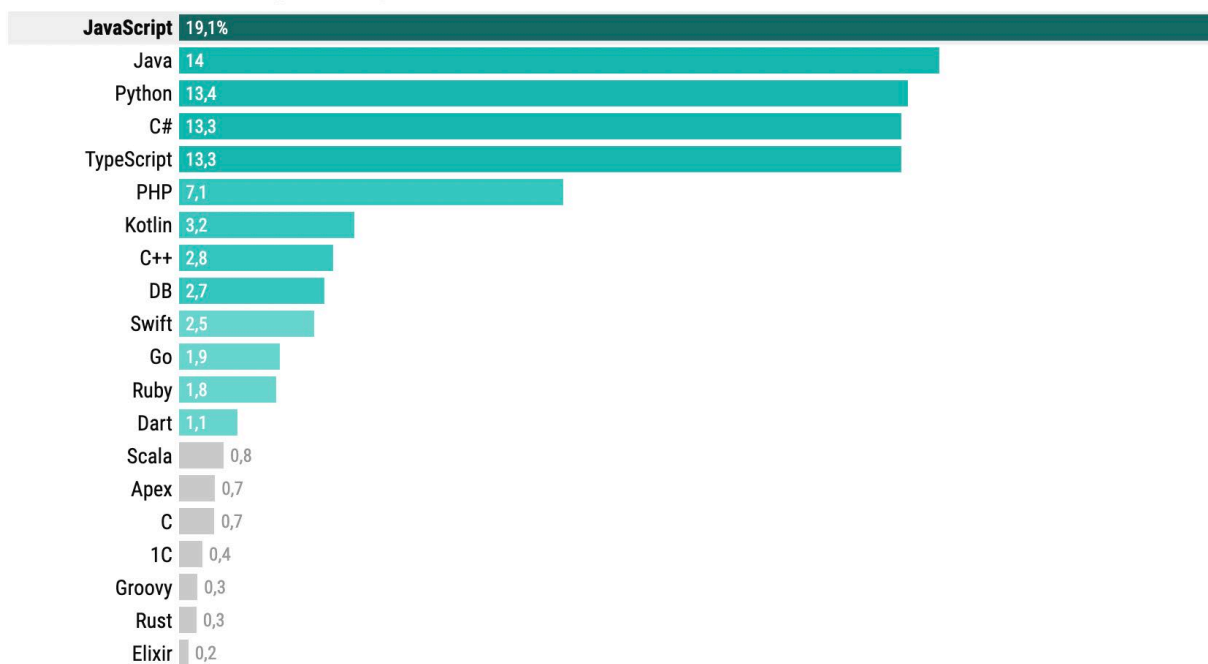


Рис. 2.4 Опитування по використанню мов програмування

JavaScript - це фундаментальна мова програмування, яка у поєднанні з HTML і CSS формує основу сучасних веб-інтерфейсів. Він пропонує безліч можливостей, дозволяючи розробникам зберігати та маніпулювати даними, реагувати на певні дії користувача, динамічно оновлювати вміст і обробляти мультимедійні елементи, такі як відео та анімація. Завдяки тому, що JavaScript виконується на стороні клієнта у веб-браузерах, він користується широкою популярністю та може безперервно працювати в різних операційних системах, забезпечуючи кросплатформну сумісність веб-додатків [14].

JavaScript знаходить широке використання в різних доменах. Наприклад, він відіграє важливу роль у активації сценаріїв AJAX, які сприяють асинхронному обміну даними між браузерами користувачів і ресурсами сервера, покращуючи взаємодію з користувачем. Крім того, у Comet використовується JavaScript - техніка, за допомогою якої сервери можуть надсилати дані браузерам без додаткових запитів, забезпечуючи оновлення та взаємодію в реальному часі. Він також використовується в операційних системах браузерів, закладках, скриптах, серверних програмах, розробці

мобільних програм, віджетах і навіть створенні автономних програм, включаючи прості ігри.

Універсальність і широкі можливості застосування JavaScript сприяють його популярності серед розробників. Він має надійну сумісність з усіма основними браузером, що робить його надійним вибором для проектів веб-розробки. Крім того, він легко інтегрується з популярним програмним забезпеченням для дизайну, таким як Adobe Photoshop, Illustrator, InDesign і Dreamweaver, розширюючи його охоплення за межі веб-розробки.

Сильні сторони JavaScript виходять за рамки основних функцій. Це дозволяє розробникам виконувати перевірку форм на стороні клієнта, покращуючи взаємодію з користувачем, виявляючи помилки перед надсиланням даних на сервер. Мова також дозволяє створювати візуально привабливі та інтерактивні веб-сторінки, привертаючи увагу користувачів і посилюючи залучення. Крім того, JavaScript полегшує складні обчислення, надаючи необхідні інструменти для маніпулювання та обробки даних [15].

Таким чином, JavaScript - це універсальна та широко використовувана мова програмування, яка вдосконалює веб-інтерфейси, пропонуючи розробникам безліч можливостей. Його сумісність, широкий вибір програм і здатність легко інтегруватися з різними інструментами роблять його популярним вибором для сучасних проектів веб-розробки.

Особливо вирішальну роль у спрощенні розробки та інтеграції компонентів у проекти програмного забезпечення відіграють бібліотеки та фреймворки. Вони служать основою, яка дозволяє розробникам швидко створювати цільові програми, легко додаючи та налаштовуючи компоненти. Такий підхід скорочує час і витрати на розробку, дозволяючи розробникам зосередитися на основній логіці, а не на повторному винаході загальних блоків і компонентів.

Серед численних фреймворків і бібліотек JavaScript, доступних сьогодні, кілька здобули значну популярність. Провідними конкурентами в екосистемі JavaScript є React JS, Vue JS і Angular JS. Якщо подивитись на опитування

розробників Stack Overflow (рис. 2.5), то можна проаналізувати популярність React JS серед інших [16]. Аналізуючи тенденції популярності за останні п'ять років, React виявляється найбільш поширеним фреймворком, за яким слідує зростаюча популярність Vue JS.

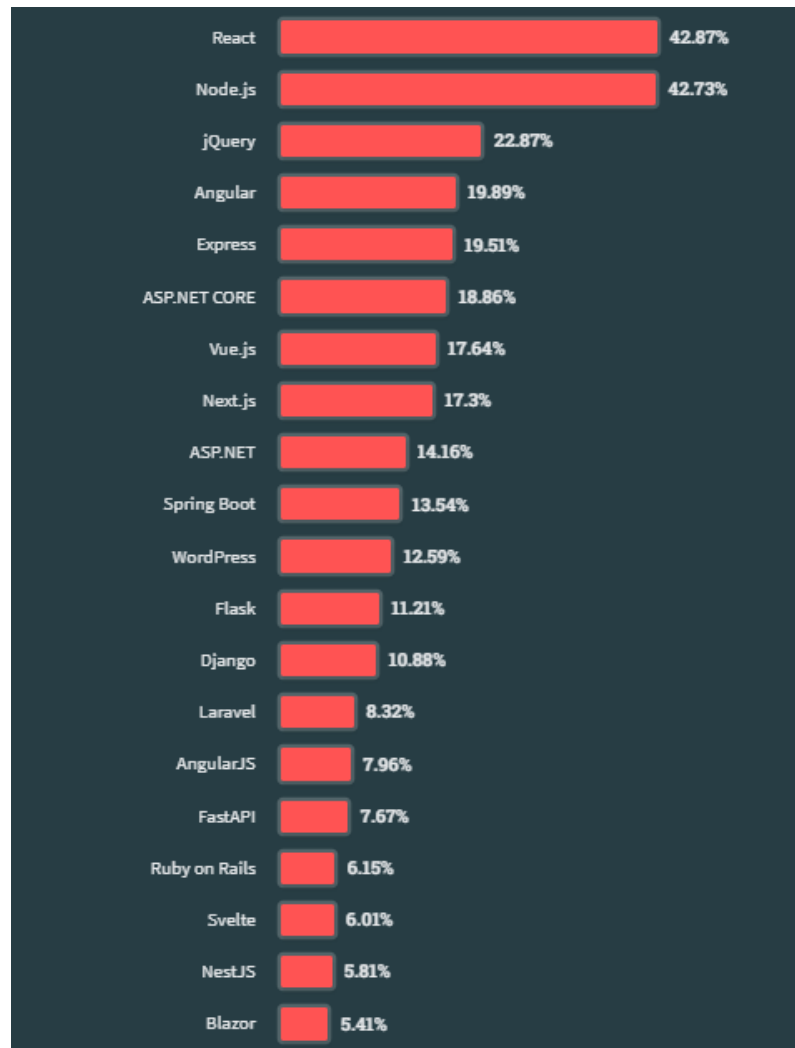


Рис. 2.5 Опитування по використанню веб-фреймворків

React JS розроблений і підтримуваний Facebook, набув величезної популярності з моменту свого створення. Спочатку він використовувався на веб-сайті Facebook для обробки різних практичних блоків. І у 2013 році Facebook випустив вихідний код React для громадськості. React представляє концепцію компонентів, де весь код міститься в таких функціях, як функція

App. Компоненти представляють фрагменти інтерфейсу, які інкапсулюють як розмітку, так і пов'язану логіку. Хоча компонентний підхід виник ще до React, його поєднали з декларативним програмуванням, створивши потужну комбінацію. Комбінуючи компоненти, розробники можуть створювати повні інтерфейси веб-додатків [17].

Однією з помітних переваг React є його здатність динамічно оновлювати інтерфейс без необхідності перезавантажувати сторінку. Це призводить до швидшої взаємодії користувача, наприклад надсилання форм, фільтрації сторінок або додавання продуктів у кошик для покупок, оскільки під час перемикання між сторінками не завантажуються дані. React також інтегрується з бібліотеками стану та маршрутизації, такими як Redux або React Router, дозволяючи компонентам автоматично повторно відтворюватися, коли змінюються реактивні змінні. Цей детальний підхід до візуалізації дозволяє розробникам оновлювати окремі елементи, не оновлюючи всю сторінку, що забезпечує ефективну обробку. Іншою ключовою особливістю React є використання JSX, розширення синтаксису JavaScript, яке полегшує бездоганну інтеграцію розмітки HTML у код React. Дуже нагадуючи HTML, JSX спрощує процес розробки, надаючи знайомий синтаксис для опису інтерфейсу. Порівняння JS та JSX наведено на рис.2.6 [17].

✓ index.jsx

```
import React from 'react'

const Component = () => {
  |   return <h1>Це JSX-розмітка</h1>
}

```

✗ index.jsx → index.js

```
const Component = () => {
  |   return 'А це вже просто рядок'
}

```

Рис. 2.6 Порівняння JS та JSX

Для ілюстрації уявіть собі створення веб-додатку для онлайн-магазину за допомогою React. Завдяки ефективному рендерингу React ви можете миттєво оновлювати лічильник кошика, коли користувач додає продукт, не оновлюючи всю сторінку. Повна інтеграція JSX дає змогу писати код, подібний до HTML, у ваших компонентах React, полегшуючи визначення елементів інтерфейсу та керування ними.

У більшості випадків важливо, щоб редактор коду міг правильно інтерпретувати та виділяти код, який містить як звичайну розмітку JavaScript, так і JSX. Хоча технічно можливо написати код без розширення .jsx і забезпечити правильну компіляцію та роботу програми, корисно дотримуватися правильних правил синтаксису. Це допомагає розробникам ідентифікувати та розуміти наявність розмітки JSX у файлі, забезпечуючи ясність під час роботи з інтерфейсом у коді JavaScript. Крім того, JSX у React допомагає надавати чіткіші повідомлення про помилки та попередження, що високо цінується розробниками [18].

Таким чином, бібліотеки та фреймворки, такі як React JS, зробили революцію у веб-розробці, спростивши компонентне програмування, увімкнувши динамічні оновлення та запропонували інтуїтивно зрозумілий синтаксис. Використовуючи ці інструменти, розробники можуть ефективно створювати надійні та інтерактивні веб-програми.

Node.js - це однопотокowe кросплатформове середовище виконання з відкритим вихідним кодом і бібліотека, яка використовується для запуску веб-додатків, написаних на JavaScript, поза браузером клієнта [19].

Простіше кажучи, Node.js - це програмне середовище, яке дозволяє запускати програми, написані мовою Javascript, поза браузером. Історично програми, написані на Javascript, на відміну інших мов програмування, можна було запускати лише у браузерах, які мали спеціальний вбудований движок виконання коду цієї мови. Поза браузером Javascript, можна сказати, не працював.

При розробці Node.js за основу було взято двигун виконання JavaScript під назвою V8, який був створений компанією Google і використовувався в браузері Google Chrome. Оскільки після створення Node.js Javascript код можна запустити фактично в будь-якому середовищі, за допомогою цієї бібліотеки можна написати не лише фронтенд, а й серверну частину веб-програми.

Простіше кажучи, це означає, що цілі сайти тепер можуть працювати з використанням єдиного «стека», що робить розробку та обслуговування набагато швидшою та легшою, дозволяючи зосередитися на досягненні бізнес-цілей проекту [18].

Node.js також отримує переваги від великої екосистеми модулів і бібліотек, доступних через його менеджер пакетів, npm. Ця обширна колекція інструментів дає розробникам змогу швидко створювати та розгортати веб-програми, використовуючи вже існуючі рішення та компоненти. Це не лише прискорює швидкість розробки, але й покращує якість коду та зменшує ймовірність помилок [20].

Крім того, процвітаюча спільнота, що оточує Node.js, забезпечує постійну підтримку, активний розвиток і багатство ресурсів. Це гарантує, що Node.js залишається в курсі останніх веб-технологій і стандартів безпеки, дозволяючи розробникам використовувати нові функції та вдосконалення.

Використовуючи Node.js, розробники можуть створювати ефективні, масштабовані та високопродуктивні веб-програми, які використовують увесь потенціал JavaScript у всьому стеку програмного забезпечення. Це дає їм змогу зосередитися на досягненні бізнес-цілей свого проекту, водночас користуючись швидкістю, гнучкістю та простотою, яку Node.js привносить у веб-розробку.

Express.js - це мінімалістичний і гнучкий фреймворк веб-додатків для Node.js. Він надає набір надійних функцій і утиліт, які допомагають спростити процес створення веб-додатків і API. За допомогою Express.js розробники можуть швидко й ефективно створювати програми на стороні сервера, використовуючи його інтуїтивно зрозумілу систему маршрутизації, підтримку проміжного програмного забезпечення та методи утиліт HTTP [20].

Express.js дозволяє розробникам визначати маршрути для обробки різних запитів HTTP, таких як GET, POST, PUT і DELETE, що полегшує створення кінцевих точок, які обробляють різні функції програми. Він забезпечує чіткий і організований підхід до обробки запитів і керування потоком даних.

Підтримка проміжного програмного забезпечення в Express.js дозволяє розробникам додавати додаткові функції до своїх програм. Функції проміжного програмного забезпечення можна використовувати для обробки вхідних запитів, модифікації об'єктів запиту або відповіді, обробки помилок і виконання різноманітних інших завдань. Ця гнучкість дозволяє розробникам налаштовувати поведінку програм відповідно до своїх конкретних вимог [20].

Загалом Express.js широко відомий своєю простотою, гнучкістю та продуктивністю, що робить його популярним вибором серед розробників для створення веб-додатків та API за допомогою Node.js.

2.5. Опис структури програми та алгоритмів її функціонування

Структура проекту складається з папок backend та frontend, у кожній з них знаходиться відповідна частина створеного веб-додатку. Кожна з цих папок виконує свою функціональну роль, а також файлів, які відповідають за різні аспекти роботи проекту, такі як функціональність, стилі оформлення та адаптивний дизайн інтернет-магазину, серверну частину, а також конфігурацію проекту.

Структура додатку наведена на рис. 2.7 .



Рис. 2.7 Структура веб-додатку

Структура проекту:

- папка `backend` містить файл `server.js`, у якому реалізовано серверну логіку додатку;
- папка `ecommerce` містить клієнтську частину додатку, яка відповідає за інтерфейс користувача;
- папка `node_modules` має службове призначення та не несе сенсу для логіки проекту, в них знаходяться скомпільована програма та пакети для роботи React;

- папка `public` містить папку `img` в якій знаходяться всі зображення, які використовуються на сайті;
- файл `favicon.ico` є піктограмою, яка з'являється на вкладці браузера або на панелі закладок, коли веб-сайт завантажується;
- файл `index.html` служить точкою входу для програми React;
- файл `manifest.json` надає метадані та інформацію про конфігурацію веб-програми;
- файл `robots.txt` – індексний текстовий документ, який надає інструкції сканерам пошукової системи;
- папка `components` зберігає повторно використовувані елементи інтерфейсу для кращої організації коду та можливості повторного використання;
- файли `Card_c` відтворюють картку розділу каталогу та його загальне зображення;
- файли `Card_h` відтворюють картку товару;
- файли `Favorite` відтворює список товарів, які додані в обране;
- файли `Footer` представляють розділ нижнього колонтитула веб-сторінки;
- файли `Header` представляють розділ верхнього колонтитула веб-сторінки;
- файли `Slider` представляє карусель фотокарток з актуальними новинами;
- папка `pages` розташовані окремі сторінки, які призначені для використання і вбудовування на веб-сайті;
- файл `Authorisation.jsx` відтворює сторінку авторизації користувача;
- файл `Basket.jsx` відтворює сторінку корзини;
- файл `Content.jsx` відтворює сторінку з товарами;
- файл `Favorite.jsx` відтворює сторінку товарів які сподобались;
- файл `Home.jsx` відтворює головну сторінку веб-сайта;
- файл `Order.jsx` відтворює сторінку створення замовлення;

- файл `ProductPage.jsx` відтворює сторінку товару з його описом;
 - файл `Profile.jsx` відтворює сторінку профілю користувача;
 - файл `App.jsx` об'єднує всі React компоненти та сторінки;
 - файл `index.scss` – загальний файл стилів для усього додатку;
 - файл `index.js` – загальний функціональний файл, у якому відбувається монтування усього додатку;
- файл `.gitignore` містить в собі налаштування системи контролю версій Git;
- файл `package-lock.json` містить інформацію про проект та його залежності;
 - файли `package.json` містить точну інформацію про версії пакетів та їх залежності, які встановлені у проекті.

Опис структури бази даних.

Структура бази даних, що використовується в додатку, розробленому з використанням Node.js та Express.js, відповідає наступним принципам. Комунікація з базою даних відбувається шляхом виконання запиту GET до серверу, який повертає дані у форматі JSON. Кожен елемент бази даних має свій унікальний первинний ключ - атрибут або групу атрибутів, які служать єдиним ідентифікатором для кожного елемента.

Крім первинного ключа, база даних також містить інші атрибути, які містять дані про інформаційне відображення властивостей об'єкта. Ці атрибути надають додаткову інформацію про об'єкти і використовуються для їх відображення в інтерфейсі користувача.

Опис властивостей таблиць бази даних представлено у таблицях 2.1-2.7 .

Таблиця 2.1 містить в собі інформацію про властивості, які мають товари у веб-додатку.

Опис властивостей таблиці «Товари»

№	Поле	Опис
1.	id_product	Ідентифікатор товару
2.	id_category	Ідентифікатор категорії
3.	article	Артикул товару
4.	title	Назва товару
5.	info	Опис товару
6.	compound	Склад товару
7.	price	Ціна товару
8.	imgUrl1	Посилання на головне зображення товару
9.	imgUrl2	Посилання на друге зображення товару
10.	imgUrl3	Посилання на третє зображення товару
11.	hit	Хіт продаж

Таблиця 2.2 містить інформацію та властивості про користувача-клієнта.

Опис властивостей таблиці «Клієнти»

№	Поле	Опис
1.	id_client	Ідентифікатор користувача
2.	surname	Прізвище та ім'я
3.	email	Електронна пошта
4.	phone	Номер телефону
5.	city	Місто
6.	post	Пошта
7.	address	Адреса відділення
8.	password	Пароль від облікового запису

Таблиця 2.3 містить інформацію та властивості про категорії товарів.

Таблиця 2.3

Опис властивостей таблиці «Категорії»

№	Поле	Опис
1.	id_category	Ідентифікатор категорії
2.	title	Назва категорії
3.	imgUrl	Посилання на зображення категорії

Таблиця 2.4 містить інформацію та властивості про товари додані до кошика.

Таблиця 2.4

Опис властивостей таблиці «Кошик»

№	Поле	Опис
1.	id_cart	Ідентифікатор кошика
2.	id_client	Ідентифікатор клієнта
3.	id_prod	Ідентифікатор товару
4.	kol	Кількість товару

Таблиця 2.5 містить інформацію та властивості про товари які додані до обраного.

Таблиця 2.5

Опис властивостей таблиці «Обране»

№	Поле	Опис
1.	id_favorite	Ідентифікатор замовлення
2.	id_client	Ідентифікатор клієнта
3.	id_prod	Ідентифікатор товару

Таблиця 2.6 містить інформацію та властивості про замовлення.

Таблиця 2.6

Опис властивостей таблиці «Замовлення»

№	Поле	Опис
1.	id_order	Ідентифікатор замовлення
2.	id_client	Ідентифікатор клієнта
3.	id_prod	Ідентифікатор товару
4.	koll	Кількість товару
5.	price	Ціна за товар в момент замовлення
6.	num_ord	Номер замовлення
7.	dtime	Дата та час замовлення
8.	pay	Тип оплати

Таблиця 2.7 містить інформацію та властивості про слайди, які відображаються на головній сторінці.

Таблиця 2.7

Опис властивостей таблиці «Слайди»

№	Поле	Опис
1.	id_slide	Ідентифікатор слайду
2.	imgUrl	Посилання на зображення слайду

Структура бази даних зображена на рис. 2.8 .

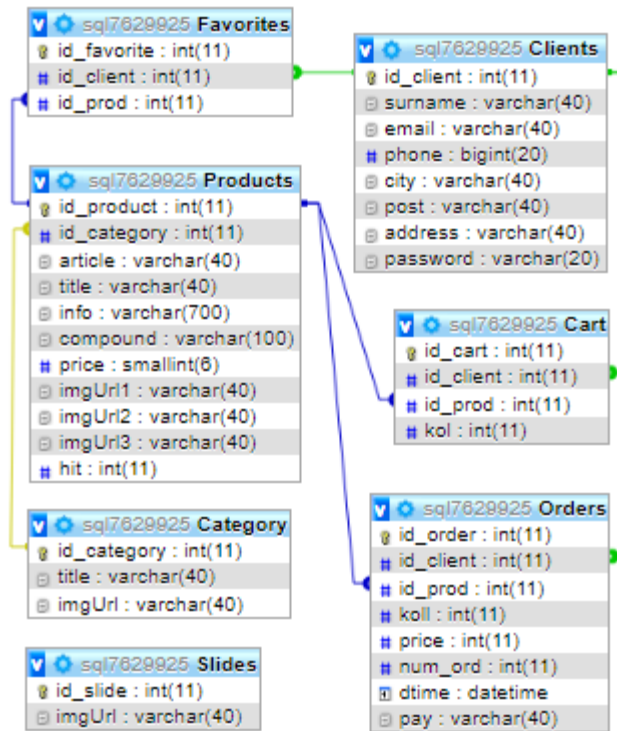


Рис.2.8 Структура бази даних

Для успішного запуску веб-додатку, потрібно виконати декілька кроків. Спочатку необхідно запустити серверну частину, яка побудована з використанням Express.js. Це дозволить серверу обробляти запити та надавати необхідні дані для вашого додатку. Щоб запустити серверну частину, потрібно викликати термінал у середовищі розробки Visual Studio Code, після чого перейти до папки серверної частини та викликати команду `node server.js`.

Після успішного запуску сервера, можна переходити до клієнтської частини. Спочатку потрібно зібрати всі компоненти та бібліотеки, необхідні для вашого проекту, використовуючи команду `npm install`. Це дозволить вам завантажити всі необхідні модулі та залежності. Для запуску клієнтської частини необхідно перейти до папки фронтенда та викликати команду у терміналі `npm start`.

Після успішного збирання модулів, ваш веб-додаток може почати взаємодіяти з компонентами. За допомогою фреймворка React, будується

віртуальне дерево DOM, на якому можуть бути монтувані компоненти. Крім того, інші бібліотеки також можуть бути інтегровані у ваш проект. Наприклад, SCSS-модулі перетворюються на звичайні CSS-стили з унікальними ключами для кожного компонента.

Після завантаження сторінки, за допомогою хуків `useEffect` або спеціальних функцій та запитів `Axios`, ваш додаток може отримувати, обробляти та зберігати дані з бази даних. Після завершення завантаження сторінки, `React` реагує на дії, які користувач здійснює у веб-додатку, та відображає відповідні дані та сторінки на запити користувача.

Розглянемо деякі ключові елементи функціонування веб-додатку.

Файл `index.html` є важливим компонентом веб-додатку і розташований у папці `public`. Він містить кореневий елемент, в який `React` монтує всі компоненти, утворюючи віртуальне DOM-дерево. Крім цього, `index.html` включає необхідні підключення та налаштування для коректного відображення сторінки у браузері. Цей файл виступає як точка входу для веб-додатку та забезпечує ініціалізацію `React`-додатку, його зв'язок зі сторінкою та взаємодію з іншими компонентами та скриптами.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta
      name="description"
      content="Thesis on the creation of a site for the sale of cosmetics of
a year's work,
      created with the help of the React.js library"
    />
    <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
    <title>BioBeauty - Натуральна косметика ручної роботи</title>
  </head>
  <body>
```

```
    <div id="root"></div>
  </body>
</html>
```

Файл `index.js`, розташований у папці `src`, відповідає за монтування всіх компонентів у головний елемент `div` у `html` файлі проекту за допомогою бібліотеки `React`. У цьому файлі виконується підключення необхідних модулів, таких як `React`, `ReactDOM`, стилі та компоненти додатку. Також використовується бібліотека `react-router-dom` для маршрутизації. За допомогою `ReactDOM.createRoot`, встановлюється кореневий елемент, у який рендериться весь додаток. Застосовується `<React.StrictMode>`, що допомагає виявляти потенційні проблеми та попереджати про них. Все це відбувається під час виклику функції `root.render`, де монтується головний компонент `App` в межах `Router`.

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.scss';
import App from './App';
import 'macro-css';
import {BrowserRouter as Router} from 'react-router-dom';

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(
  <React.StrictMode>
    <Router>
      <App />
    </Router>
  </React.StrictMode>
);
```

У файлі `App.jsx` ми збираємо усі компоненти веб-додатку. Цей єдиний файл імпортується до `index.js`, використовується та налаштовується бібліотека `React Router`, яка дозволяє зв'язувати сторінки сайту з відповідними адресами. Залежно від `URL`-адреси, на якій знаходиться користувач, відображається

відповідний інтерфейс. Це дозволяє переключати сторінки сайту без перезавантаження, оскільки на сторінці змінюється лише контент.

У файлі `App.jsx` також ідентифікуються та зберігаються основні хуки стану та контексту, а також функції обробки інформації, які передаються звідти до інших сторінок або компонентів через хуки контексту та пропси. При завантаженні усього сайту ми робимо асинхронний запит `Axios` для отримання необхідної інформації з бази даних. Отримані дані записуються у хук `items` для відображення на сторінці. Код цього файлу наведено у додатку А під назвою файлу `App.jsx`.

Файл `Authorisation.jsx` відповідає за відображення вікна авторизації на сторінці. Цей компонент включає форми для входу та реєстрації користувача.

У компоненті використовуються станові змінні, такі як `isPasswordRecoveryModalOpen`, яка відповідає за відкриття та закриття вікна відновлення пароля.

Метод `handleTabClick` викликається при кліку на вкладку «Вхід» або «Реєстрація» і відповідає за зміну видимості відповідного вмісту вкладки.

Компонент повертає розмітку, що містить дві вкладки: «Вхід» та «Реєстрація». Кожна вкладка містить форму, яка дозволяє користувачеві ввести свої дані для авторизації або реєстрації.

Компонент також містить модальне вікно для відновлення пароля. Це вікно з'являється, коли `isPasswordRecoveryModalOpen` має значення `true`. Користувач може ввести свою електронну пошту для відновлення пароля.

Якщо користувач бажає додати товар у корзину або в обране, він спочатку повинен увійти в свій профіль і лише після в нього буде можливість це зробити, а також створити замовлення. Код цього файлу наведено у додатку А під назвою файлу `Authorisation.jsx`.

Файл `Home.jsx` відповідає за відображення головної сторінки, на якій розташована вся необхідна інформація. Цей компонент отримує дані через пропси та `useContext` з `App.jsx`.

На головній сторінці відображаються наступні елементи:

- слайди новин, які передаються компоненту Slider за допомогою пропсу slides.

- категорії товарів, представлені у вигляді карток, що відображаються у компоненті Card_c. Ці категорії передаються як масив arr_Card_c і відображаються на сторінці у вигляді рядків.

- картки товарів, які є популярними. Вони відображаються у компоненті Card_h, що отримує дані з масиву. Кожен елемент масиву відповідає окремому товару з бази даних, який має позначку «хіт».

Схожий принцип відображення елементів використовується і в інших частинах сайту, таких як Content.jsx, Favorite.jsx та Order.jsx. У цих компонентах також відображаються списки елементів, які перебираються за допомогою map і відображаються на сторінці.

2.6. Обґрунтування та організація вхідних та вихідних даних програми

Програма отримує вхідні дані з різних джерел для забезпечення свого функціонування. Ці дані можуть бути завантажені з бази даних, локального сховища або передані через глобальні змінні додатку інтернет-магазину.

Вхідні дані:

- авторизаційні дані користувача та його особиста інформація;
- інформація про замовлення під час його оформлення;
- пошуковий запит через поле пошуку;
- натискання на візуальні елементи управління застосунку.

Вихідні дані:

- інформація про товари;
- інформація про бренд;
- інформація про оплату та доставку;
- контактна інформація;

– веб-сторінки інтернет-магазину, з якими взаємодіє користувач при перегляді сайту.

2.7. Опис розробленого програмного продукту

2.7.1. Використані технічні засоби

Програмний продукт є досить гнучким щодо складу та параметрів технічних засобів, на яких він може працювати. Він може бути запущений на різних типах та конфігураціях пристроїв, таких як ПК, ноутбуки, планшети або смартфони, незалежно від їхньої операційної системи.

Для доступу до програми потрібний лише веб-браузер. Програмний код розроблявся у середовищі Visual Studio Code з використанням бібліотеки React. Це дозволяє розробникам створювати інтерактивні та динамічні веб-додатки, які можуть працювати на різних платформах та пристроях.

Сайт розроблено на персональному комп'ютері:

- Операційна система Windows 10;
- процесор класу AMD Ryzen 5 3600 (3.6 - 4.2 ГГц);
- материнська плата GIGABYTE B450M S2H;
- два модулі пам'яті Kingston Fury DDR4-3200 8192MB;
- Відеоадаптер GeForce RTX 3050;
- SSD диск Samsung 870 Evo-Series 250GB 2.5" SATA III, HDD диск 3.5" SATA 1TB WD Purple (WD10PURZ);
- рідкокристалічний IPS монітор з діагоналлю 27";
- доступ до мережі Internet;
- маніпулятор «миша»;
- маніпулятор «клавіатура».

Зазначені технічні характеристики є рекомендованими, вони вказують на мінімальні вимоги до наявних технічних засобів для оптимальної роботи розробленого програмного продукту. Це означає, що якщо пристрої мають не

нижчі параметри, визначені у специфікації, то програма буде працювати з відповідною надійністю, швидкістю обробки даних і забезпеченням безпеки, що були вимогами замовника.

2.7.2. Використані програмні засоби

Під час розробки даного застосунку були використані такі програмні засоби:

- Visual Studio Code (VS Code);
- Браузер Google Chrome;
- Операційна система: Windows 10.

Visual Studio Code - це один із найпопулярніших редакторів коду, розроблений корпорацією Microsoft. Він розповсюджується у безкоштовному доступі та підтримується всіма актуальними операційними системами: Windows, Linux та macOS. VS Code є звичайним текстовим редактором з можливістю підключення різних плагінів, що дає можливість працювати з різними мовами програмування для розробки будь-якого ІТ-продукту. Редактор коду зображено на рис. 2.9 .

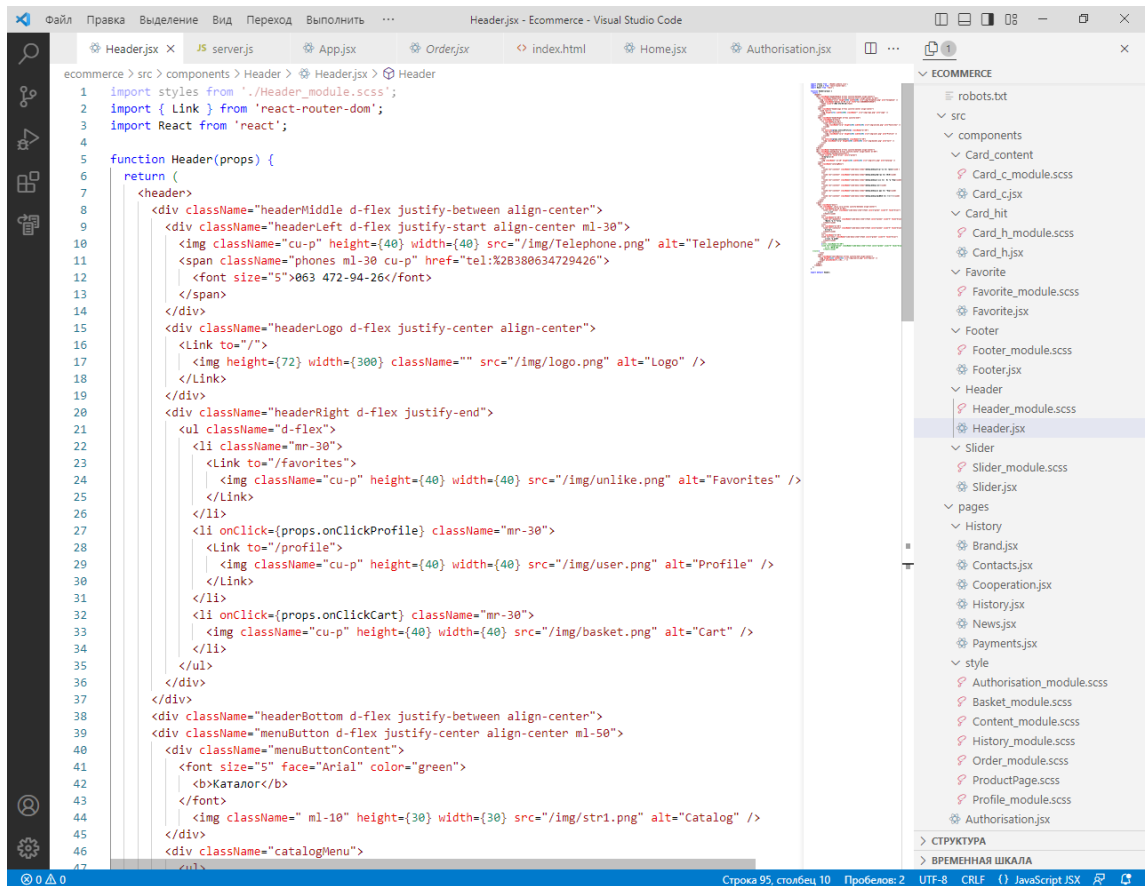


Рис. 2.9 Редактор коду Visual Studio Code

Особливості Visual Studio Code [21]:

- використання різних мов програмування, включаючи C, C#, C++, Fortran, Go, Java, JavaScript, Node.js, Python, Rust;
- базується на структурі Electron і використовує механізм компонування Blink;
- використовує компонент редактора «Моноасо», який також використовується в Azure DevOps;
- забезпечує базову підтримку більшості мов програмування з можливістю розширення за допомогою розширень з VS Code Marketplace;
- дозволяє відкривати каталоги замість проектів, що дає можливість працювати як мовно-агностичний редактор коду для будь-якої мови;
- має налагодження, підсвічування синтаксису, згортання коду, налаштовані фрагменти та IntelliSense для різних мов;

- можливість розширення редактора за допомогою створення розширень для підтримки нових мов, тем, налагоджувачів, лінтерів коду та іншого;
- вбудована підтримка керування версіями, що дозволяє працювати з репозиторіями Git, Apache Subversion та інших;
- розширення для FTP, які дозволяють використовувати програму для веб-розробки з безпосереднім доступом до сервера;
- налаштування кодової сторінки, символу нового рядка та мови програмування для активного документа;
- збір телеметрії та можливість вимкнути його надсилання до Microsoft, з відкритим вихідним кодом програми.

2.7.3. Виклик та завантаження програми

Для завантаження веб-додатку необхідно зробити наступні кроки:

- завантажити файл веб-додатку з фізичного носія;
- встановити NodeJS, бажано версію не менше v.18.16.1;
- відкрити папку з кодом додатку в програмі VS Code;
- в терміналі додати всі пакети для роботи React, ввівши команду `npm install`;
- перейти в терміналі до папки `backend` та ввести команду `node server.js` для запуску серверної частини;
- перейти в терміналі до папки `frontend` та виконати команду `npm start`, після чого запуситься локальний сервер та відкриється сторінка в браузері.

Програмний продукт не вимогливий до виклику та завантаження і може працювати на ПК, ноутбуках, планшетах чи смартфонах різних типів та конфігурацій під управлінням різних ОС.

2.7.4. Опис інтерфейсу користувача

Після запуску веб-додатку, користувач потрапляє на головну сторінку (рис. 2.10), яка є однією з основних сторінок проекту, на ній користувач може взаємодіяти з елементами меню, які сприяють зручній навігації користувача по сайту. Наприклад, навігаційне меню містить посилання на різні частини сайту, дозволяючи користувачеві швидко переходити до бажаної сторінки або функції.

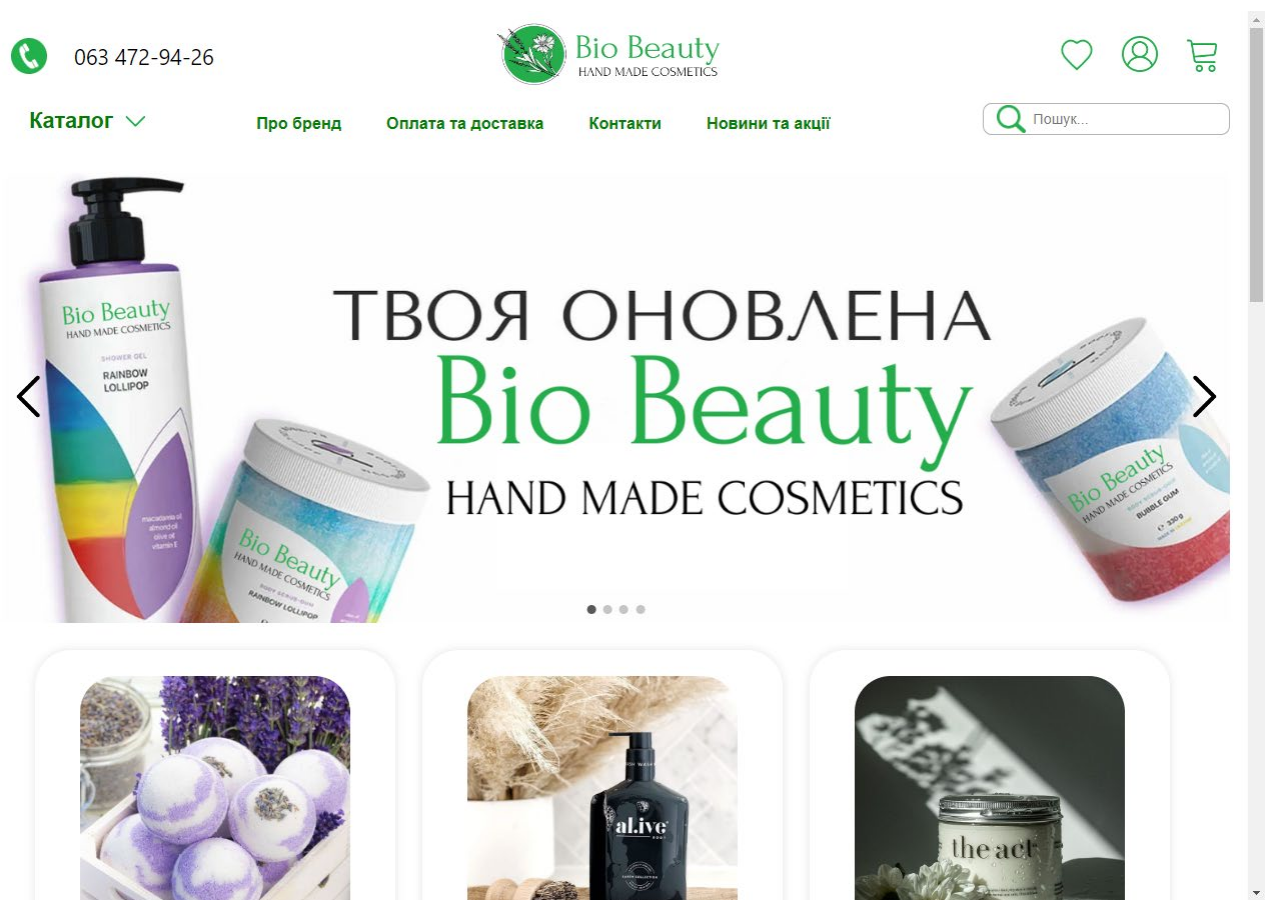


Рис. 2.10 Головна сторінка проекту

На головній сторінці користувачеві надається інформація про актуальні новини у вигляді слайдів (рис. 2.10), швидкий перехід за необхідною йому категорією (рис. 2.11).

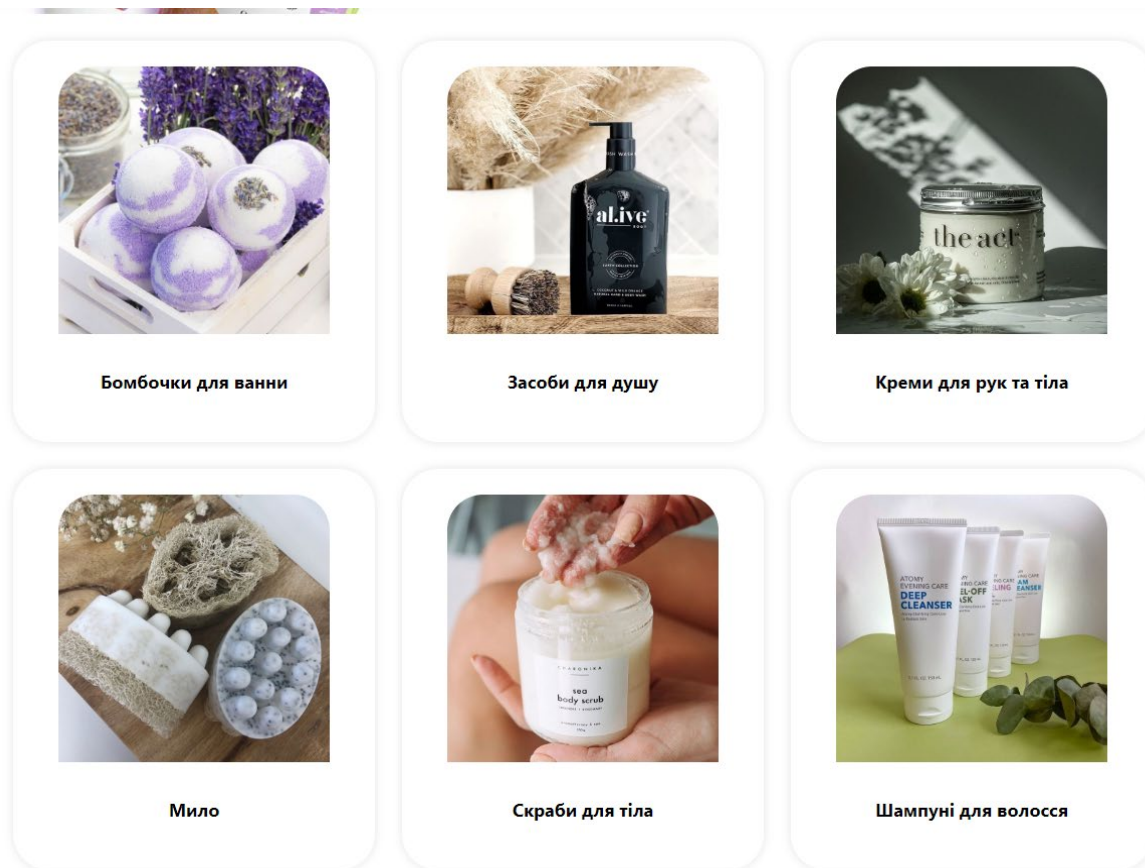


Рис. 2.11 Категорії на головній сторінці

Після категорій йдуть товари які користуються популярністю, тобто користувачеві буде виведено список товарів які є хітами продаж (рис. 2.12).

Хіти продаж

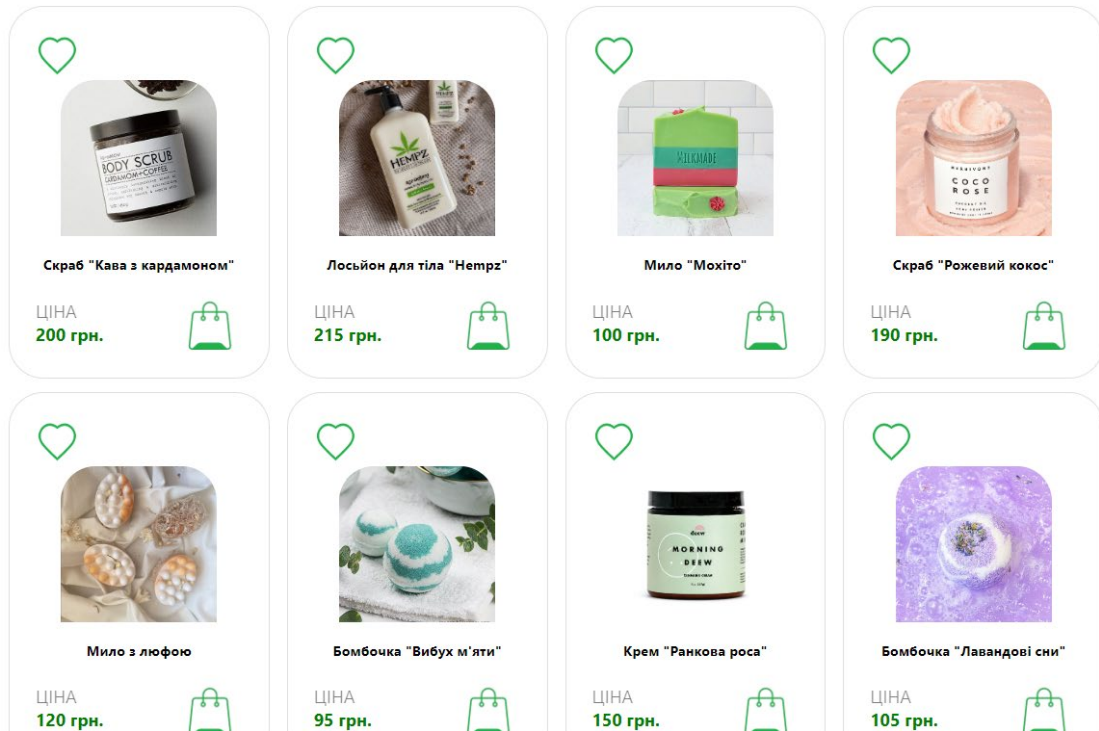


Рис. 2.12 Хіти продаж на головній сторінці

Для переходу до списку товарів, користувачеві необхідно обрати категорію в шапці профілю (рис. 2.13), або одразу під слайдами (рис. 2.11).

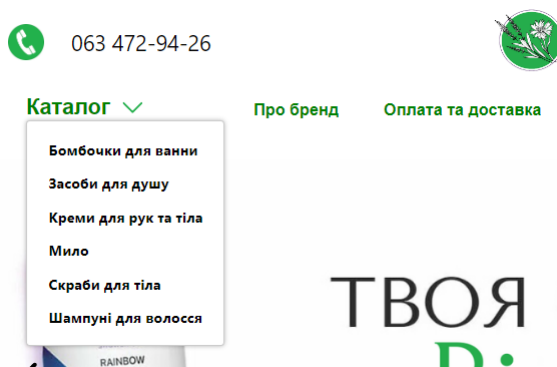


Рис. 2.13 Категорії товарів

Після переходу користувачеві буде виведено список товарів за обраною ним категорією (рис. 2.14).

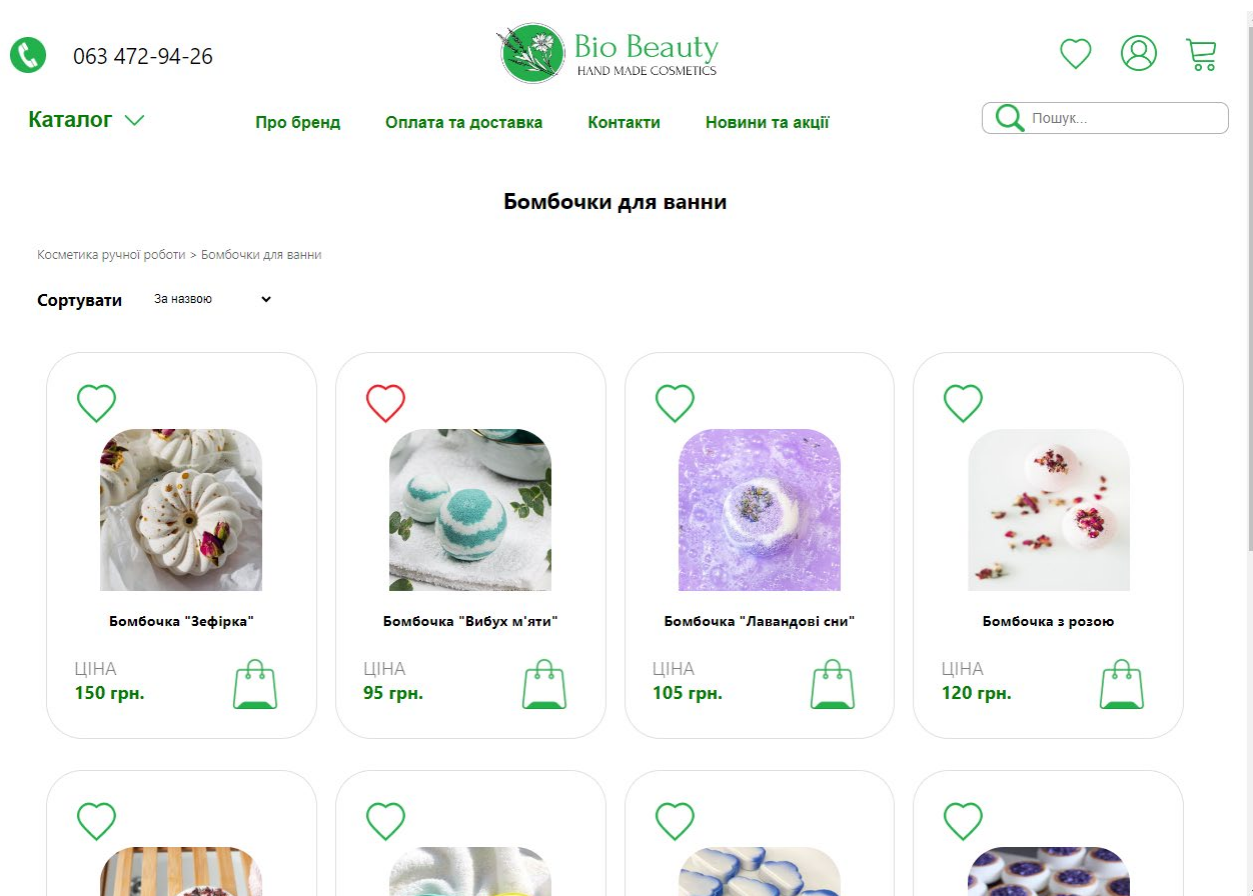
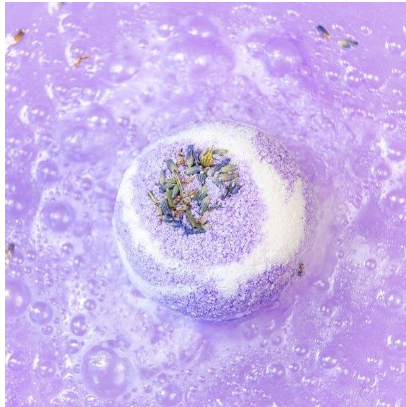


Рис. 2.14 Список товарів категорії «Бомбочки для ванни»

Щоб подивитися детальну інформацію про товар, користувач повинен натиснути на зображення чи назву товару, після чого його буде перенесено на нову сторінку де він зможе ознайомитись з повною інформацією обраного товару, а саме: актуальною ціною, опис та склад товару, а також інформацією про доставку (рис. 2.15).

Косметика ручної роботи > Бомбочки для ванни > Бомбочка "Лавандові сні"



Бомбочка "Лавандові сні"

Ціна: 105 грн.



Опис

Лавандові бомбочки для ванни - це чудовий спосіб розслабитися та пом'якшити шкіру під час ванни. Лавандові бомбочки для ванни можуть створити чудову атмосферу релаксу та сприяти вашому загальному благополуччю. Аромат лаванди має властивість заспокоювати нервову систему і знижувати рівень стресу. Під час ванни з лавандовою бомбочкою ви можете відчувати пом'якшення напруження та розслаблення. Бомбочки для ванни часто містять зволожуючі та живильні інгредієнти, такі як кокосова олія. Вони допомагають зробити вашу шкіру м'якою, гладкою і зволоженою після ванни. Лавандові бомбочки для ванни - це чудовий спосіб відпочити, розслабитися та насолодитися благоуханням лаванди.

Склад

На 100% екологічний.

Доставка та оплата

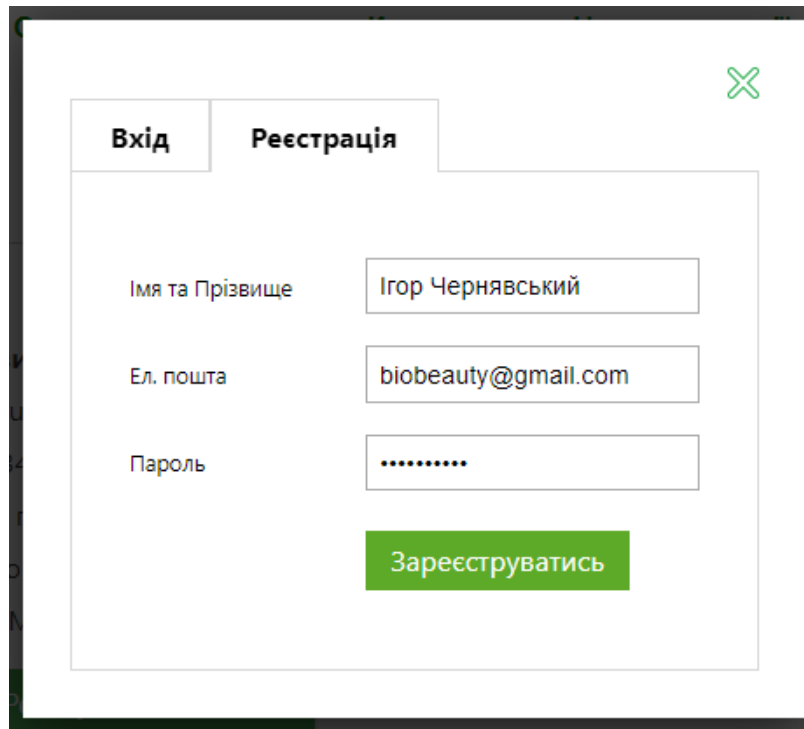
«Новою поштою» по Україні — за тарифами перевізника

«Укрпоштою» по Україні — за тарифами перевізника

[Більше інформації про доставку](#)

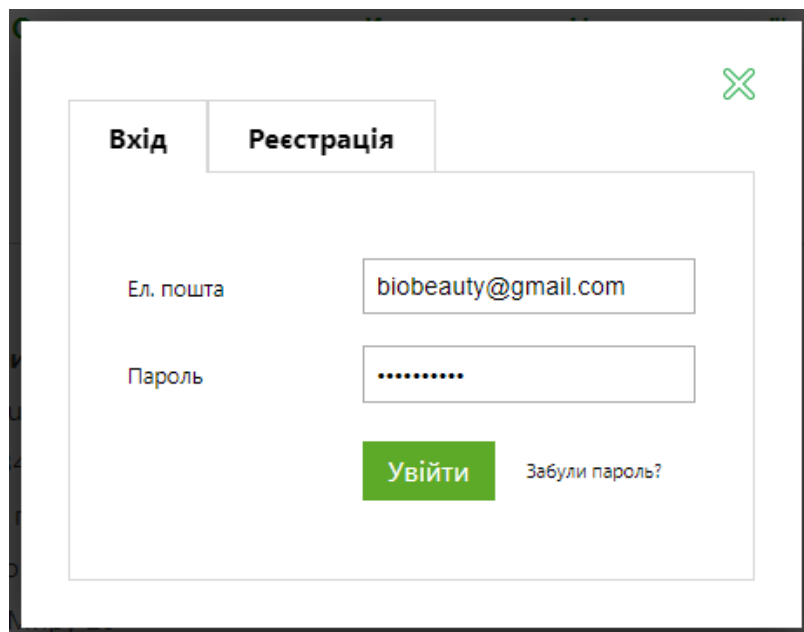
Рис. 2.15 Сторінка обраного товару

Для того щоб користувач міг взаємодіяти за сайтом з усіма можливостями, йому необхідно створити (рис.2.16) та увійти в обліковий запис (рис.2.17). Для цього користувач повинен натиснути на значок профілю та на відкритій сторінці ввести реєстраційні дані. У полях введення присутня валідація, яка перевіряє правильність введених даних. Одразу після коректного введення та успішної реєстрації, увійти в обліковий запис для подальшої взаємодії з інтернет-магазином.



The image shows a registration form with two tabs: "Вхід" (Login) and "Реєстрація" (Registration). The "Реєстрація" tab is active. The form contains three input fields: "Імя та Прізвище" (Name and Surname) with the value "Ігор Чернявський", "Ел. пошта" (Email) with the value "biobeauty@gmail.com", and "Пароль" (Password) with masked characters ".....". A green button labeled "Зареєструватись" (Register) is positioned below the password field. A green 'X' icon is in the top right corner.

Рис. 2.16 Форма реєстрації



The image shows a login form with two tabs: "Вхід" (Login) and "Реєстрація" (Registration). The "Вхід" tab is active. The form contains two input fields: "Ел. пошта" (Email) with the value "biobeauty@gmail.com" and "Пароль" (Password) with masked characters ".....". A green button labeled "Увійти" (Login) is positioned below the password field, followed by a link "Забули пароль?" (Forgot password?). A green 'X' icon is in the top right corner.

Рис. 2.17 Форма входу

Кожен товар який сподобався користувач може додати до «обраного» натиснувши на значок серця який присутній на кожній картці товару, щоб подалі він міг швидко знайти товар на сторінці «Мої бажання» (рис. 2.18).

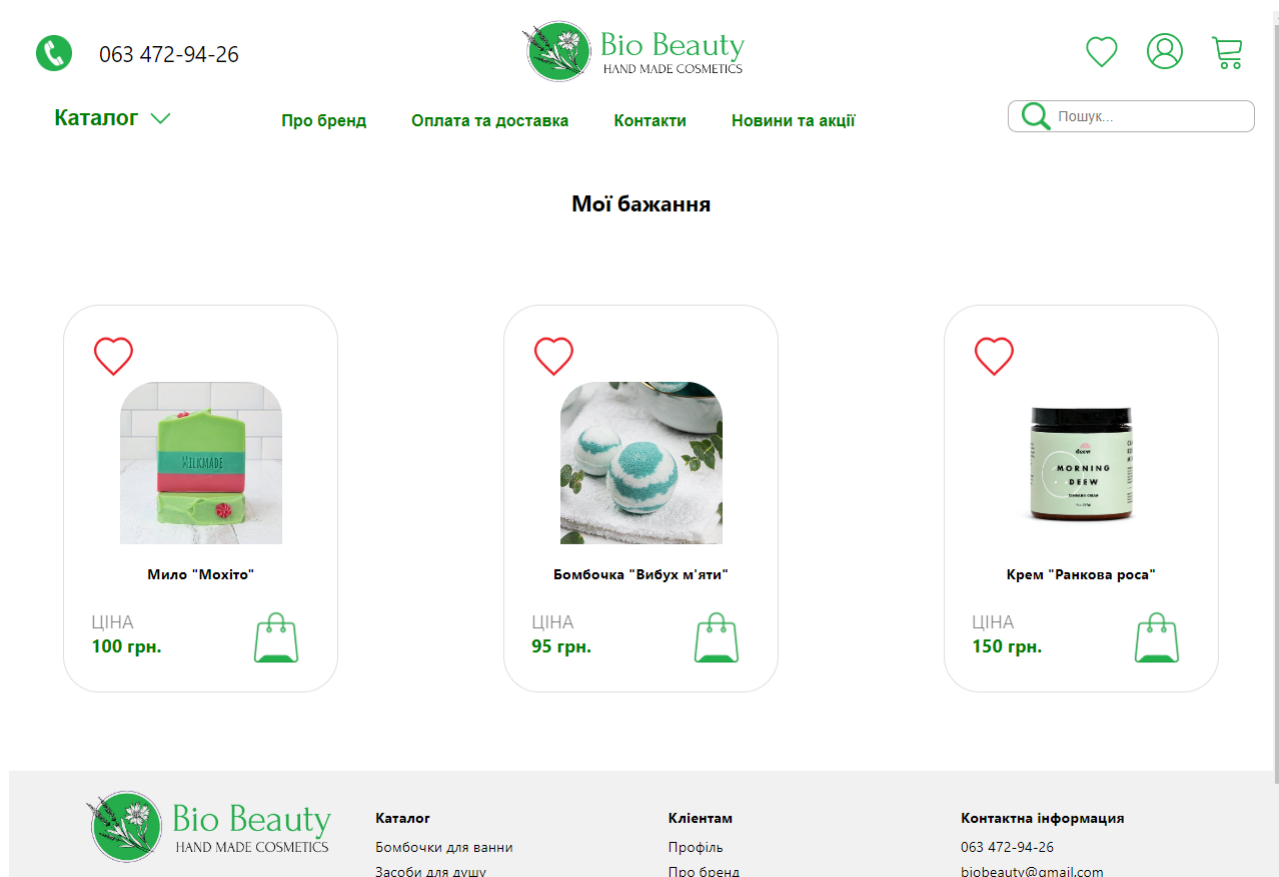


Рис. 2.18 Сторінка «Мої бажання»

А у разі якщо користувач хоче зробити замовлення, то йому необхідно натиснути на кнопку купити, після чого товар автоматично буде додано до кошика (рис. 2.19). Якщо кошик порожній то при натисканні на нього, користувачу буде виведено кошик з текстом що він порожній (рис. 2.20).

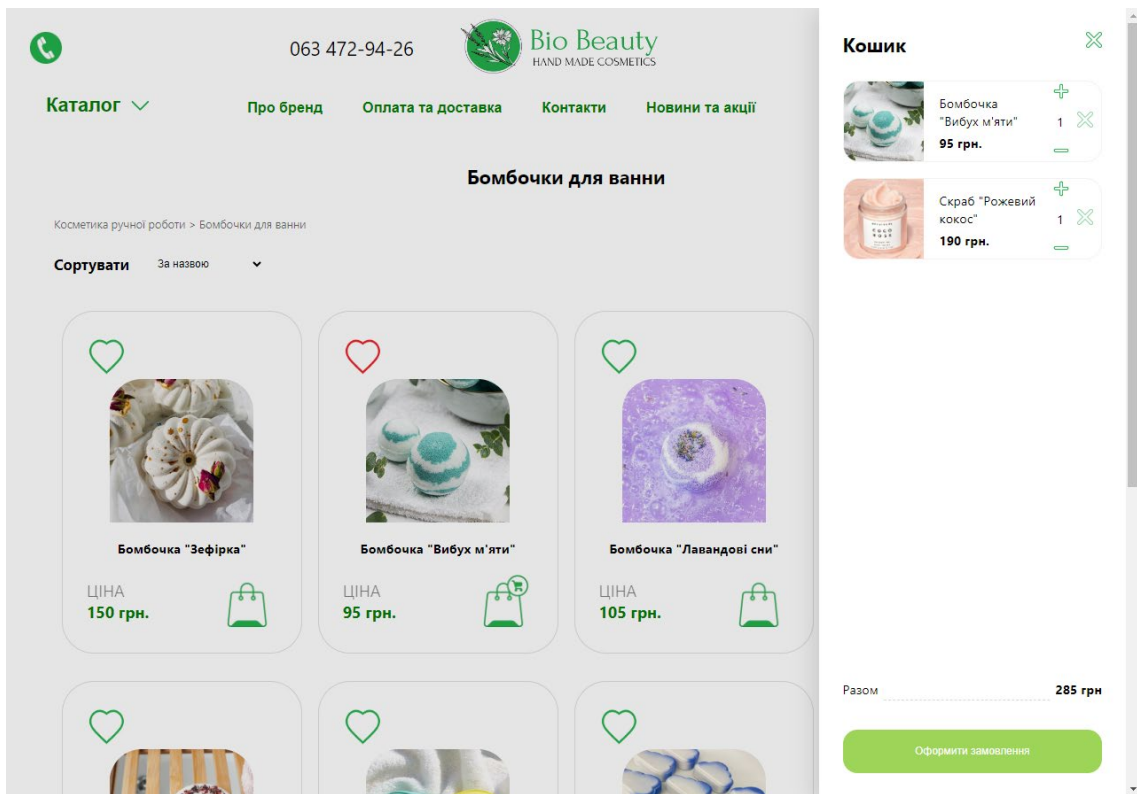


Рис 2.19 Наповнений кошик товарами

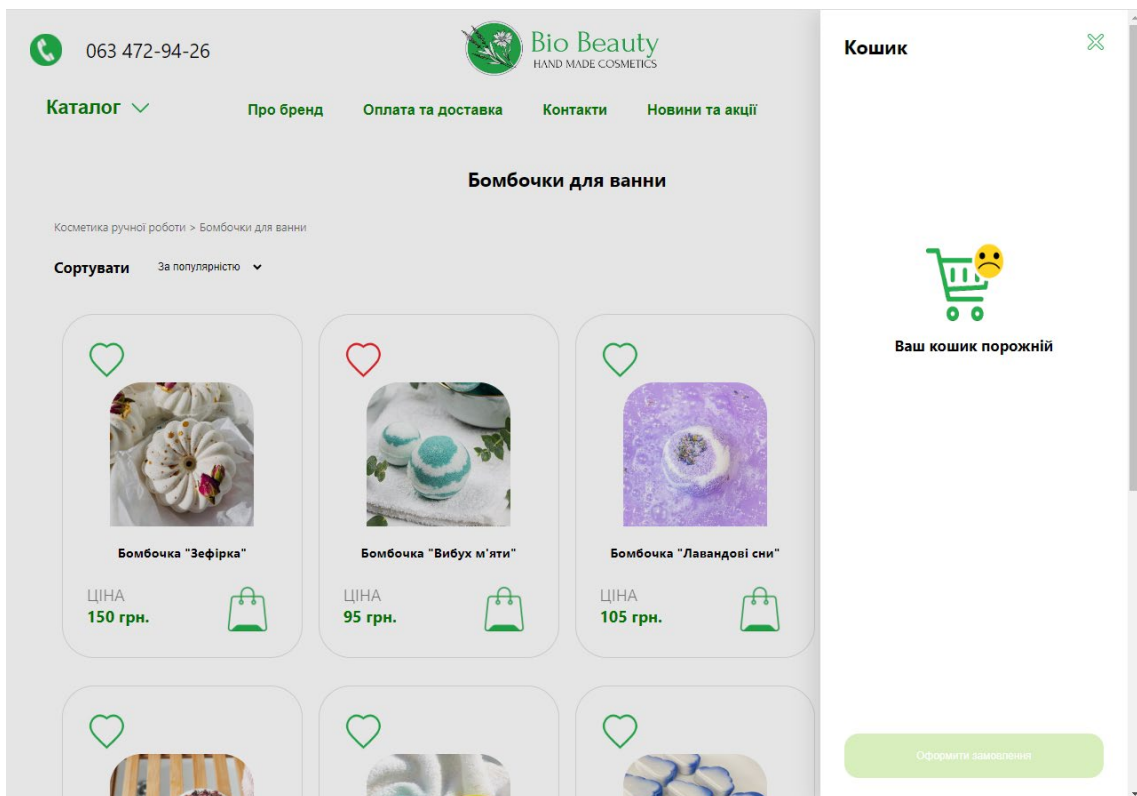


Рис. 2.20 Порожній кошик

На кожній сторінці у шапці профіля є рядок пошуку, за допомогою якого можна швидко знайти необхідний товар, або схожі за назвою товари, яких може бути декілька (рис. 2.21-2.22).

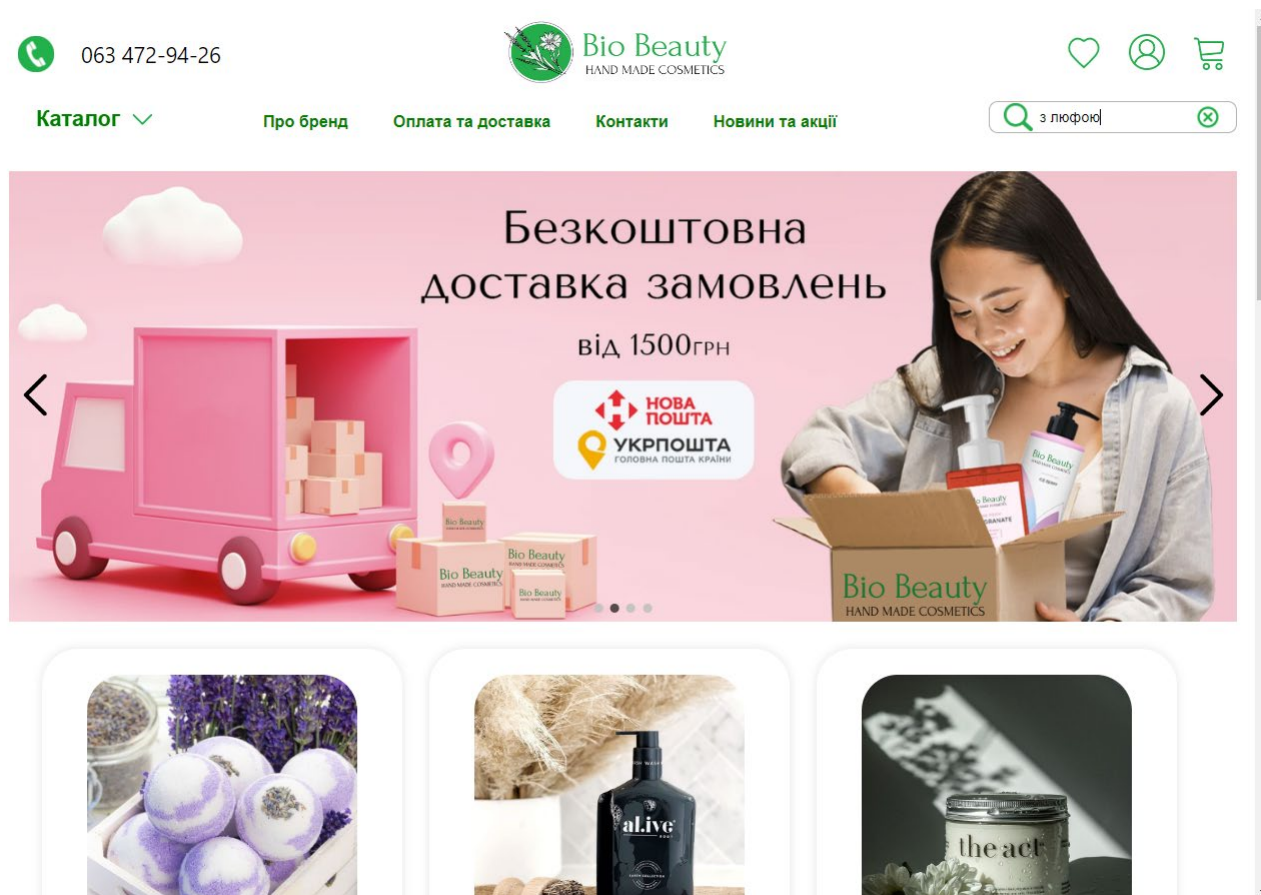


Рис. 2.21 Введення пошукового запиту

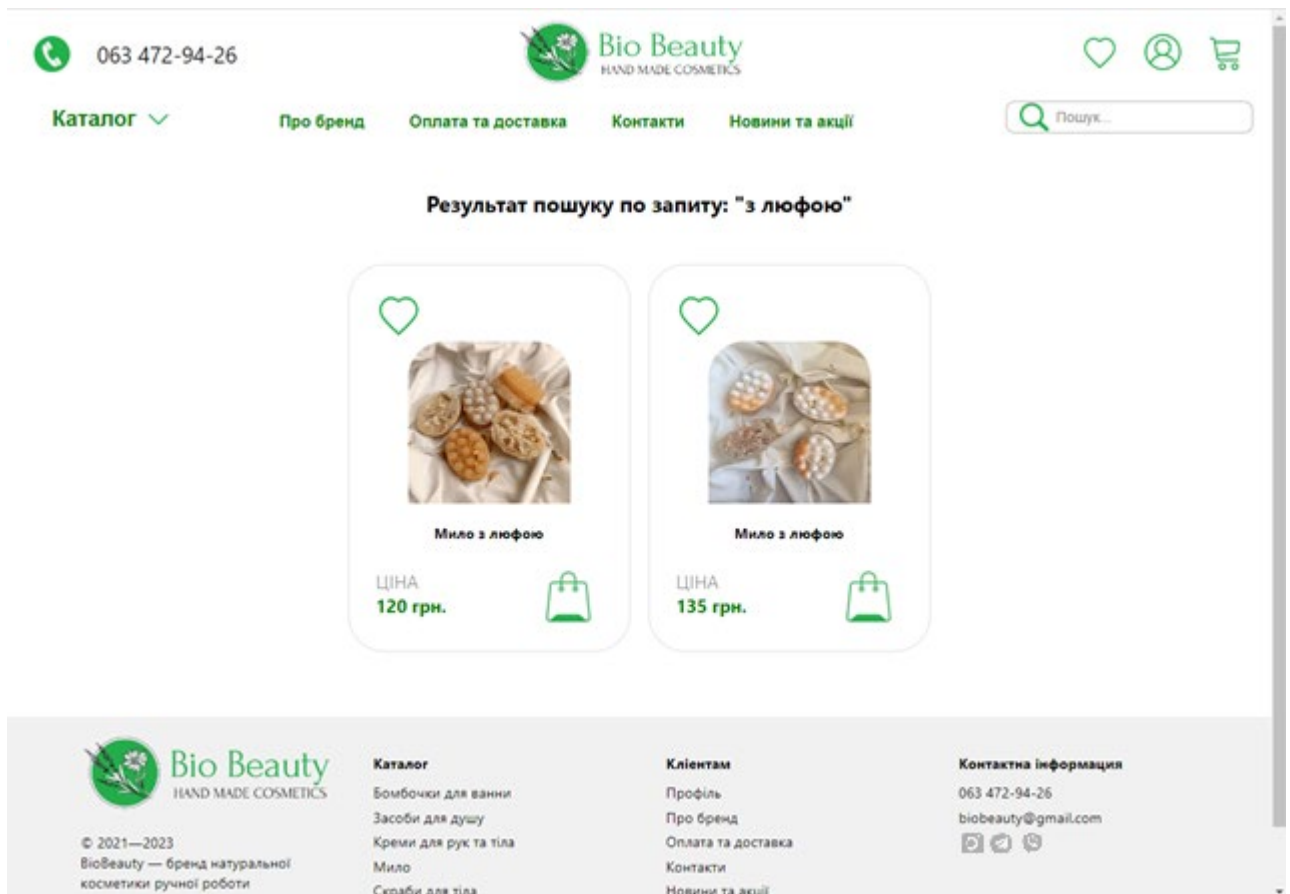


Рис. 2.22 Результат за пошуковим запитом

Щоб переглянути свою особисту інформацію, а також свої замовлення чи бажання, користувачеві потрібно натиснути на кнопку профіля де йому буде показано вкладку «Особисті дані» (рис. 2.23) і за потребою їх можна відредагувати натиснувши на відповідну клавішу (рис. 2.24).

Профіль

Особисті дані

Мої замовлення

Мої бажання

[Вийти з облікового запису](#)

Ім'я та Прізвище: Ігор Чернявський
Email: biobeauty@gmail.com
Телефон: 0634729426
Пошта: Нова пошта
Місто: Дніпро
Адреса: вул. Миру 20

Редагувати



© 2021—2023
BioBeauty — бренд натуральної
косметики ручної роботи

Каталог

Бомбочки для ванни
Засоби для душу
Креми для рук та тіла
Мило
Скраби для тіла
Шампуні для волосся

Клієнтам

Профіль
Про бренд
Оплата та доставка
Контакти
Новини та акції

Контактна інформація

063 472-94-26
biobeauty@gmail.com
📞 📧 📍

Рис. 2.23 Сторінка профілю

Профіль

Особисті дані

Мої замовлення

Мої бажання

[Вийти з облікового запису](#)

Ім'я та Прізвище:

Ігор Чернявський

Email:

biobeauty@gmail.com

Телефон:

0634729426

Пошта:

Нова пошта

Місто:

Дніпро

Адреса:

вул. Миру 20

Зберегти

Рис. 2.24 Редагування особистої інформації

У вкладці «Мої замовлення» буде виведено всі замовлення які робив користувач, якщо їх немає, то буде виведено повідомлення про їх відсутність (рис. 2.25).

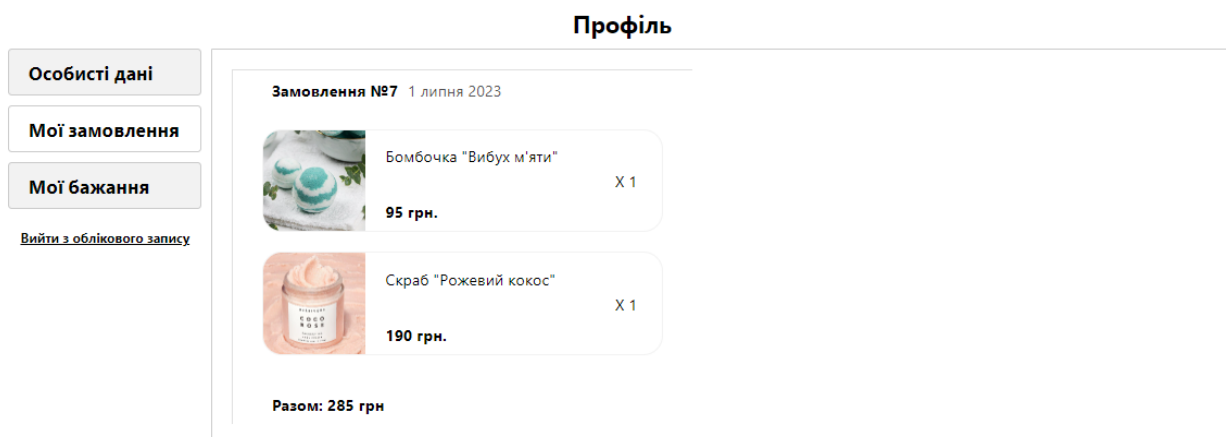


Рис. 2.25 Вкладка профіля «Мої замовлення»

У вкладці «Мої бажання» буде виведено всі товари які користувач зберіг до обраного, аналогічно сторінці «Мої бажання» (рис. 2.26).

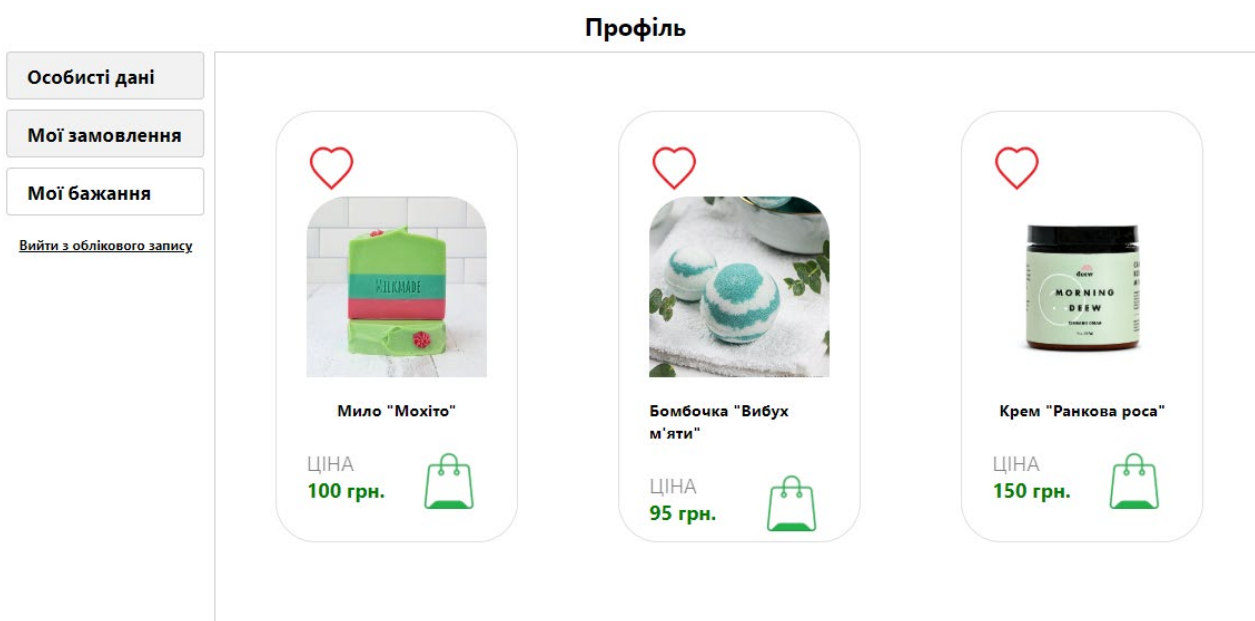


Рис. 2.26 Вкладка профіля «Мої бажання»

Коли користувач сформував кошик та хоче зробити замовлення, то йому необхідно натиснути кнопку «Оформити замовлення» яка знаходиться у кошику (рис. 2.19).

Після чого користувача буде перенесено на нову сторінку, де потрібно вказати актуальну інформацію про себе, тип оплати, службу доставки та адресу за якою буде відправлено замовлення і натиснути на кнопку підтвердження замовлення (рис. 2.27).

Замовлення №7

The screenshot displays a checkout page with two main sections. On the left, under the heading 'Особиста інформація', there are input fields for 'Ім'я та Прізвище' (Igor Чернявський), 'Email' (biobeauty@gmail.com), 'Телефон' (0634729426), 'Пошта' (Нова пошта), 'Місто' (Дніпро), 'Адреса' (вул. Миру 20), and 'Тип оплати' (Передплата). On the right, under 'Ви обрали:', two items are listed: 'Бомбочка "Вибух м'яти"' (95 грн) and 'Скраб "Рожевий кокос"' (190 грн). Below the items, the total 'До сплати: 285 грн' is shown, followed by a green button labeled 'Оформити замовлення'.

Рис. 2.27 Сторінка оформлення замовлення

Для отримання інформації про бренд косметики, оплати чи доставку, контактну інформацію, або новини та акції, користувачеві потрібно натиснути на відповідне посилання у шапці профілю, після чого його буде перенесено до необхідної сторінки з інформацією, яка його цікавить. Сторінки зображено на рис. 2.28 – 2.31 .

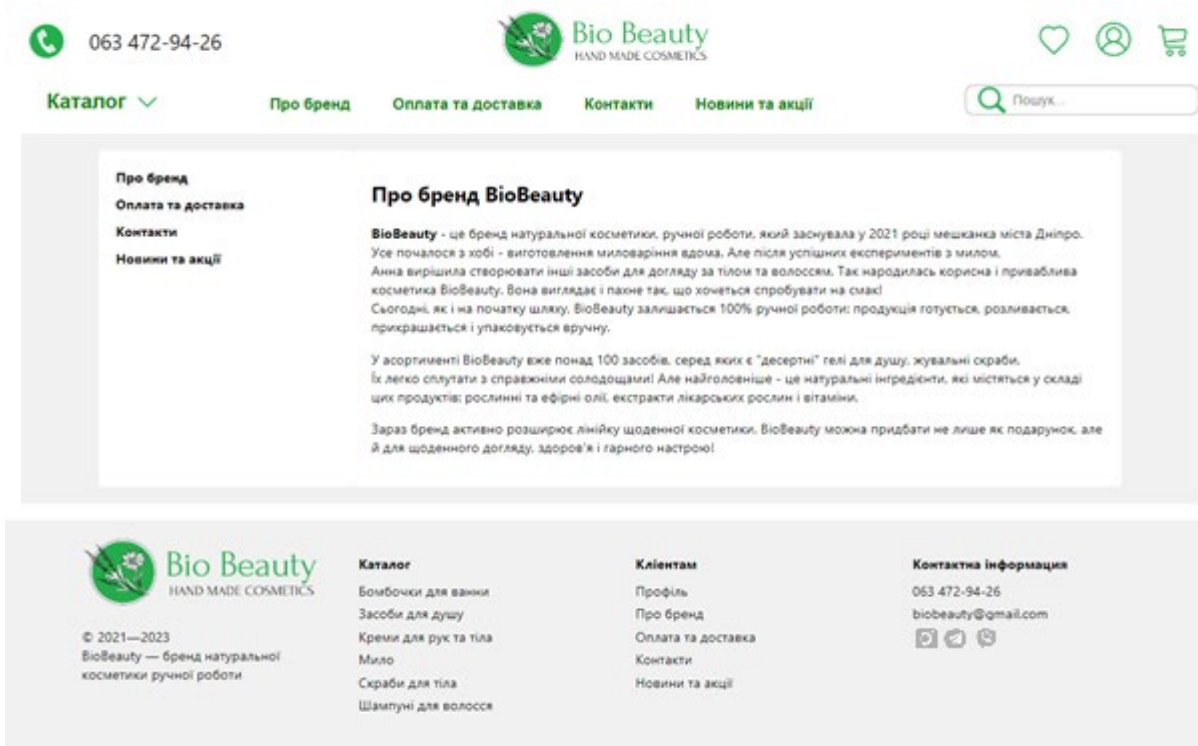


Рис. 2.28 Сторінка «Про бренд»

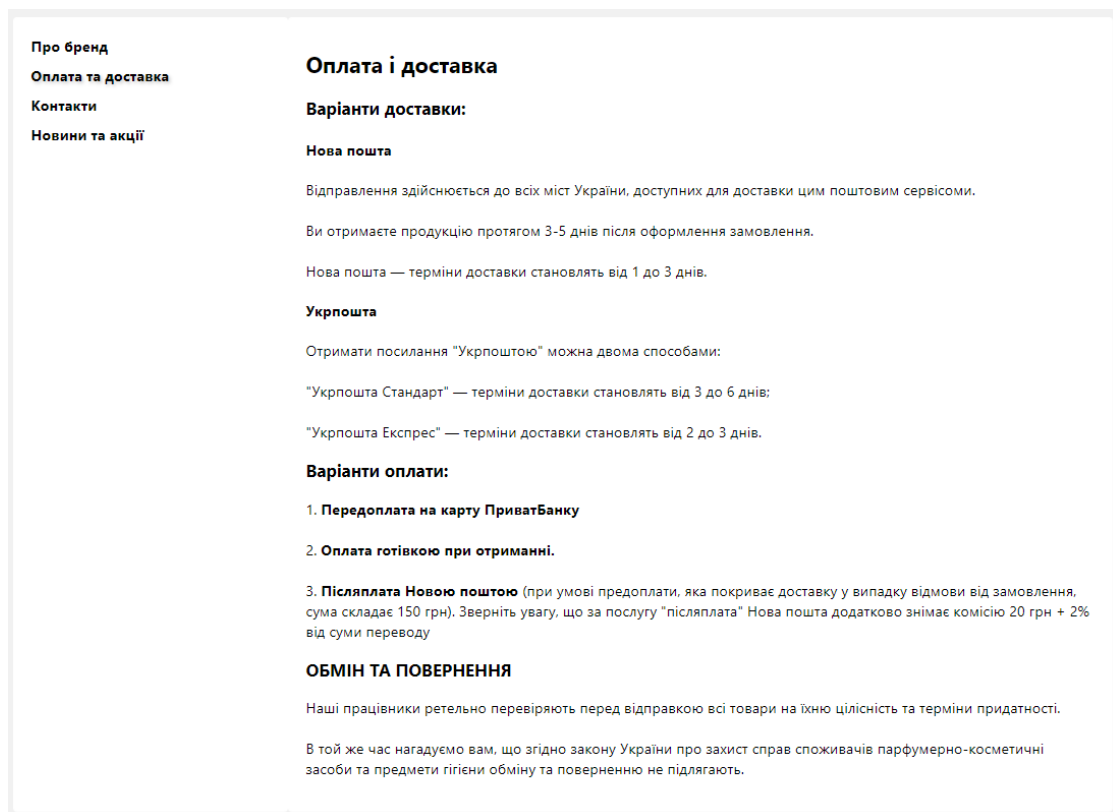


Рис. 2.29 Сторінка «Оплата та доставка»

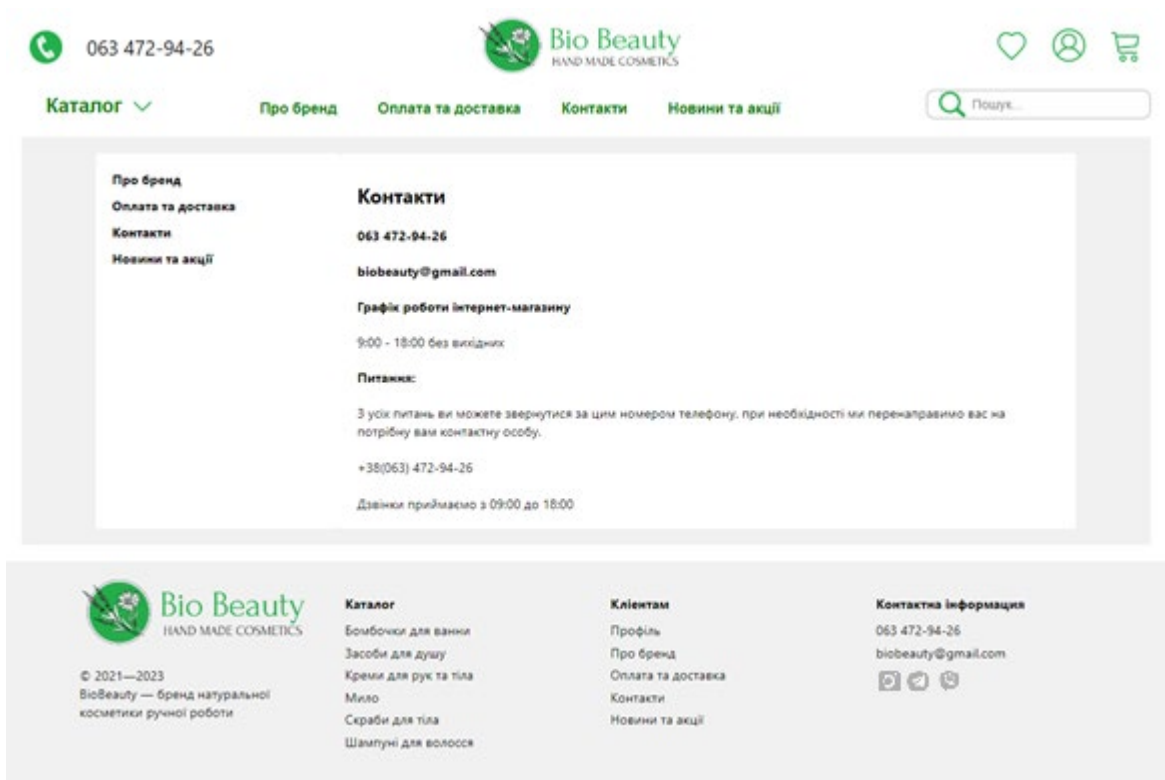


Рис. 2.30 Сторінка «Контактна інформація»

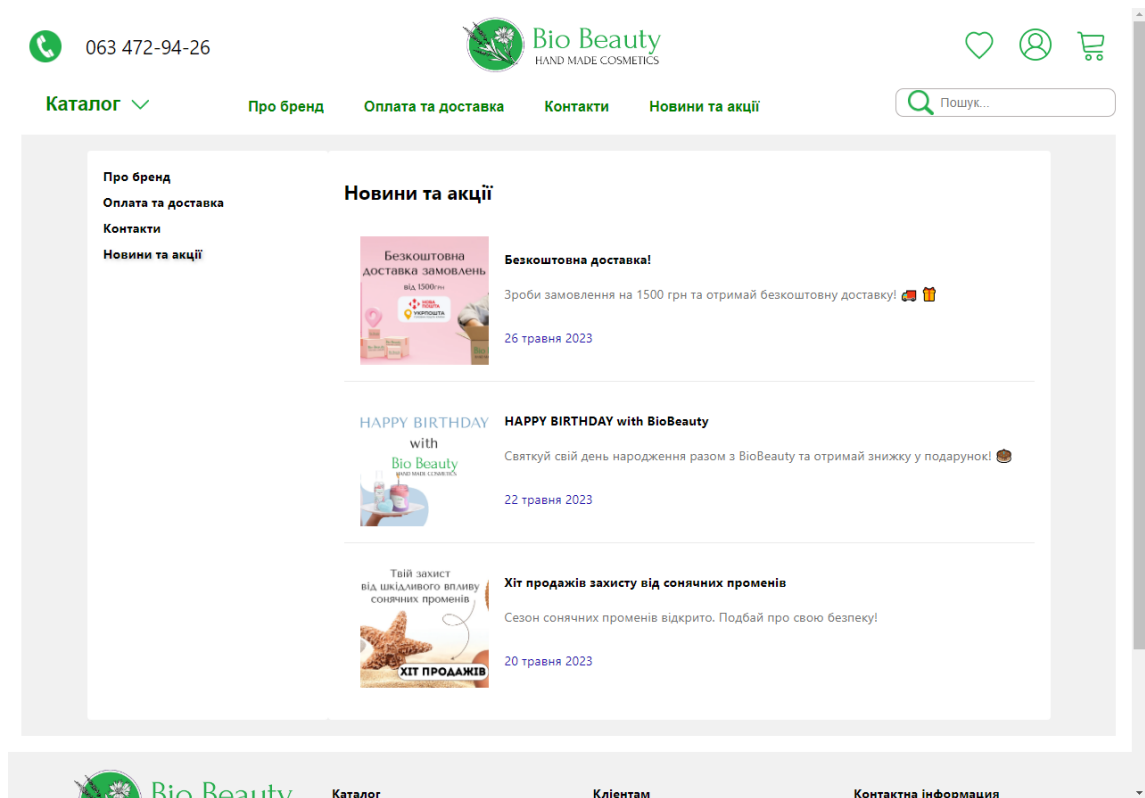


Рис. 2.31 Сторінка «Новини та акції»

Внизу кожної сторінки є підвал, де користувачеві буде продубльовано посилання, які є у шапці профіля і контактну інформацію для зручності навігації сайту, коли користувач знаходиться внизу сторінки (рис. 2.32).

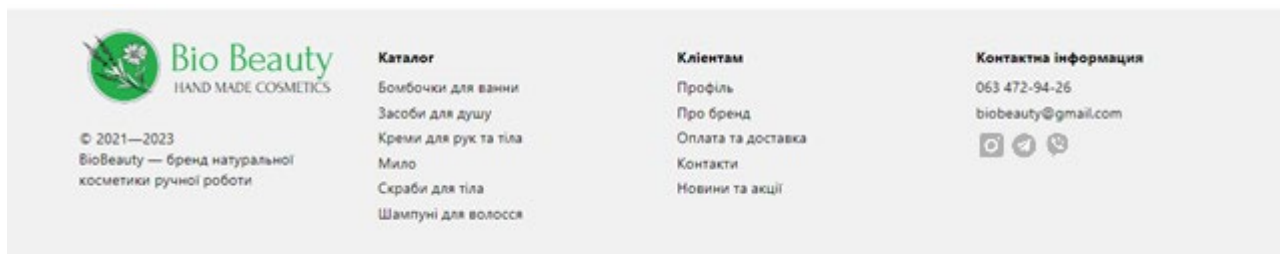


Рис. 2.32 «Підвал» сторінки веб-додатку

Якщо користувач потребує допомоги, він може зателефонувати за номером який вказаний як у шапці профілю, так і у підвалі.

Для виходу з облікового запису необхідно у кабінеті натиснути на відповідну кнопку у профілі яка знаходиться зліва під вкладками (рис. 2.23).

РОЗДІЛ 3

ЕКОНОМІЧНА ЧАСТИНА

3.1. Визначення трудомісткості та вартості розробки програмного продукту

Початкові дані:

1. Передбачуване число операторів програми - 1000.
2. Коефіцієнт складності програми - 1,7.
3. Коефіцієнт корекції програми в ході її розробки - 0,3.
4. Годинна заробітна плата програміста - 220 грн/год.

За даними DOU, в середньому, Junior Full Stack Developer в Україні отримує зарплату близько 1000 доларів на місяць, що еквівалентно приблизно 37000 гривень(за курсом 1\$=37грн) [20]. Враховуючи 8-годинний робочий день та приблизно 21 робочий день на місяць, погодинна ставка для таких розробників становить приблизно 220 гривень.

5. Коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі - 1,3.

6. Коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності - 1,2.

7. Вартість машино-години ЕОМ - 2,14 грн/год.

ПК споживає 250 ватт на годину, а монітор - 40 ватт на годину. Вартість електроенергії за даними постачальника становить 2,64 грн/кВт на годину то за підрахунками вартість електроенергії становить $(250 + 40) * 2,64 / 1000 = 0,77$ грн/год. Витрати на домашній інтернет від HomeNet складають 230 грн/міс. Таким чином, загальна вартість роботи ПК виходить $0,77 * 8 * 21 + 230 = 359,36$ грн/міс. або 2,14 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:

$$t = t_o + t_u + t_a + t_n + t_{omл} + t_{\partial}, \text{ людино-годин,} \quad (3.1)$$

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;

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

t_{∂} - витрати праці на підготовку документації.

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

Умовне число операторів:

$$Q = q \cdot C \cdot (1 + p) \quad (3.2)$$

де q - передбачуване число операторів (1000);

C - коефіцієнт складності програми (1,7);

p - коефіцієнт корекції програми в ході її розробки (0,3).

Отже, умовне число операторів в програмі має значення:

$$Q = 1000 \cdot 1,7 \cdot (1 + 0,3) = 2210$$

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. При стажі роботи близько 3 років він складає 1,3.

Будемо вважати збільшення витрат праці внаслідок недостатнього опису завдання не більше 50% ($B = 1,2$). З урахуванням коефіцієнта кваліфікації $k = 1,3$, отримуємо витрати праці на вивчення опису завдання:

$$t_u = (2210 \cdot 1,2) / (85 \cdot 1,3) = 24 \text{ людино-годин.}$$

Витрати праці на розробку алгоритму рішення задачі визначаються за формулою:

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

де Q – умовне число операторів програми; (2210)

k – коефіцієнт кваліфікації програміста. (1,3)

Підставивши відповідні значення в формулу (3.4), отримуємо:

$$t_a = 2210 / (23 \cdot 1,3) = 73,91 \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

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

Підставимо значення у формулу (3.5) та отримаємо витрати на складання програми по готовій блок-схемі:

$$t_n = 2210 / (25 \cdot 1,3) = 68 \text{ людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

– за умови автономного налагодження одного завдання:

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

$$t_{oml} = 2210 / (4 \cdot 1,3) = 425 \text{ людино-годин.}$$

– за умови комплексного налагодження завдання:

$$t_{oml}^k = 1,5 \cdot t_{oml}, \text{ людино-годин} \quad (3.7)$$

$$t_{oml}^k = 1,5 \cdot 425 = 637,5 \text{ людино-годин.}$$

Витрати праці на підготовку документації визначаються за формулою:

$$t_d = t_{dp} + t_{do}, \text{ людино-годин} \quad (3.8)$$

де t_{dp} - трудомісткість підготовки матеріалів і рукопису:

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

t_{do} - трудомісткість редагування, печатки й оформлення документації:

$$t_{\partial o} = 0,75 \cdot t_{\partial p}, \text{ людино-годин.} \quad (3.10)$$

Підставляючи відповідні значення, отримаємо:

$$t_{\partial p} = 2210 / (20 \cdot 1,3) = 85 \text{ людино-годин.}$$

$$t_{\partial o} = 0,75 \cdot 85 = 63,75 \text{ людино-годин.}$$

$$t_{\partial} = 85 + 63,75 = 148,75 \text{ людино-годин.}$$

Повертаючись до формули (3.1), отримаємо повну оцінку трудомісткості розробки програмного забезпечення:

$$t = 50 + 24 + 73,91 + 68 + 425 + 148,75 = 789,66 \text{ людино-години.}$$

3.2. Розрахунок витрат на створення програми

Витрати на створення ПЗ $K_{ПО}$ включають витрати на заробітну плату виконавця програми $Z_{ЗП}$ і витрат машинного часу, необхідного на налагодження програми на ЕОМ:

$$K_{ПО} = Z_{ЗП} + Z_{МВ}, \text{ грн} \quad (3.11)$$

Заробітна плата виконавців визначається за формулою:

$$Z_{ЗП} = t \cdot C_{ПР}, \text{ грн} \quad (3.12)$$

де: t - загальна трудомісткість, людино-годин;

$C_{ПР}$ - середня годинна заробітна плата програміста, грн/година.

З урахуванням того, що середня годинна зарплата програміста становить 176 грн / год, отримуємо:

$$Z_{зп} = 789,66 \cdot 176 = 138980,16 \text{ грн.}$$

Вартість машинного часу, необхідного для налагодження програми на ЕОМ, визначається за формулою:

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

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

$C_{мч}$ - вартість машино-години ЕОМ, грн/год (1,18 грн/год).

Підставивши в формулу (3.3) відповідні значення, визначимо вартість необхідного для налагодження машинного часу:

$$Z_{мв} = 425 \cdot 1,18 = 501,5 \text{ грн.}$$

Звідси витрати на створення програмного продукту:

$$K_{по} = 138980,16 + 501,5 = 139481,66 \text{ грн.}$$

Очікуваний період створення ПЗ:

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

де B_k - число виконавців (дорівнює 1);

F_p - місячний фонд робочого часу (при 40 годинному робочому тижні $F_p=176$ годин).

Очікуваний період створення ПЗ:

$$T = 789,66 / 1 \cdot 176 \approx 4,49 \text{ міс.}$$

Висновок: Розроблений продукт є повністю функціонуючим веб-застосунком, який дозволяє зручно та швидко покупцям вибирати та купувати продукт, а продавцеві - додавати нові види продукції або редагувати вже існуючі. Вартість даного програмного забезпечення, згідно розрахункам, становить 139481,66 грн. Також, згідно розрахункам, термін створення цього продукту становить маже 5 місяців. Цей термін обумовлений складністю поставлених задач та великою кількістю правок від замовника, щоб отримати фінальний результат.

ВИСНОВКИ

В результаті проведення кваліфікаційної роботи було успішно розроблено інтернет-магазин, спеціалізований на продажу косметики ручної роботи, з використанням потужної бібліотеки React. Основне призначення цієї системи полягає в тому, щоб забезпечити відвідувачам сайту простий і комфортний доступ до широкого асортименту товарів, пропонуваних у каталозі. Це досягається завдяки оптимальному візуальному представленню вмісту каталогу, що робить процес покупок швидким, зручним та ефективним.

Під час розробки проекту було виконано ряд важливих завдань:

- була проведена детальна аналітична робота та дослідження відповідної галузі, що дозволило зрозуміти специфіку розв'язуваної задачі;
- була створена та налаштована база даних, в якій зберігаються інформація про товари та замовлення, що створюються на сайті;
- був розроблений ефективний алгоритм реалізації поставленого завдання;
- не менш важливою складовою проекту стало створення веб-сторінки додатку з урахуванням всіх вимог та функціональних можливостей.

Отримане програмне забезпечення надає користувачам декілька важливих можливостей. Перш за все, вони можуть формувати веб-сторінки, використовуючи шаблони та контент з бази даних, з можливістю сортування, фільтрації та пошуку. Крім того, вони можуть зв'язатися з адміністратором магазину та здійснювати замовлення товарів.

Весь проект був реалізований з використанням потужної бібліотеки React та мови програмування JavaScript, а також з використанням розмітки HTML та стилів SCSS. Це дозволило створити ефективну та сучасну систему, яка задовольняє високі вимоги користувачів у сфері електронної комерції.

В економічному розділі проведено розрахунок вартості роботи по створенню програми, яка становить близько 140 тис. грн. та розраховано час на його створення - 790 людино-годин, тобто 4,5 місяці.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Електронна комерція: переваги та недоліки [Електронний ресурс] // URL: <http://www.economy.nayka.com.ua/?op=1&z=4505> (дата звернення: 27.05.2023).
- 2 Кондіус І. С. Електронний посібник з Електронного бізнесу [Електронний ресурс] // URL: https://elib.lntu.edu.ua/sites/default/files/elib_upload/ (дата звернення: 27.05.2023).
- 3 МастХев для сучасного інтернет-магазину [Електронний ресурс] // URL: <https://web-systems.solutions/blog/musthave-dlya-suchasnogo-internet-magazynu/> (дата звернення: 27.05.2023) .
- 4 Загрози інформаційної безпеки [Електронний ресурс] // URL: https://uk.wikipedia.org/wiki/Загрози_інформаційної_безпеки (дата звернення: 27.05.2023).
- 5 "Web Development with Node and Express: Using the JavaScript Stack" by Ethan Brown, O'Reilly Media, Inc., Gravenstein Highway North, Севастополь, 2014, 2 с
- 6 «Murach's MySQL 2th Edition» by Joel Ant and Ray Harris: Mike Murach & Associates Inc. Cambridge, 2015. 20 с
- 7 "Learning MySQL: Get a Handle on Your Data" by Seyed M.M. (Saied) Tahaghoghi and Hugh E. Williams, O'reilly, Cambridge, 2006 135 с
- 8 PhpMyAdmin URL: <https://uk.wikipedia.org/wiki/PhpMyAdmin>
- 9 «HTML and CSS: Design and Build Websites» by Jon Duckett, John Wiley & Sons inc, Індіанаполіс, Індіана, 2011, 6 с, 431 с
- 10 «HTML & CSS: The Good Parts» by Ben Henick, O'Reilly Media, Inc., Gravenstein Highway North, Севастополь, 2010, 23 с
- 11 Sass Basics [Електронний ресурс] // URL: <https://sass-lang.com/guide/> (дата звернення: 14.06.2023).
- 12 JavaScript [Електронний ресурс] // URL: <https://astwellsoft.com/uk/blog/tehnology/javascript.html> (дата звернення: 14.06.2023).

- 13 Рейтинг мов програмування [Електронний ресурс] // URL: <https://ain.ua/2023/02/20/rejtyng-mov-programuvannya-2023/> (дата звернення: 15.06.2023).
- 14 «JavaScript: The Good Parts» by Douglas Crockford, O'Reilly Media, Inc., Gravenstein Highway North, Севастополь, 2008, 3 с
- 15 «JavaScript: The Missing Manual» by David Sawyer McFarland, O'Reilly Media, Inc., Gravenstein Highway North, Севастополь, 2008, 26 с
- 16 Most popular technologies URL: <https://survey.stackoverflow.co/2023/#most-popular-technologies-webframe-prof>
- 17 «Learning React: Functional Web Development with React and Redux» by Alex Banks and Eve Porcello, O'Reilly Media, Inc., Gravenstein Highway North, Севастополь, 2017, 2 с, 81 с
- 18 Вступ до JSX [Електронний ресурс] // URL: <https://uk.legacy.reactjs.org/docs/introducing-jsx.html> (дата звернення: 15.06.2023).
- 19 Node.js — що це? URL: <https://dan-it.com.ua/blog/chto-jeto-takoe-node-js-prostymi-slovami/> (дата звернення: 15.06.2023).
- 20 «Express in Action: Написання, створення та тестування програм Node.js», Еван М. Хан, Manning Publications, Shelter Island, 2016 , 34 с
- 21 Visual Studio Code [Електронний ресурс] // URL: https://uk.wikipedia.org/wiki/Visual_Studio_Code (дата звернення: 15.06.2023).

КОД ПРОГРАМИ

```

import React from 'react';
import styles from './Card_h_module.scss'
import { Link } from 'react-router-dom';

function Card_h({id, title, imageUrl,price, onAddCart, onFavorites , favorited =
false}){
  const [isAdded, setIsAdd] = React.useState(false);
  const [isFavorite, setIsFavorit] = React.useState(favorited);

  const onAddCart = () =>{
    onAddCart({title, imageUrl,price});
    setIsAdd(!isAdded);
  };
  const onFavorite =() => {
    onFavorites({title, imageUrl,price});
    setIsFavorit(!isFavorite)
  }

  return(
    <div className="card_h ">
      <img className="favoritesimg" height={50} width={50}
onClick={onFavorite}
      src={isFavorite ? '/img/like.png' : '/img/unlike.png'}
      />
      <Link className='titl1' to="/ProductPage"><img className="imgProd"
height={180} width={180} src={imageUrl}/>
      <h4 className='titl1'>{title}</h4></Link>

      <div className="d-flex justify-between align-center">
        <div className="d-flex flex-column ">
          <span>Ціна</span>
          <b>{price} грн.</b>
        </div>

        <img className={styles.plus} onClick={onAddCart} width={70} height={70}
src={isAdded ? '/img/add_to_cart.png' : '/img/to_cart.png'}
        />
      </div>
    </div>
  )
}
export default Card_h

```

ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ

ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім'я файлу	Опис
Пояснювальні документи	
Кваліфікаційна робота_Чернявський.doc	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Кваліфікаційна робота_Чернявський.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF.
Програма	
diplom.zip	Архів. Містить коди програми.
Презентація	
Презентація Чернявський.pptx	Презентація кваліфікаційної роботи.