

**МІНІСТЕРСТВО ФІНАНСІВ УКРАЇНИ
ДЕРЖАВНИЙ ПОДАТКОВИЙ УНІВЕРСИТЕТ**

**ФАКУЛЬТЕТ ФІНАНСІВ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І
СИСТЕМ**

**КВАЛІФІКАЦІЙНА РОБОТА
на тему:**

**РОЗРОБКА ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ
ВЕДЕННЯ ОБЛІКУ МАГАЗИНУ**

Здобувача вищої освіти першого (бакалаврського) рівня,
4 курсу, групи КБз-19-1, спеціальності 122 «Комп'ютерні науки»,
освітня програма «Інформаційні управляючі системи і технології (за галузями)»


(підпис)

Синегубової Марії Євгенівни
(прізвище, ім'я, по батькові)

Науковий керівник


(підпис)

Гладченко Оксана Вікторівна
(прізвище, ім'я, по батькові, науковий ступінь,
вчене звання)

Керівник освітньо-
професійної програми


(підпис)

Федотова - Півень Ірина
Миколаївна, к.т.н, доцент
(прізвище, ім'я, по батькові, науковий ступінь,
вчене звання)

Допущено до захисту за рішенням кафедри
протокол від «01» червня 2023 № 14

Завідувач кафедри


(підпис)

Філоненко Михайло Миколайович,
к.ф.-м.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене
звання)

Ірпінь – 2023

Реєстрація № 41

07.06.2023
(дата)


(підпис відповідальної особи)

Результати перевірки

До захисту
(до захисту, на доопрацювання)

01.06.2023
(дата)


(підпис)

Результат захисту

5 (відмінно)
(оцінка)

16.06.2023
(дата)


(підпис)

Голова комісії:


(підпис)

В.М. Франчук

Члени комісії:


(підпис)

М.М. Філоненко


(підпис)

В.О. Ніжегородцев


(підпис)

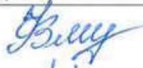
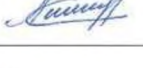
Є.В. Котух

БЛАНК ОЦІНЮВАННЯ ВИПУСКНОЇ РОБОТИ

Об'єкти контролю	Діапазон оцінки в балах	Фактична кількість балів	Примітки	Дата, П.І.Б.: керівника, голови комісії членів комісії, секретаря, (підписи)
ОЦІНЮВАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ КОМІСІЄЮ ПО ЗАХИСТУ				
1. Зміст випускної роботи (якість теоретичного матеріалу, його зв'язок з практичним, вміння представляти матеріал в логічній послідовності, наявність висновків та пропозицій)	від 0 до 30	29		16.06.2023 Франчук В.М.  Філоненко М.М.  Ніжегородцев В.О.  Котух С.В.  Суховерх Л.П. 
2. Дотримання вимог щодо оформлення	від 0 до 10	9		
3. Вчасне виконання роботи на всіх етапах у встановлені кафедрою терміни	від 0 до 5	5		
4. Рецензент	від 0 до 5	4		
РАЗОМ БАЛІВ	від 0 до 50	47		

Оцінювання кваліфікаційної роботи науковим керівником - 46 балів.

ПЕРЕВЕДЕННЯ ОЦІНКИ ЗА 100-БАЛЬНОЮ ШКАЛОЮ ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою	Дата	П.І.Б.: голови, членів комісії, секретаря, (підпис)
93	A	5 (відмінно)	16.06.2023	Франчук В.М.  Філоненко М.М.  Ніжегородцев В.О.  Котух С.В.  Суховерх Л.П. 

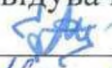
ПРАВИЛА ПЕРЕВЕДЕННЯ ОЦІНОК ДО НАЦІОНАЛЬНОЇ ШКАЛИ ТА ШКАЛИ ЄКТС

Оцінка за 100-бальною шкалою	Оцінка за шкалою ЄКТС	Оцінка за національною шкалою
90-100	A	5 (відмінно)
80-89	B	4 (добре)
70-79	C	
60-69	D	3 (задовільно)
50-59	E	
35-49	FX	2 (незадовільно)

Факультет фінансів та цифрових технологій
Кафедра комп'ютерних та інформаційних технологій і систем
Навчальна група: 12 «Інформаційні технології»
Спеціальність: 122 «Комп'ютерні науки»
Освітня програма: «Інформаційні управляючі системи і технології (за спеціальностями)»

Затверджую

Завідувач кафедри

 Філоненко М.М.
« 16 » 11 2022

Завдання на кваліфікаційну роботу
здобувачу вищої освіти
Синегубова Марія Євгенівна

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи

РОЗРОБКА ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ
ВЕДЕННЯ ОБЛІКУ МАГАЗИНУ

Затверджена наказом ректора від «15» листопада 2022 №1205 (зі змінами наказу від «04» квітня 2023 № 361)

2. Строк здачі закінченої роботи «26» травня 2023

3. Цільова установка

Розробити веб-орієнтовану інформаційну систему, дослідити та проаналізувати процес створення інформаційних систем.

Мета роботи : Розробка веб-орієнтованої інформаційної системи для ведення обліку інтернет-магазину на мові програмування Java Script.

Об'єкт дослідження : Розробка клієнтської та серверної частини програми.

Предмет дослідження : Розробка технологій, що дозволяють створювати функціональну та ефективну інформаційну систему для інтернет-магазину.

База дослідження : Веб-орієнтована інформаційна система

4. Вихідні дані до роботи : Технічна документація

5. Перелік графічного матеріалу : Рисунки, Таблиці

Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом):
Вступ. Розділ 1. Аналіз вимог та проєктування системи. 1.1. Огляд наукової та технічної літератури з питань розробки веб-орієнтованих інформаційних систем, ведення обліку в інтернет-магазинах та мови програмування Java Script. 1.2. Аналіз вимог до системи. 1.3. Вибір технологій реалізації системи. Розділ 2. Розробка та реалізація системи. 2.1. Опис структури та функціональності системи. 2.2. Реалізація системи мовою програмування Java Script. 2.3. Тестування системи. Розділ 3. Пропозиції за результатами дослідження. Висновки. Список використаних джерел.

6. Календарний план виконання роботи

№ п/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	
		За планом	Факти
1	2	3	4
1	Вибір теми, підбір літературних джерел, формування структури кваліфікаційної роботи	До 27.12.2022 р.	26.12.2022
2	I розділ (розкрити сутність і стан дослідження наукової проблематики роботи, після чого навести її загальну характеристику; огляд літератури за темою і вибір конкретних напрямів досліджень)	26.03.2023 р.	26.03.2023
3	II розділ (аналіз експериментальної частини про методики досліджень)	20.04.2023 р.	20.04.2023
4	III розділ (проведені теоретичні і (або) експериментальні дослідження та проектні рішення)	12.05.2023 р.	12.05.2023
5	Висновки	25.05.2023 р.	25.05.2023
6	Доопрацювання роботи з урахуванням зауважень наукового керівника	01.06.2023 р.	01.06.2023
7	Отримання відгуку керівника	03.06.2023 р.	03.06.2023
8	Отримання зовнішньої рецензії	05.06.2023р.	05.06.2023

8. Дата видачі завдання «16» листопада 2022

9. Керівник випускної кваліфікаційної роботи


(підпис)

Гладченко С
(прізвище, ініціали)

10. Керівник освітньо-професійної програми


(підпис)

Федотова-П
(прізвище, ініціали)

11. Завдання до виконання отримав здобувач вищої освіти


(підпис)

Синегубова
(прізвище, ініціали)

АНОТАЦІЯ

РОЗРОБКА ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ВЕДЕННЯ ОБЛІКУ МАГАЗИНУ

Метою дипломної роботи є розробка веб-орієнтованої інформаційної системи для ведення обліку інтернет-магазину на мові програмування Java Script. Розроблена система повинна бути надійною, швидкою та зручною в користуванні. У роботі будуть розглянуті основні принципи розробки веб-систем, зокрема архітектура клієнт-сервер, мова програмування Java Script, а також методології розробки програмного забезпечення.

Завданням дипломної роботи є аналіз потреб та вимог магазину, проектування структури бази даних, розробка користувацького інтерфейсу, розробка функціональності системи, тестування та валідація.

Предметом дослідження виступає розробка технологій, що дозволяють створювати функціональну та ефективну інформаційну систему для інтернет-магазину. Це включає дослідження архітектури системи, дизайн інтерфейсу користувача, функціональності системи та її інтеграції з іншими системами, безпеки даних, процесів взаємодії користувачів з системою, оптимізації швидкодії та інших аспектів, які впливають на якість та ефективність роботи інформаційної системи.

Об'єктом дослідження у розробці веб-орієнтованої інформаційної системи для ведення обліку інтернет-магазину на мові програмування JavaScript є розробка клієнтської та серверної частини програми.

Базою дослідження виступає реальний магазин, який обрано як об'єкт дослідження. Дослідження буде здійснюватись на основі спостережень, аналізу та збору даних з цього магазину.

За результатами дослідження, виявлено потребу магазину у впровадженні інформаційної системи для ведення обліку. Існуючі методи та процеси обліку не є ефективними та вимагають значних зусиль і ресурсів; визначено ключові

функціональні вимоги до веб-орієнтованої інформаційної системи, включаючи збір та збереження даних про товари, замовлення, клієнтів, складський облік та фінансові операції; встановлено, що використання веб-технологій та бази даних дозволить забезпечити доступ до системи з будь-якого пристрою з підключенням до Інтернету, спростить процеси обліку та підвищить ефективність роботи магазину.

Отримані результати дослідження можуть бути використані для розробки та впровадження веб-орієнтованої інформаційної системи для ведення обліку в магазині. Розроблені концепція та функціонал системи, архітектурні рішення та вибрані технології можуть слугувати основою для подальшої розробки та реалізації системи. А також для оптимізації робочих процесів в магазині. Розроблений функціонал системи, який передбачає автоматизацію процесів замовлення, обліку товарів, клієнтів та фінансових операцій, дозволить зменшити час та зусилля, потрібні для виконання цих завдань, та забезпечить більш точний облік.

Кваліфікаційна робота містить 56 сторінок, 1 таблицю, 25 рисунків, список використаних джерел 21 найменувань, додатків.

Ключові слова: розробка веб-орієнтованої інформаційної системи, Java Script, інтернет-магазин.

Рік виконання кваліфікаційної роботи: 2023 р.

Рік захисту роботи: 2023 р.

ANNOTATION

DEVELOPMENT OF A WEB-BASED INFORMATION SYSTEM FOR STORE MANAGEMENT

The aim of the thesis is to develop a web-based information system for managing the accounting of an online store using the Java Script programming language. The developed system should be reliable, fast, and user-friendly. The work will cover the fundamental principles of web system development, including client-server architecture, Java Script programming language, and software development methodologies.

The tasks of the thesis include analyzing the needs and requirements of the store, designing the database structure, developing the user interface, implementing system functionality, testing, and validation.

The subject of the research is the development of technologies that allow creating a functional and efficient information system for an online store. This includes studying the system architecture, user interface design, system functionality, integration with other systems, data security, user-system interaction processes, performance optimization, and other aspects that affect the quality and efficiency of the information system's operation.

The object of the research in the development of the web-based information system for store management using the JavaScript programming language is the development of the client and server parts of the program.

The research is based on a real store chosen as the object of study. The research will be conducted through observations, analysis, and data collection from this store.

Based on the research results, the need for implementing an information system for store management has been identified. The existing accounting methods and processes are not efficient and require significant efforts and resources. Key functional requirements for the web-based information system have been established, including data collection and storage for products, orders, customers, inventory management, and financial transactions. It has been determined that the use of web

technologies and databases will enable access to the system from any device with an Internet connection, streamline accounting processes, and improve the store's efficiency.

The obtained research results can be used for the development and implementation of a web-based information system for store management. The developed system concept, functionality, architectural solutions, and selected technologies can serve as a basis for further development and implementation of the system, as well as for optimizing workflow processes in the store. The implemented system functionality, which includes the automation of ordering processes, product management, customer management, and financial operations, will reduce the time and effort required to perform these tasks and ensure more accurate accounting.

The qualification work consists of 56 pages, 1 table, 25 figures, a list of 21 references, and appendices.

Keywords: web-based information system development, Java Script, online store.

Year of qualification work: 2023.

Year of thesis defense: 2023.

ЗМІСТ

РОЗРОБКА ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ВЕДЕННЯ ОБЛІКУ МАГАЗИНУ	1
ВСТУП	11
РОЗДІЛ 1. АНАЛІЗ ВИМОГ ТА ПРОЄКТУВАННЯ СИСТЕМИ	14
1.1. Огляд наукової та технічної літератури з питань розробки веб-орієнтованих інформаційних систем, ведення обліку в інтернет-магазинах та мови програмування JAVA SCRIPT	14
1.2. Аналіз вимог до системи	19
1.3. Вибір технологій реалізації системи	24
РОЗДІЛ 2. РОЗРОБКА ТА РЕАЛІЗАЦІЯ СИСТЕМИ	25
2.1. Опис структури та функціональності системи	25
2.2. Реалізація системи мовою програмування JAVA SCRIPT	32
2.3. Тестування системи	39
РОЗДІЛ 3. ПРОПОЗИЦІЇ ЗА РЕЗУЛЬТАТАМИ ДОСЛІДЖЕННЯ	45
ВИСНОВКИ	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	49
ДОДАТКИ	51

ВСТУП

Інтернет-магазини стали неодмінним атрибутом сучасного онлайн-світу. Для ефективної роботи такого магазину потрібна надійна система ведення обліку замовлень, клієнтів, товарів та інших важливих показників. У зв'язку з цим, розробка веб-орієнтованої інформаційної системи є актуальною задачею в сфері електронної комерції.

Метою дипломної роботи є розробка веб-орієнтованої інформаційної системи для ведення обліку інтернет-магазину мовою програмування Java Script. Розроблена система повинна бути надійною, швидкою та зручною в користуванні. У роботі розглянуто основні принципи розробки веб-систем, зокрема архітектура клієнт-сервер, мова програмування Java Script, а також методології розробки програмного забезпечення.

Особливість розробки інформаційної системи полягає в тому, що вона розроблена мовою програмування Java Script, яка є однією з найпопулярніших мов програмування у світі. Крім того, у роботі використано найновіші технології розробки веб-додатків, що дозволить розробити систему з високою продуктивністю та функціональністю.

Багато відомих дослідників та розробників здійснили значний вплив у розробку веб-інформаційних систем на мові JavaScript. Наприклад:

1. Брендан Айк. Він є співавтором мови JavaScript і відіграв важливу роль у її розробці. Айк працював у Netscape Communications, де була створена перша версія JavaScript (тоді вона називалась Mocha, потім LiveScript, а пізніше отримала назву JavaScript). Він також допомагав визначити стандартизацію JavaScript і став одним зі співзасновників Mozilla.

2. Дуглас Крокфорд. Цей відомий дослідник та програміст має великий вплив на розробку JavaScript. Він вніс значний внесок у стандартизацію мови JavaScript, популяризуючи правильні практики програмування і розробивши багато інструментів для поліпшення якості коду, зокрема JSLint і JSON.

3. Джон Резіг. Він відомий своїми внесками у розробку JavaScript-бібліотеки jQuery, яка стала популярним інструментом для спрощення роботи з DOM та взаємодії з JavaScript на веб-сторінках. Резіг також вніс вагомий внесок у розробку багатьох інших веб-технологій.

4. Андерс Хейлсберг. Він є головним архітектором мови TypeScript, яка є розширенням JavaScript з підтримкою типізації. TypeScript дозволяє розробникам створювати складні веб-додатки з вищим рівнем безпеки та продуктивності. Хейлсберг також був важливим фігурантом у розробці інших важливих продуктів Microsoft, таких як C# та .NET Framework. (додала інформацію про відомих дослідників).

Дипломна робота складається з вступу, основної частини, в якій будуть розглянуті теоретичні аспекти розробки веб-орієнтованої інформаційної системи та практичної частини, яка передбачає розробку та тестування розробленої системи. В результаті дослідження та розробки буде отримана веб-орієнтована інформаційна система для ведення обліку інтернет-магазину, яка дозволить оптимізувати процеси взаємодії з клієнтами, управління товарним асортиментом, контроль за замовленнями та іншими важливими процесами.

У результаті розробки веб-орієнтованої інформаційної системи для ведення обліку інтернет-магазину мовою програмування Java Script, можна отримати значні переваги в роботі інтернет-магазину, зокрема збільшення продуктивності, зниження часу на обробку замовлень, збільшення задоволеності клієнтів та збільшення прибутку. Отже, розробка веб-орієнтованої інформаційної системи для ведення обліку інтернет-магазину мовою програмування Java Script є актуальною задачею в сучасному онлайн-світі, яка може принести значну користь для бізнесу та його клієнтів.

Для реалізації цієї задачі, в дипломній роботі проведено детальний аналіз потреб і вимог інтернет-магазинів та розглянуто існуючі рішення у цій галузі. На основі отриманої інформації буде розроблено веб-орієнтована інформаційна система мовою програмування Java Script з використанням сучасних технологій веб-розробки та баз даних.

Дана дипломна робота має на меті дослідити можливості та переваги використання мови програмування Java Script для розробки веб-орієнтованих інформаційних систем в контексті інтернет-магазинів. Для досягнення цієї мети проведено аналіз сучасних тенденцій у веб-розробці та існуючих рішень у галузі інтернет-магазинів на основі попередніх досліджень.

В дипломній роботі використано такі методи дослідження, як аналіз літератури та наукових джерел, аналіз вимог та потреб інтернет-магазинів, проектування веб-орієнтованої інформаційної системи, програмування на мові Java Script та тестування отриманої системи.

Отже, результатом даної дипломної роботи є розроблена веб-орієнтована інформаційна система для ведення обліку інтернет-магазину мовою програмування Java Script, яка спроможна відповідати потребам сучасного онлайн-бізнесу та підвищувати ефективність роботи інтернет-магазинів.

Основні завдання дипломної роботи полягають у наступному:

- Провести аналіз вимог та потреб інтернет-магазинів у веб-орієнтованій інформаційній системі для ведення обліку.
- Дослідити сучасні технології веб-розробки та баз даних, які можуть бути використані для розробки інформаційної системи.
- Розробити веб-орієнтовану інформаційну систему для ведення обліку інтернет-магазину мовою програмування Java Script з використанням сучасних технологій веб-розробки та баз даних.
- Протестувати розроблену систему та забезпечити її високу якість та надійність.
- Оцінити ефективність використання мови програмування Java Script у розробці веб-орієнтованих інформаційних систем для інтернет-магазинів.

Результати досліджень та розробки даної дипломної роботи можуть бути корисними для компаній, які займаються розробкою веб-орієнтованих інформаційних систем для інтернет-магазинів, а також для веб-розробників та

студентів, які цікавляться цією галуззю та мають бажання збільшити свої знання та навички у розробці веб-додатків мовою програмування Java Script.

РОЗДІЛ 1. АНАЛІЗ ВИМОГ ТА ПРОЄКТУВАННЯ СИСТЕМИ

1.1. Огляд наукової та технічної літератури з питань розробки веб-орієнтованих інформаційних систем, ведення обліку в інтернет-магазинах та мови програмування JAVA SCRIPT

Огляд наукової та технічної літератури з питань розробки веб-орієнтованих інформаційних систем, ведення обліку в інтернет-магазинах та мови програмування Java Script відображає сучасні тенденції та розробки у сфері веб-програмування та електронної комерції.

Одним із найважливіших аспектів розробки веб-орієнтованих інформаційних систем є створення дизайну інтерфейсу користувача. Це дозволяє забезпечити ефективне взаємодію з користувачем та зробити його досвід використання системи якомога більш приємним та зручним. Дослідження з цієї області показали, що успіх веб-дизайну полягає у зручності та інтуїтивній зрозумілості інтерфейсу, а також у здатності до адаптації до різних пристроїв та платформ.

З урахуванням розширення електронної комерції, ведення обліку в інтернет-магазинах є необхідною складовою процесу. Забезпечення безпеки інформації та ефективного управління складом є ключовими факторами, які впливають на успіх електронної комерції. Дослідження показують, що використання баз даних та спеціальних програм може значно полегшити цей процес, зменшити ризик втрати даних та забезпечити ефективний контроль над складом.

Мова програмування JavaScript є однією з найпопулярніших мов програмування у веб-розробці. Вона дозволяє розробникам створювати динамічні та інтерактивні інтерфейси користувача, що забезпечують покращений досвід взаємодії та зручність використання для користувачів. Крім того, JavaScript забезпечує зв'язок між клієнтом та сервером та дозволяє виконувати асинхронні запити до сервера, що дозволяє збільшити продуктивність та швидкість роботи веб-додатків.

У наукових статтях та публікаціях з веб-розробки та електронної комерції активно обговорюються різні аспекти використання JavaScript у веб-розробці, включаючи його можливості для розробки додатків, розробку веб-фреймворків, створення мультимедійних ефектів та багато іншого. Також активно вивчаються питання безпеки використання JavaScript, зокрема, забезпечення захисту від кросс-сайтових скриптів та інших видів атак на веб-додатки.

У загальному, вивчення наукової та технічної літератури з питань розробки веб-орієнтованих інформаційних систем, ведення обліку в інтернет-магазинах та мови програмування JavaScript є необхідним для успішної розробки та впровадження веб-додатків та електронної комерції. Застосування цих знань та практик у розробці може допомогти підвищити продуктивність, ефективність та безпеку веб-додатків, забезпечивши при цьому більш зручний та приємний досвід користувача.

Одним з напрямків досліджень у галузі веб-розробки та електронної комерції є розробка алгоритмів та методів аналізу даних, що дозволяють відстежувати поведінку користувачів на веб-сайтах та в інтернет-магазинах. Ці дані можуть бути використані для підвищення ефективності рекламних кампаній, прогнозування збуту товарів та послуг, а також для забезпечення більш персоналізованого досвіду користувача.

У сучасних дослідженнях також активно обговорюються нові технології та методики розробки веб-додатків, такі як Progressive Web Apps (PWA) та Single-Page Applications (SPA), що дозволяють покращити продуктивність та швидкість роботи веб-додатків та забезпечити кращий досвід користувача.

Нарешті, наукові та технічні дослідження з питань веб-розробки та електронної комерції допомагають визначати та вирішувати проблеми, які виникають у процесі розробки веб-додатків та електронної комерції, такі як захист від кібератак, оптимізація продуктивності та швидкості роботи, покращення інтерфейсів та зручності використання для користувачів.

У підсумку, наукова та технічна література з питань розробки веб-орієнтованих інформаційних систем, ведення обліку в інтернет-магазинах та мови програмування JavaScript є важливим ресурсом для підвищення рівня

знань та професійної компетентності у цій галузі, що може допомогти в розробці та впровадженні успішних та інноваційних веб-додатків та електронної комерції. Знання та розуміння останніх тенденцій та новітніх технологій є важливим елементом в сучасному світі бізнесу та технологій, і можуть допомогти підприємствам та розробникам успішно конкурувати та розвиватися.

Зокрема, вивчення мови програмування JavaScript є надзвичайно важливим для розробки веб-додатків та створення динамічних інтерфейсів, що дозволяють користувачам взаємодіяти з додатками без необхідності перезавантаження сторінки. Крім того, знання JavaScript дозволяє використовувати бібліотеки та фреймворки, такі як React, Angular та Vue, для підвищення ефективності та швидкості розробки веб-додатків[4].

Окрім цього, ведення обліку в інтернет-магазинах є важливим елементом в електронній комерції, який дозволяє відстежувати замовлення, платежі та доставку товарів. Для цього використовуються спеціальні системи обліку, такі як Shopify, Magento, WooCommerce, які дозволяють забезпечити ефективність та точність процесів обробки замовлень.

Усе вищенаведене підкреслює важливість знання та розуміння наукової та технічної літератури з питань розробки веб-орієнтованих інформаційних систем, ведення обліку в інтернет-магазинах та мови програмування JavaScript. Це допомагає розробникам та підприємствам розуміти та використовувати нові технології та методики розробки, що дозволяє покращити ефективність та конкурентоспроможність їхніх продуктів та послуг. Наукова та технічна література є важливим джерелом інформації для розробників, які хочуть бути в курсі останніх тенденцій у своїй галузі та вдосконалювати свої навички.

Наприклад, серед недавніх публікацій можна виділити книгу "Eloquent JavaScript: A Modern Introduction to Programming" автора Маріїн Хавербеке, яка описує сучасні методики програмування на JavaScript та допомагає розуміти його основні концепції. Ця книга є важливим ресурсом для розробників, які хочуть покращити свої навички в програмуванні на JavaScript[2].

Також наукова та технічна література з питань електронної комерції включає в себе багато практичних посібників та книг зі стратегій маркетингу, аналізу даних та управління продуктами. Наприклад, книга "The Lean Product Playbook: How to Innovate with Minimum Viable Products and Rapid Customer Feedback" автора Дена Олсена, яка описує методики створення продуктів з мінімальними витратами та швидким отриманням зворотного зв'язку від клієнтів, може бути корисною для підприємців, які планують створити свій власний інтернет-магазин[6].

Отже, наукова та технічна література з питань розробки веб-орієнтованих інформаційних систем, ведення обліку в інтернет-магазинах та мови програмування JavaScript є важливим джерелом знань та інформації для розробників та підприємців. Розуміння останніх тенденцій та новітніх технологій допомагає підвищити ефективність та конкурентоспроможність продуктів та послуг, що пропонується в інтернеті. Важливо також зазначити, що наукова та технічна література не є єдиним джерелом інформації, але вона може бути корисною для тих, хто бажає отримати глибокі знання та розуміння технічних та наукових аспектів своєї роботи.

Крім того, важливим елементом розробки веб-орієнтованих інформаційних систем та інтернет-магазинів є знання мови програмування JavaScript. Ця мова вважається однією з найпоширеніших мов програмування для веб-розробки, тому знання її основних концепцій та методів є необхідним для розробників. Останні роки були важливими для мови програмування JavaScript, оскільки з'явилися нові версії та бібліотеки, які значно полегшують розробку веб-додатків.

Наприклад, однією з нових технологій на базі JavaScript є React.js – бібліотека для створення інтерфейсів користувача, яка дозволяє створювати високопродуктивні додатки, що працюють в режимі реального часу. Також важливим елементом веб-розробки є знання CSS та HTML, які допомагають створювати веб-сторінки та забезпечувати їх правильний вигляд[8].

Отже, знання наукової та технічної літератури з питань розробки веб-орієнтованих інформаційних систем, ведення обліку в інтернет-магазинах та

мови програмування JavaScript це невід’ємна частина аналізу та проєктування для розробників та підприємців, які бажають успішно функціонувати в інтернет-середовищі та забезпечити ефективну роботу своїх продуктів.

Крім того, у світі існує велика кількість онлайн-курсів та ресурсів, які надають доступ до безкоштовної та платної наукової та технічної літератури, що дозволяє збільшити свій рівень знань та навичок. Такі ресурси, як Coursera, edX, Udacity, Codecademy та багато інших, пропонують широкий вибір онлайн-курсів та інших навчальних ресурсів з веб-розробки та мов програмування, що допомагає розширювати знання та отримувати нові навички.

Окрім цього, важливим елементом успішної розробки веб-додатків та інтернет-магазинів є знання процесів та методів управління проєктами. Наприклад, методологія Scrum дозволяє забезпечити ефективне управління проєктом та підвищити якість продукту. Вивчення цих методів допоможе підприємцям та розробникам ефективніше керувати своїми проєктами та досягати більш якісних результатів.

У підсумку, знання наукової та технічної літератури з питань розробки веб-орієнтованих інформаційних систем, ведення обліку в інтернет-магазинах та мови програмування JavaScript є необхідним для успішного функціонування в інтернет-середовищі. Важливо також звернути увагу на вивчення процесів та методів управління проєктами, що дозволяє досягати більш ефективних результатів у розробці веб-додатків та інтернет-магазинів.

Крім того, потрібно мати практичний досвід у розробці веб-додатків та інтернет-магазинів, який можна отримати шляхом виконання проєктів та стажувань. Застосування здобутих знань у практиці допоможе покращити розуміння теорії та набути практичного досвіду у розробці веб-додатків та інтернет-магазинів.

Також важливо бути в курсі новинок та тенденцій у галузі веб-розробки та програмування, оскільки ця сфера швидко розвивається та змінюється. Розуміння новітніх технологій та інструментів дозволяє підтримувати конкурентоспроможність та відповідність до вимог ринку.

Нарешті, мати підтримку спільноти та можливість обміну досвідом та знаннями з іншими розробниками та підприємцями – це також невід’ємна частина вивчення програмування та вдосконалення своїх навичок. Для цього можна долучитися до спільнот, таких як Stack Overflow, GitHub, Reddit, де можна обговорювати технічні питання, ділитися досвідом та знаннями та отримувати підтримку та поради від інших фахівців.

Отже, наукова та технічна література з питань розробки веб-орієнтованих інформаційних систем, ведення обліку в інтернет-магазинах та мови програмування JavaScript є ключовими для успішної роботи в інтернет-середовищі. Окрім того, важливо мати практичний досвід, бути в курсі новинок та тенденцій у галузі та мати підтримку спільноти.

1.2. Аналіз вимог до системи

Аналіз вимог до веб-орієнтованої інформаційної системи для ведення обліку інтернет-магазину є ключовим етапом в розробці такої системи. Він допомагає визначити функціональні вимоги та необхідність додаткових функцій, щоб забезпечити ефективну та надійну роботу інформаційної системи.

Основні функціональні вимоги до веб-орієнтованої інформаційної системи для ведення обліку інтернет-магазину включають:

1. Управління каталогом товарів: система повинна дозволяти додавати нові товари, редагувати існуючі, видаляти товари з каталогу, робити категоризацію товарів.

2. Керування замовленнями: система повинна дозволяти користувачам робити замовлення згідно зібраної інформації на основі ресурсів, на яких розміщені оголошення та зберігати інформацію про них. Адміністратор повинен мати можливість переглядати та редагувати замовлення, стежити за їх статусом.

3. Управління складом: система повинна дозволяти відслідковувати кількість товарів на складі, отримувати повідомлення про закінчення запасів, автоматично зменшувати кількість товарів на складі при оформленні замовлення.

4. Управління користувачами: система повинна дозволяти реєструвати користувачів, зберігати їхню особисту інформацію, забезпечувати авторизацію та автентифікацію користувачів.

5. Аналітика та звітність: система повинна дозволяти отримувати статистики та аналізувати дані про продажі, відвідуваність сайту та іншу важливу інформацію. Для цього система повинна мати можливість збирати, аналізувати та візуалізувати дані.

6. Безпека: система повинна забезпечувати захист від хакерських атак та інших видів зломів, забезпечувати безпеку обміну даними та оплати[1].

Крім функціональних вимог, система повинна задовольняти і нефункціональні вимоги, такі як швидкість роботи, масштабованість, надійність, зручність використання та інші.

Враховуючи всі вимоги до веб-орієнтованої інформаційної системи для ведення обліку інтернет-магазину, розробники можуть створити ефективну та надійну систему, яка задовольнить потреби та користувачів.

Для розробки веб-орієнтованої інформаційної системи для ведення обліку інтернет-магазину можна використовувати різні технології та мови програмування. Однією з таких мов є JavaScript, яка є однією з найбільш поширених мов програмування для веб-розробки.

JavaScript дозволяє створювати динамічні та інтерактивні елементи на сторінках веб-сайтів, такі як форми, кнопки, меню, слайдери та інші. JavaScript дозволяє також забезпечувати валідацію введених даних користувачем та здійснювати перевірку правильності введення.

Однією з популярних технологій для розробки веб-орієнтованих інформаційних систем є Node.js, яка дозволяє використовувати JavaScript для створення серверної частини додатків. Node.js дозволяє створювати високопродуктивні та масштабовані додатки, які можуть обслуговувати тисячі користувачів одночасно.

Для забезпечення безпеки та захисту від хакерських атак, можна використовувати різні технології, такі як SSL-шифрування, двофакторна аутентифікація, механізми захисту від переповнення буфера та інші.

Узагалі, розробка веб-орієнтованої інформаційної системи для ведення обліку інтернет-магазину є складним та багатоплановим процесом, який потребує досвідченого та кваліфікованого персоналу та використання сучасних технологій. Важливо також бути в курсі останніх тенденцій та новин в галузі веб-розробки, щоб розробляти систему, яка буде задовольняти потреби користувачів.

Крім того, для ведення обліку в інтернет-магазині можна використовувати різні бази даних, такі як MySQL, PostgreSQL, MongoDB та інші. Вибір бази даних залежить від конкретних вимог та потреб системи, а також від кількості користувачів та обсягів даних, які необхідно зберігати.

Також важливим елементом веб-орієнтованої інформаційної системи для ведення обліку в інтернет-магазині є система керування контентом (Content Management System, CMS). CMS дозволяє зручно та ефективно керувати контентом на сайті, таким як товари, замовлення, користувачі та інші.

Для забезпечення зручності та зручного користування системою, важливо створити простий та зрозумілий інтерфейс користувача, який буде добре працювати на різних пристроях та з різних браузерів.

Нарешті, для успішної розробки веб-орієнтованої інформаційної системи для ведення обліку в інтернет-магазині важливо мати хорошу знайомість з різними стандартами та протоколами, такими як HTML, CSS, HTTP та інші.

Отже, розробка веб-орієнтованої інформаційної системи для ведення обліку в інтернет-магазині є складним та багатогранним процесом, який потребує розгляду різних аспектів та використання різних технологій. Важливо розуміти потреби користувачів та забезпечити зручність та безпеку системи.

Нарешті, розробка веб-орієнтованої інформаційної системи для ведення обліку в інтернет-магазині повинна бути підтримувана та оновлювана відповідно до змін у потребах користувачів та ринку. Важливо забезпечити

підтримку та технічну підтримку користувачів, а також оновлювати систему для забезпечення безпеки та оптимальної роботи.

Отже, веб-орієнтована інформаційна система для ведення обліку в інтернет-магазині є надзвичайно важливим елементом успішної діяльності магазину. Вона забезпечує зручне та ефективне керування даними, які є ключовими для успішного функціонування бізнесу. Розробка такої системи потребує розгляду різних аспектів та використання різних технологій, включаючи мову програмування JavaScript, бази даних та системи керування контентом. Важливо забезпечити зручний та безпечний інтерфейс користувача та підтримувати систему оновленою та підтриманою, щоб вона відповідала потребам користувачів та змінювалася відповідно до ринкових потреб.

Крім того, при розробці веб-орієнтованої інформаційної системи для ведення обліку в інтернет-магазині потрібно враховувати кілька ключових вимог. Перш за все, система повинна бути надійною та безпечною. Для цього можна використовувати різні методи та технології, такі як шифрування даних, захист від вірусів та інших зловмисних програм, забезпечення резервного копіювання даних тощо.

Крім того, важливо забезпечити зручний та інтуїтивно зрозумілий інтерфейс користувача. Система повинна бути легкою у використанні та мати зрозумілі та логічні функції та можливості. Також важливо забезпечити можливість доступу до системи з різних пристроїв та різних місць, що дозволить користувачам керувати своїм магазином з будь-якого місця у світі.

До інших важливих вимог до веб-орієнтованої інформаційної системи для ведення обліку в інтернет-магазині можна віднести підтримку багатомовності, можливість інтеграції з різними платіжними системами та сервісами доставки товарів, а також можливість інтеграції з соціальними мережами та іншими інтернет-сервісами.

Отже, при розробці веб-орієнтованої інформаційної системи для ведення обліку в інтернет-магазині необхідно враховувати багато різних факторів та вимог. Це може бути складним та витратним процесом, але результатом

повинна стати надійна та ефективна система, яка допоможе забезпечити успішну діяльність інформаційного веб-ресурсу.

На підставі зібраних вимог можна скласти список функцій та можливостей, які має мати веб-орієнтована інформаційна система для ведення обліку в інтернет-магазині. Основні функції інформаційної системи повинні включати такі можливості:

- Управління товарами – система повинна дозволяти додавати, редагувати та видаляти товари з каталогу, а також встановлювати їх ціну та кількість.
- Управління замовленнями – система повинна дозволяти адміністраторам розміщувати замовлення та стежити за їх статусом, а також дозволяти адміністраторам змінювати статус замовлень.
- Управління користувачами – система повинна дозволяти реєструватися користувачам, зберігати їхні дані та історію замовлень.
- Аналітика – система повинна збирати та обробляти дані про замовлення, товари та користувачів, а також дозволяти адміністраторам створювати звіти та аналізувати ці дані для покращення ефективності роботи інтернет-магазину.
- Керування запасами – система повинна дозволяти контролювати рівень запасів товарів[15].

Зважаючи на всі вищезазначені вимоги, розробка веб-орієнтованої інформаційної системи для ведення обліку інтернет-магазину є складним і багатоетапним процесом, який вимагає глибоких знань у галузі програмування, веб-технологій та бізнес-процесів, що залучаються у проєкті. Однак, вірно розроблена система дозволить оптимізувати процеси управління інтернет-магазином, збільшити ефективність роботи персоналу, знизити ризики виникнення помилок та збільшити задоволеність клієнтів.

У зв'язку з тим, що розробка веб-орієнтованої інформаційної системи для ведення обліку інтернет-магазину є складним технічним завданням,

рекомендується звернутися до досвідчених фахівців у даній області або залучити професійну компанію для розробки системи. Також важливо взяти до уваги всі потреби інтернет-магазину та клієнтів, щоб система відповідала всім вимогам і була ефективною у використанні.

Нарешті, після розробки системи, вона потребуватиме постійного підтримання та оновлення, щоб вона залишалася актуальною та безпечною у використанні. Тому важливо мати знання та навички у підтримці та розвитку веб-орієнтованих інформаційних систем.

В цілому, розробка веб-орієнтованої інформаційної системи для ведення обліку інтернет-магазину є складним та важливим процесом, що може вплинути на успішність бізнесу. Однак, з дотриманням вимог та кращих практик розробки, використання сучасних технологій та підходів, а також постійним вдосконаленням системи, можна досягти більшої ефективності та задоволення потреб клієнтів. Крім того, важливо не забувати про безпеку даних, щоб уникнути можливих проблем з їх втратою чи розкриттям. Розробка інформаційної системи повинна бути комплексним процесом, який враховує потреби бізнесу та клієнтів, а також забезпечує надійність та ефективність управління даними.

1.3. Вибір технологій реалізації системи

Для успішної розробки веб-орієнтованої інформаційної системи для ведення обліку інтернет-магазину мовою програмування Java Script було необхідно обрати найбільш відповідні технології. Зокрема, для створення інтерфейсу користувача було обрано React.js, HTML та CSS. React.js – це потужна JavaScript-бібліотека, яка забезпечує зручну роботу зі складними інтерфейсами. HTML та CSS, своєю чергою, відповідають за структуру та візуальне оформлення веб-сторінок.

Для забезпечення потрібного стану додатку та управління станом було використано MobX. Це дозволяє зручно та ефективно керувати станом компонентів та їх взаємодії. Node.js та Express були використані для розробки серверної частини додатку. Node.js – це платформа для виконання JavaScript на

сервері, що дозволяє створювати масштабовані та ефективні додатки. Express – це популярний фреймворк для розробки серверної частини додатків на Node.js.

Для зберігання даних було використано PostgreSQL та Sequelize. PostgreSQL – це потужна та надійна реляційна база даних, яка забезпечує стабільне та швидке зберігання даних. Sequelize – це ORM (Object-Relational Mapping) для Node.js, що дозволяє зручно та ефективно взаємодіяти з базою даних.

Node.js та Express виступають у ролі серверної частини системи та забезпечують обробку запитів користувачів, взаємодію з базою даних та забезпечують надійний та швидкий доступ до даних. React.js, HTML та CSS відповідають за розробку клієнтської частини системи та дозволяють забезпечити гарний та зручний інтерфейс користувача.

MobX використовується для забезпечення станового управління та зберігання даних, що дозволяє забезпечити більш ефективну та швидку роботу системи та полегшити розробку. Загалом, обрані технології та інструменти надають можливість розробити високоякісну, функціональну та ефективну веб-орієнтовану інформаційну систему для ведення обліку інтернет-магазину[10].

Окрім того, для забезпечення безпеки даних використовується механізм шифрування та автентифікації, який забезпечується за допомогою бібліотеки Passport.js. Для тестування коду та забезпечення його якості використовується Jest – один з найбільш популярних фреймворків для тестування JavaScript-коду.

Таким чином, обрані технології та інструменти надають можливість розробити високоякісну та функціональну веб-орієнтовану інформаційну систему для ведення обліку інтернет-магазину, яка забезпечує ефективний та безпечний процес управління товарообігом та клієнтською базою.

РОЗДІЛ 2. РОЗРОБКА ТА РЕАЛІЗАЦІЯ СИСТЕМИ

2.1. Опис структури та функціональності системи

Розробка будь-якого продукту, це довготривалий та ресурснозатратний процес, який відбувається в декілька етапів.

Першим етапом розробки системи є збір вимог, де визначається функціональність, яку повинна мати система. На цьому етапі здійснюється аналіз потреб користувачів та бізнес-вимог, щоб визначити основні вимоги до продукту.

Після збору вимог проводиться етап проектування архітектури, де визначається структура програмного продукту, його компоненти та їх взаємодія. На цьому етапі визначається технологічний стек, який буде використовуватись для реалізації системи.

Далі йде етап розробки функціональності, де розробляється програмний код, що реалізує вимоги, визначені на попередньому етапі. Важливим аспектом на цьому етапі є тестування розробленої функціональності, щоб переконатися в тому, що вона працює належним чином.

Останнім етапом є реалізація системи, де збираються всі розроблені компоненти та функціональності в єдину систему. На цьому етапі проводиться інтеграція та тестування системи в цілому.

Структура проєкту в основному полягає в розділенні додатку на дві його складові: серверна частина та клієнтська. Серверна частина додатку відповідає за зберігання та обробку даних, а також забезпечення взаємодії між клієнтською частиною (зазвичай це веб- або мобільний додаток) та базою даних. Вона забезпечує обробку запитів від користувачів, перевірку коректності введених даних, збереження інформації та відправку результатів запитів до клієнтської частини додатку. Зазвичай серверна частина складається з таких компонентів, як веб-сервер, додаткових служб та баз даних, а також API (Application Programming Interface), яке забезпечує стандартизований інтерфейс для взаємодії з клієнтською частиною додатку.

Також серверна частина може забезпечувати розширення функціональності додатку через розробку додаткових модулів та плагінів, а також інтеграцію з іншими системами та сервісами. В цілому, серверна частина додатку відповідає за забезпечення безперебійної та ефективної роботи програмного продукту, а також за забезпечення високої швидкості роботи та збереження даних користувачів в безпеці.

Клієнтська частина додатку відповідає за інтерфейс взаємодії з користувачем. Вона забезпечує відображення даних, отриманих з серверної частини, та забезпечує користувачам можливість взаємодії з додатком. Клієнтська частина може бути розроблена для різних платформ, наприклад, для веб-браузерів, мобільних пристроїв або настільних комп'ютерів. Вона забезпечує зручний та інтуїтивний інтерфейс для користувачів, який дозволяє їм взаємодіяти з додатком за допомогою різних елементів керування, таких як кнопки, поля вводу, списки та інші. Клієнтська частина також може забезпечувати реалізацію функціональності додатку на стороні клієнта, наприклад, обробку даних на клієнтському пристрої, що дозволяє знизити навантаження на серверну частину. Крім того, клієнтська частина може забезпечувати можливість використання додатку без підключення до Інтернету або з поганою якістю зв'язку, за допомогою кешування даних на клієнтському пристрої.

Структура серверної частини зображена на рисунку 2.1.

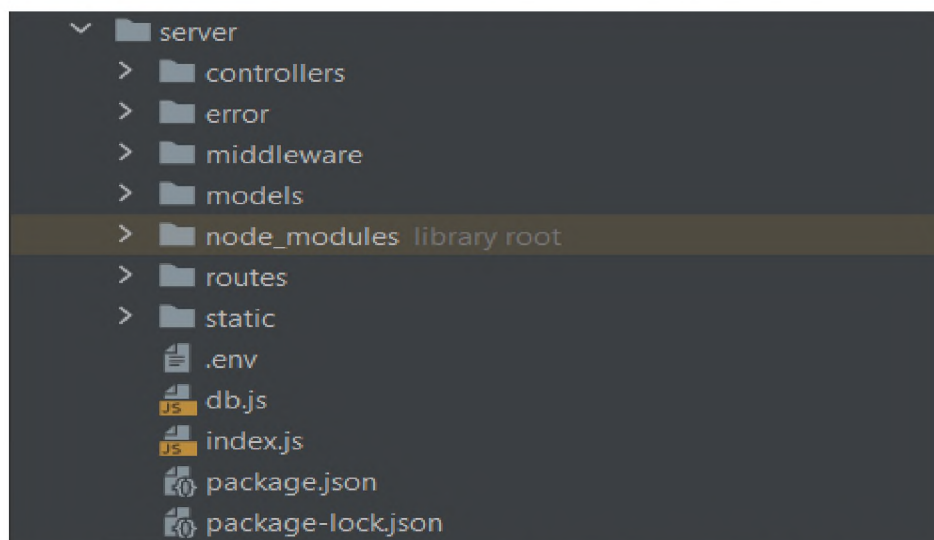


Рис. 2.1. Структура серверної частини проєкту

Загалом вона складається з таких файлів та директорій:

- controllers/ – тут розміщені файли для роботи з об'єктами системи, та реалізація таких функцій як створення, редагування, оновлення та видалення об'єктів системи.

- `error/` – тут розміщений файл для обробки помилок та роботи з ними.
- `middleware/` – тут розміщені файли для реалізації проміжних етапів при обробці запитів на сервері.
- `models/` – тут розміщений файл в якому знаходяться моделі об'єктів системи, які в подальшому використовуються при роботі з базою даних.
- `routes/` – тут розміщені файли для створення та обробки шляхів при виконанні запитів, для кожного з об'єктів.
- `static/` – тут розміщені статичні файли, які використовує сервер.
- `package.json(package-lock.json)` – файли для конфігурації проєкту.
- `db.js` – файл для роботи з базою даних.
- `index.js` – головний файл програми.

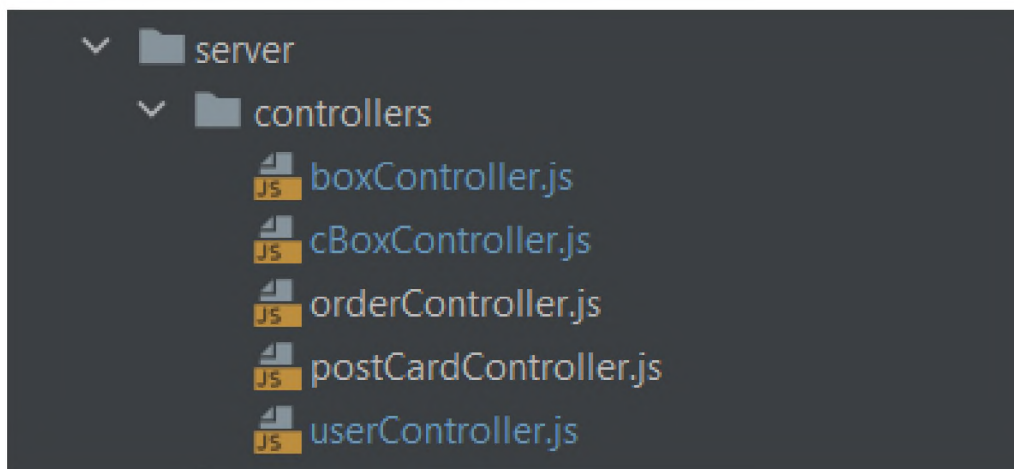
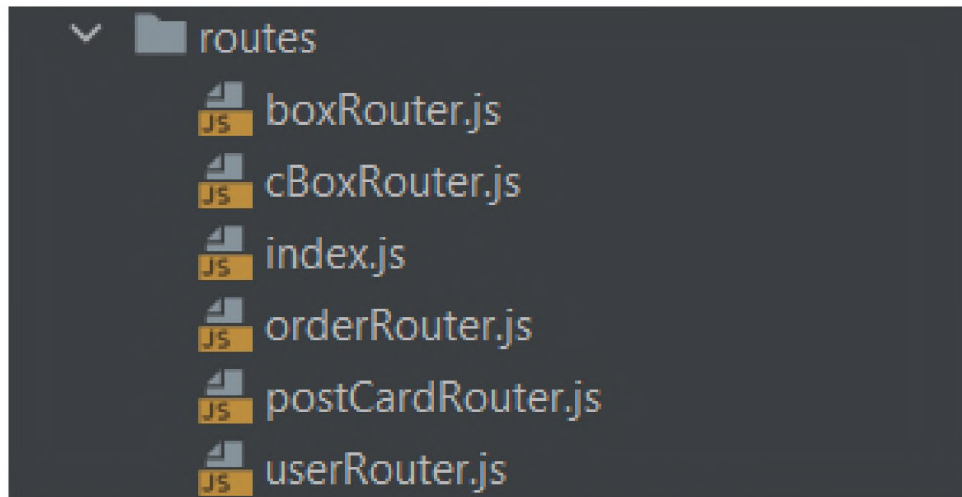


Рис. 2.2. Директорія controllers

У директорії controllers (рис. 2.2) знаходяться файли для роботи з об'єктами системи:

- `boxController.js` – файл який відповідає за роботу об'єкту «box», реалізовує такі функції як створення об'єкту, редагування, видалення, тощо.
- `cBoxController.js` – відповідає за роботу об'єкту «с-box».

- orderController.js – відповідає за роботу об'єкту «order».
- postCardController.js – відповідає за роботу об'єкту «post-card».
- userController.js – відповідає за роботу об'єкту «user», його



створення а також перевірку користувача на авторизацію.

Рис. 2.3. Директорія routes

У директорії controllers (рис. 2.3) знаходяться файли для обробки шляхів у системі:

- boxRouter.js – файл який відповідає за прив'язування конкретних шляхів, до функцій об'єкту «box».
- cBoxRouter.js – відповідає за прив'язування функцій об'єкту до шляхів «с-box».
- orderRouter.js – відповідає за прив'язування функцій до шляхів об'єкту «order».
- postCardRouter.js – відповідає за прив'язування функцій до шляхів об'єкту «post-card».
- userRouter.js – відповідає за прив'язування об'єкту до шляхів «user».
- Index.js – файл для оголошення існуючих шляхів.

Структура серверної частини зображена на рисунку 2.4.

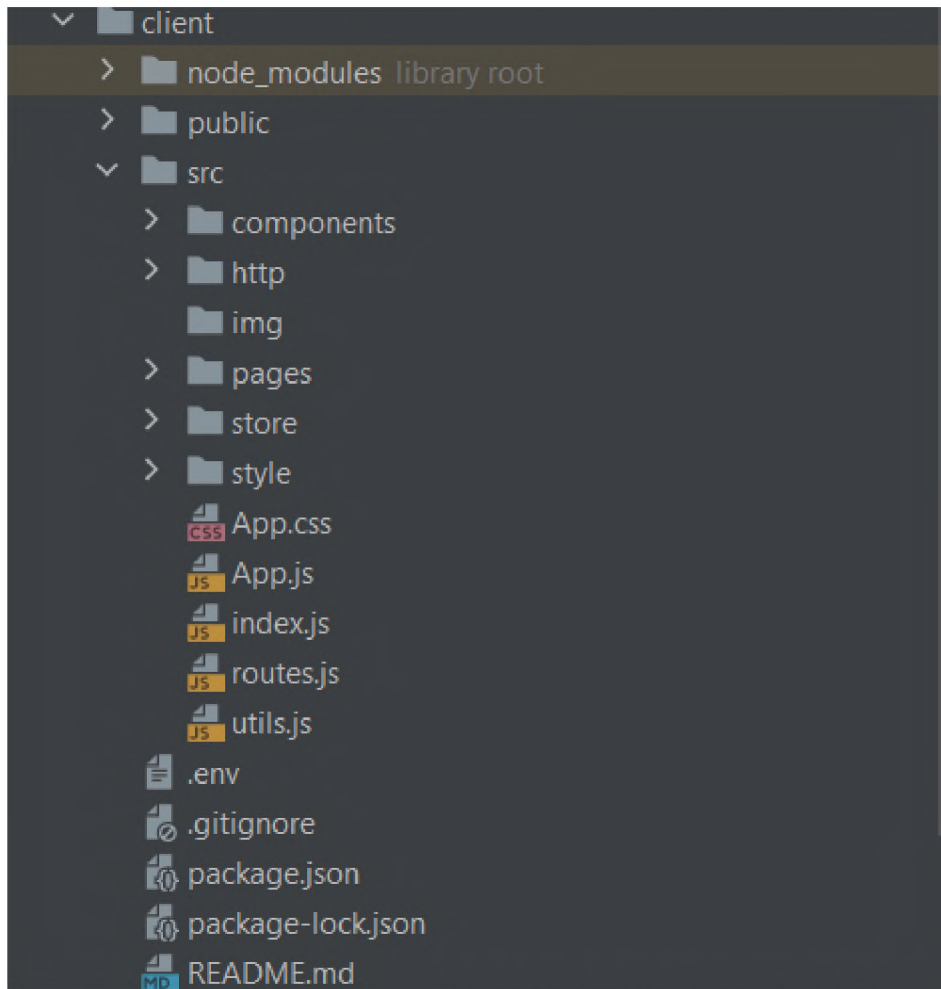


Рис. 2.4. Структура клієнтської частини проєкту

Загалом вона складається з таких файлів та директорій:

- `public/` – тут знаходяться файли для відображення виду іконки сайту.
- `src/` – корінна директорія в якій знаходяться всі файли.
- `package.json(package-lock.json)` – файли для конфігурації проєкту.
- `src/components/` – тут знаходяться файли усіх React компонентів, які створені для подальшого використання.
- `src/http/` – тут знаходяться файли для організації відправки запитів на сервер.
- `src/img/` – тут знаходяться файли картинок проєкту.

- `src/pages` – в цій директорії розташовані файли сторінок, які відображаються на сайті.
- `src/store/` – тут знаходяться файли класів об'єктів системи для зручної роботи.
- `src/style/` – тут знаходяться файли стилів для відображення вигляду веб-додатку.
- `App.css` – основний файл стилів проєкту, який напряду впливає на загальний вигляд.
- `App.js` – файл головної сторінки сайту.
- `routes.js` – файл для організації шляхів, та відображення конкретних компонентів в залежності від отриманого шляху.
- `utils.js` – файл для зберігання констант, для зручного і безпечного використання в подальшому.
- `index.js` – основний файл, де і відбувається організація та робота додатку.

На рисунку 2.5 показаний вміст директорії src/components/.

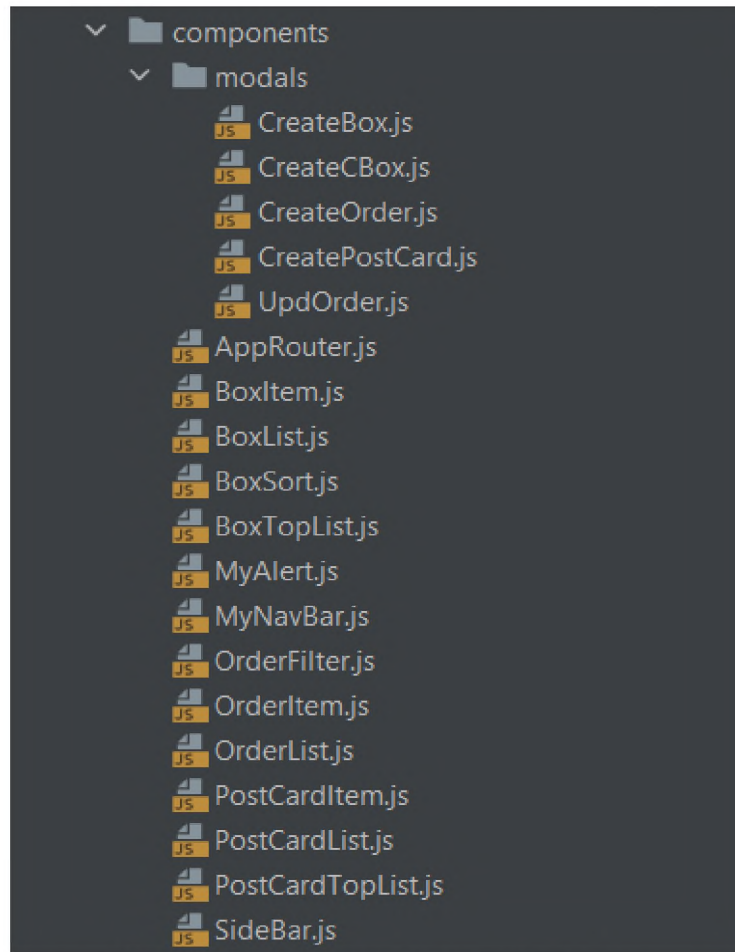


Рис. 2.5. Директорія components

У цій директорії розташовані файли React компонентів які відображаються безпосередньо на сторінці. Також в папці modals розташовані компоненти які виникають як модальні вікна з формами та полями для створення чи редагування різних об'єктів системи, в залежності від вибору користувача.

2.2. Реалізація системи мовою програмування JAVA SCRIPT

Розробка інтерфейсу веб-сайту має високу важливість, оскільки він є основним способом взаємодії користувача з веб-додатком. Нижче наведені деякі основні вимоги до розробки інтерфейсу веб-сайту:

- Наявність інтуїтивно зрозумілих і простих інтерфейсів. Інтерфейс повинен бути дружнім та легко зрозумілим користувачам. Користувач повинен знайти на сайті те, що йому потрібно, за допомогою мінімуму зусиль.

- Дизайн повинен бути чистим та простим. Зайві елементи і надмірні функції можуть призвести до збільшення часу завантаження сторінки та зниження продуктивності.

- Адаптивний дизайн – це ще одна важлива вимога до розробки інтерфейсу вебсайту. Сайт повинен коректно відображатись на будь-якому пристрої та екрані, включаючи мобільні телефони та планшети.

- Наявність логічної структури та інтуїтивного навігаційного меню. Користувач повинен знайти на сайті те, що йому потрібно, за допомогою мінімуму зусиль. Навігація повинна бути логічною та інтуїтивно зрозумілою.

- Наявність достатньої інформації для користувачів. Сайт повинен містити достатньо інформації про компанію, продукти або послуги, контактну інформацію та інше.

- Забезпечення безпеки та конфіденційності даних користувачів. Сайт повинен захищати персональну інформацію користувачів та забезпечувати безпечний процес платежів, якщо він є необхідним.

- Швидкість роботи сайту – це також важлива вимога до розробки інтерфейсу веб сайту. Сайт повинен завантажуватись швидко, оскільки довгий час очікування може знизити інтерес користувача до сайту.

- Сумісність з різними браузерами та операційними системами. Сайт повинен відображатись однаково на різних браузерах та операційних системах, щоб забезпечити максимальну доступність для користувачів.

- Наявність можливості пошуку на сайті. Користувач повинен мати можливість знайти інформацію, яку він шукає, за допомогою пошукової функції на сайті.

- Підтримка клавіатурної навігації та використання доступних технологій для осіб з обмеженими можливостями. Сайт повинен бути доступним для всіх користувачів, включаючи людей з обмеженими можливостями.

- Наявність підтримки мов. Сайт повинен підтримувати різні мови, щоб забезпечити доступність для користувачів з різних країн.
- Розміщення інформації та елементів інтерфейсу в зручних для користувача місцях. Сайт повинен мати логічну та зручну структуру, щоб користувачі могли легко знаходити необхідну інформацію.

Таблиця 2.1.

Основні елементи інтерфейсу проєкту

Назва	Опис
Сторінка «Бокси»	Показує список існуючих боксів
Сторінка «Декор»	Показує список існуючих боксів
Сторінка «Замовлення»	Показує список наявні замовлення інтернет магазину
Сторінка «Звіт»	Показує детальний звіт по існуючим замовленням
Сторінка продукту	Демонструє детальний опис обраного продукту
Сторінка обраного замовлення	Показує деталі обраного замовлення

Основна сторінка «Бокси», зображена на рисунку 2.6. Вона демонструє список існуючих на даний момент боксів солодошів. Сортування можливе як по ціні, так і по назві. Під блоком сортування знаходиться кнопка для створення нового боксу магазину.

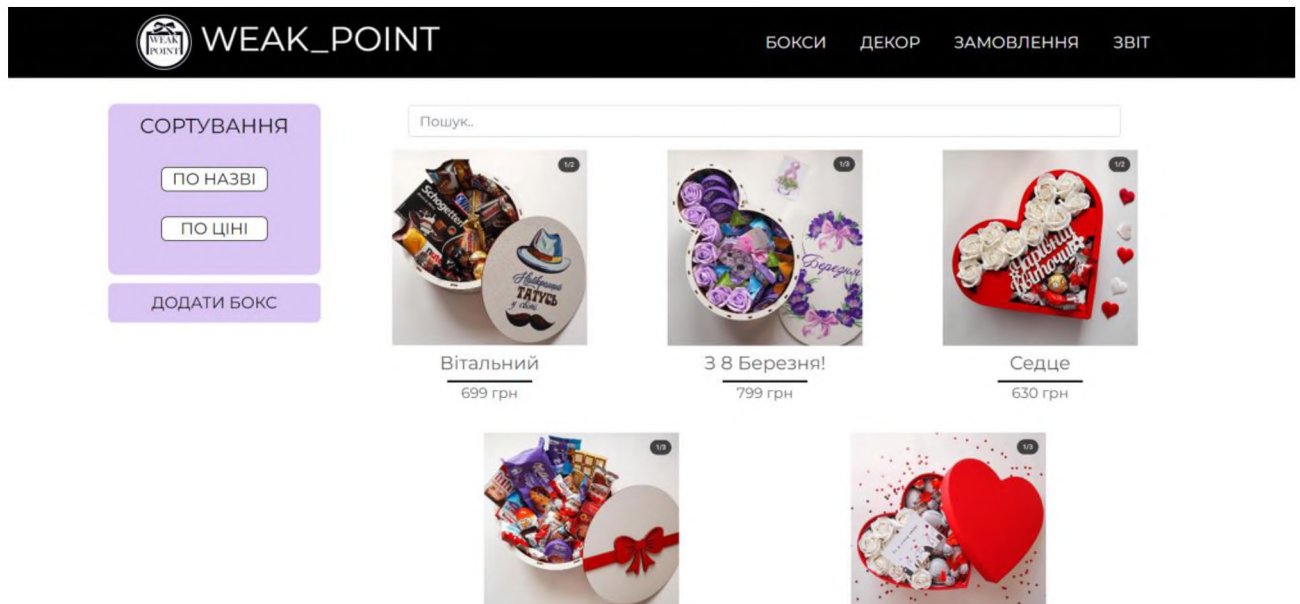


Рис. 2.6. Сторінка «Бокси»

При натисканні на конкретний бокс, переходимо на сторінку продукту. На ній можемо побачити назву боксу, його ціну та зображення. Також продемонстровано його склад. Присутні дві кнопки для зміни опису чи параметрів боксу, або ж його видалення.

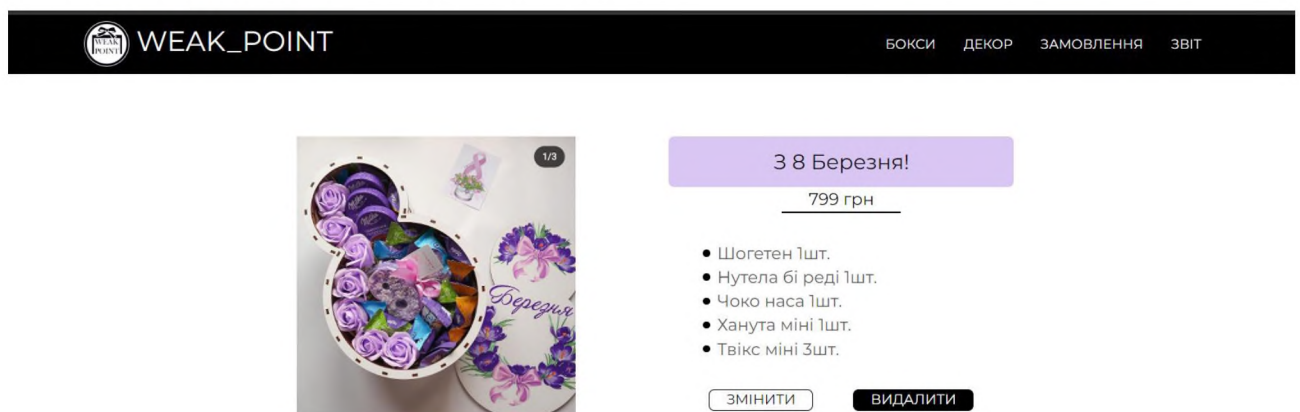


Рис. 2.7. Сторінка продукту

На сторінці «Декори» продемонстровані наявні в магазині декори. Аналогічно з сторінкою «Бокси», тут знаходиться блок для сортування відображення списку, пошуковий рядок.

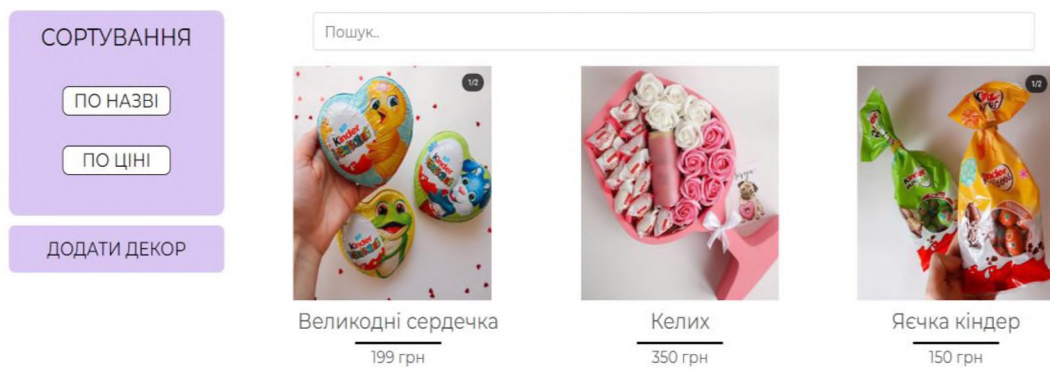


Рис. 2.8. Сторінка «Декори»

При натисканні на будь-який об'єкт декору, переходимо на сторінку з його описом. На сторінці відображені його ціна, назва та зображення. А також можливість його редагування або ж видалення.

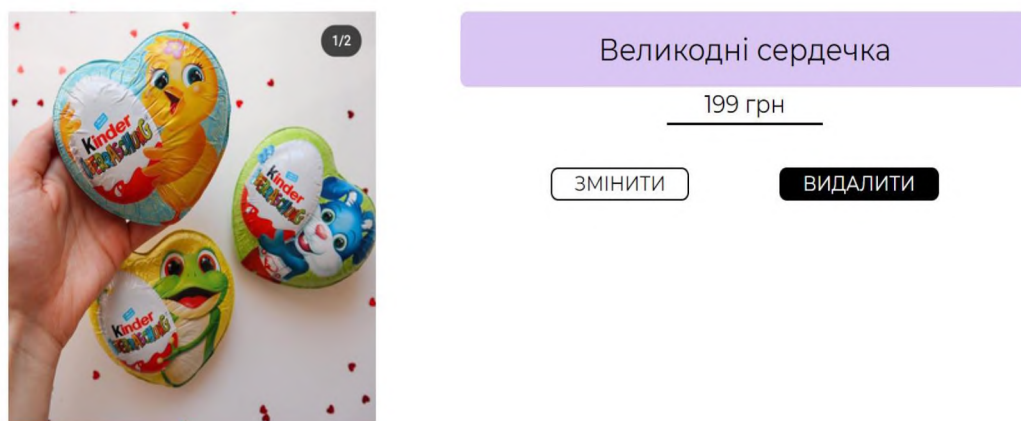


Рис. 2.9. Сторінка конкретного декору

На сторінці «Замовлення» відображається список усіх існуючих замовлень.

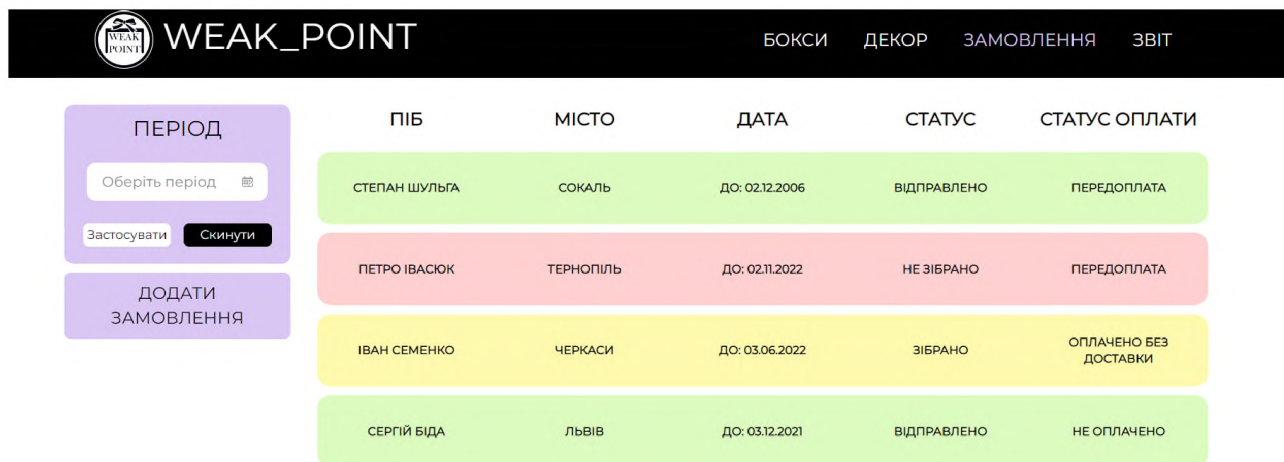


Рис. 2.10. Сторінка «Замовлення»

На сторінці відображені замовлення та їх основні дані, такі як: ПІБ замовника, місто куди прийде замовлення, термін до якого замовлення потрібно відправити, статус замовлення, та статус оплати. При натисканні на параметр «Статус», замовлення будуть відсортовані, спочатку замовлення зі статусом «Не зібрано», потім «Зібрано» і в кінці «Відправлено». В залежності від цього статусу замовлення має конкретний колір фону, для наочності.

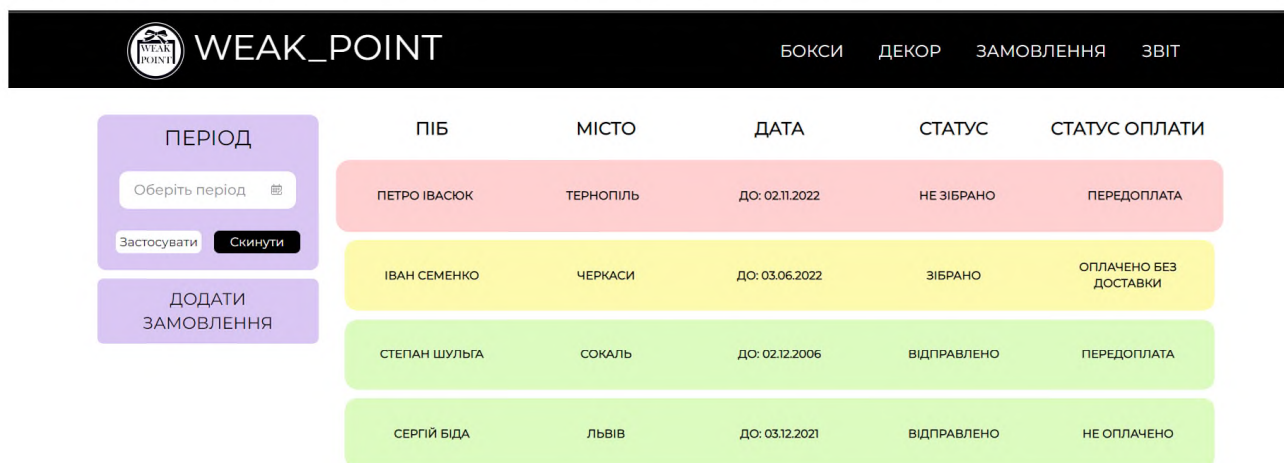


Рис. 2.11. Відсортований список замовлень

Також на сторінці надано можливість відображення замовлень за певний період, за допомогою блоку «Період». При обранні періоду який нас цікавить, будуть відображатись лише ті замовлення які були створені в обраний час.

WEAK_POINT

БОКСИ
ДЕКОР
ЗАМОВЛЕННЯ
ЗВІТ

ПЕРІОД	ПІБ	МІСТО	ДАТА	СТАТУС	СТАТУС ОПЛАТИ
2023-03-30 ~ 2... 2023-03-30 ~ 2023-04-05 05 Apr 2023 05 May 2023 Today Yesterday Last 7 Days OK	ПЕТРО ІВАСЮК	ТЕРНОПІЛЬ	ДО: 02.11.2022	НЕ ЗІБРАНО	ПЕРЕДОПЛАТА
	ЧЕРКАСИ	ДО: 03.06.2022	ЗІБРАНО	ОПЛАЧЕНО БЕЗ ДОСТАВКИ	
	СОКАЛЬ	ДО: 02.12.2006	ВІДПРАВЛЕНО	ПЕРЕДОПЛАТА	
	ЛЬВІВ	ДО: 03.12.2021	ВІДПРАВЛЕНО	НЕ ОПЛАЧЕНО	

Рис. 2.12. Обирання періоду для відображення замовлень

Після вибору періоду, та натисканні кнопки «Застосувати», в списку будуть відображатись відфільтровані замовлення.

WEAK_POINT

БОКСИ
ДЕКОР
ЗАМОВЛЕННЯ
ЗВІТ

ПЕРІОД	ПІБ	МІСТО	ДАТА	СТАТУС	СТАТУС ОПЛАТИ
2023-03-30 ~ 2... Застосувати Скинути ДОДАТИ ЗАМОВЛЕННЯ	ПЕТРО ІВАСЮК	ТЕРНОПІЛЬ	ДО: 02.11.2022	НЕ ЗІБРАНО	ПЕРЕДОПЛАТА
	ІВАН СЕМЕНКО	ЧЕРКАСИ	ДО: 03.06.2022	ЗІБРАНО	ОПЛАЧЕНО БЕЗ ДОСТАВКИ

Рис. 2.13. Відфільтровані замовлення

При необхідності фільтрацію по часу можна скинути, натиснувши на відповідну кнопку.

При натисканні на будь-яке з замовлень ми перейдемо на сторінку цього замовлення.


WEAK_POINT

БОКСИ
ДЕКОР
ЗАМОВЛЕННЯ
ЗВІТ

ПІБ

Петро Івасюк

МІСТО

Тернопіль

ВІДДІЛЕННЯ НП

2

ТЕЛЕФОН

+380945867334

ДАТА

05.04.2023

ДО

02.11.2022

ЦІНА

2149 грн

КОМЕНТАР ДО ДЕКОРУ

КОМЕНТАР

Змінити метод відправлення на пошті

ЗМІНИТИ

ВИДАЛИТИ

БОКСИ:



Солодкий мікс

1 шт.



Червоне серце

1 шт.



Вітальний

1 шт.

ДЕКОР:



Рис. 2.14. Сторінка конкретного замовлення

На даній сторінці зображена детальна інформація про обране замовлення. Відображено такі дані як: ПІБ замовника, місто доставки, відділення Нової Пошти, телефон замовника, дата створення замовлення, дата до якої замовлення має бути відправлене, а також ціна та при наявності коментар до замовлення. Також на сторінці відображено обрані в замовленні бокси та декора.

Під інформаційним блоком, наявні кнопки для редагування замовлення або ж його видалення.

На сторінці «Звіт» реалізована можливість проведення звіту по замовленнях даного магазину.


WEAK_POINT

БОКСИ
ДЕКОР
ЗАМОВЛЕННЯ
ЗВІТ

Звіт

Оберіть період

Показати

Рис. 2.15. Сторінка «Звіт»

2.3. Тестування системи

На сторінці «Бокси» є можливість проведення пошуку за допомогою відповідного пошукового поля.

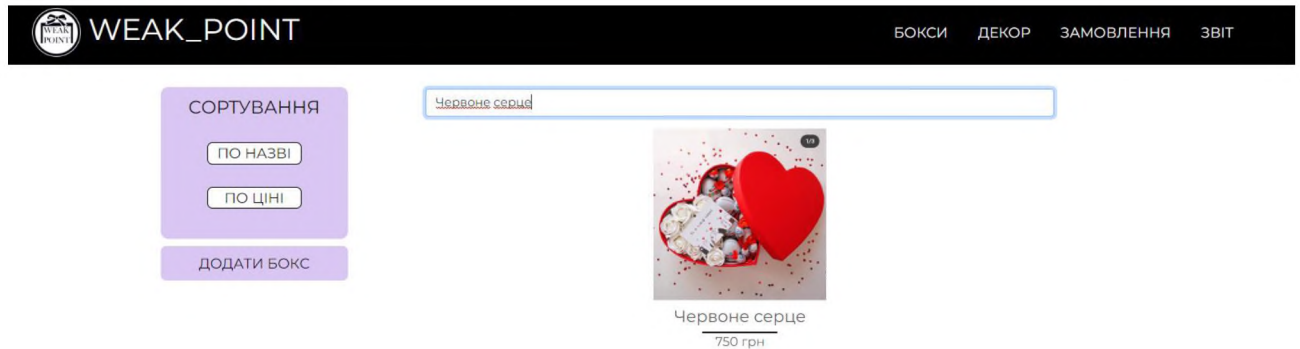


Рис. 2.16. Функція пошуку

При натисканні на кнопку «Додати бокс», відкривається модальне вікно з полями «Назва», «Ціна», а також можливість вибрати потрібне зображення з власного комп'ютеру та додати складові боксу.

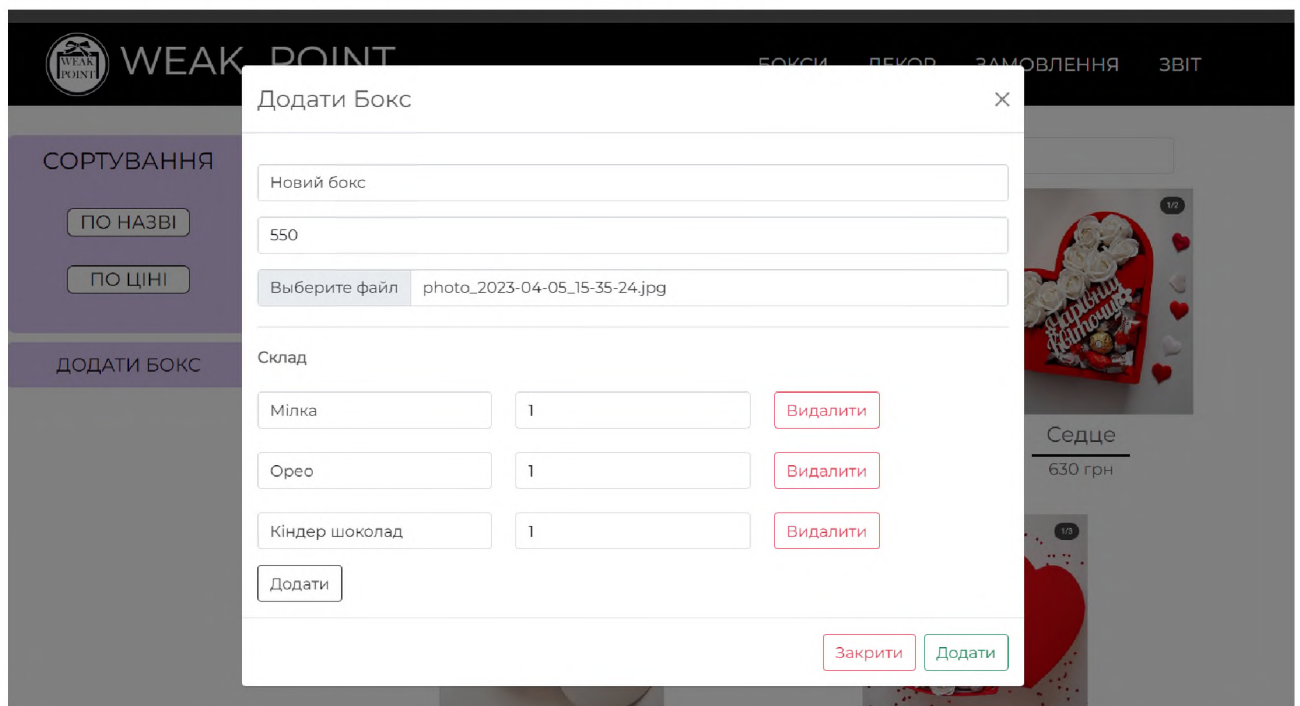


Рис. 2.17. Модальне вікно «Додати бокс»

Після додавання нового боксу він з'являється в списку наявних боксів, що подано на рисунку 2.18.

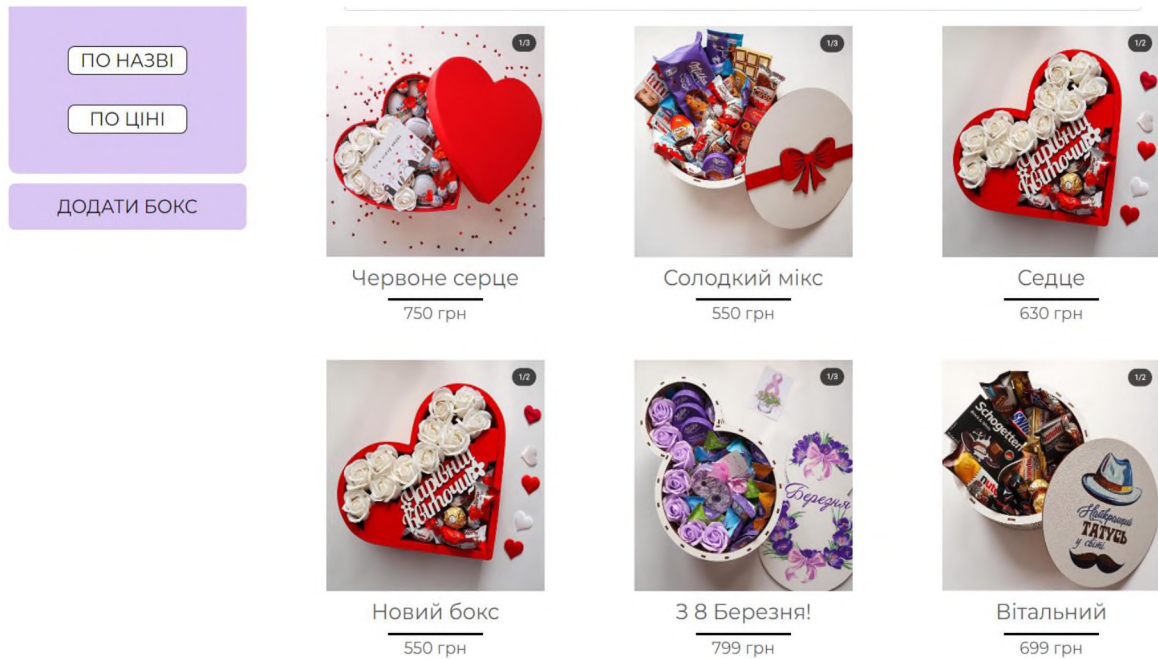


Рис. 2.18. Оновлений список боксів

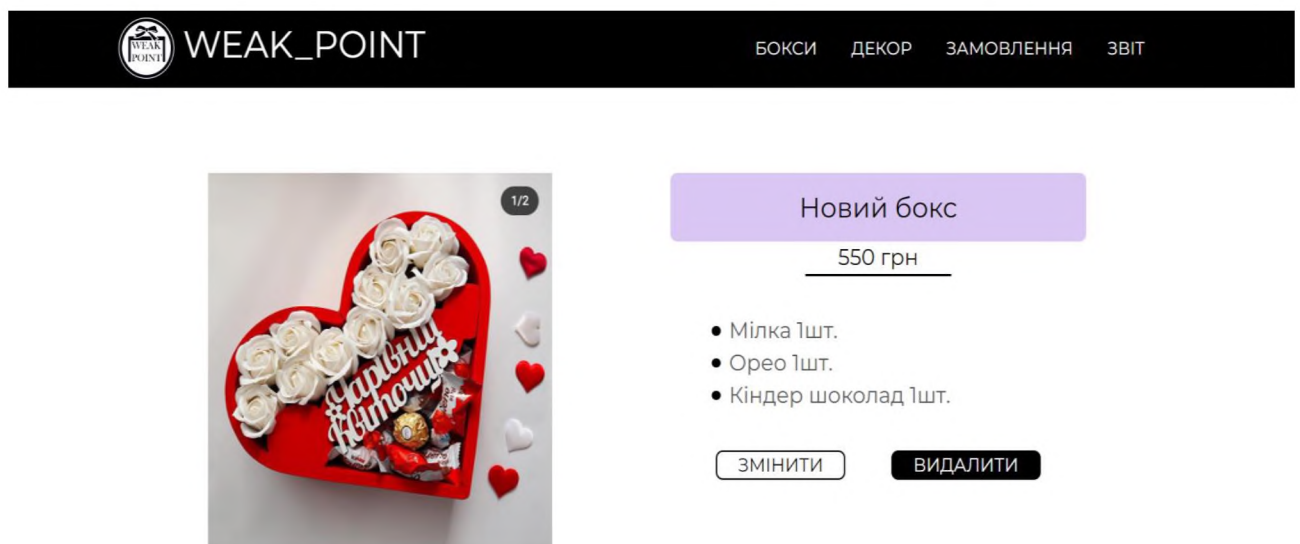


Рис. 2.19. Сторінка нового продукту

Також на сторінці «Декори» є можливість додати новий декор, використавши для цього відповідну кнопку, та заповнивши необхідні поля. На відміну від додавання боксу, тут відсутня можливість додавання його складових, та як передбачається що об'єкт декору йде завжди в одному екземплярі.

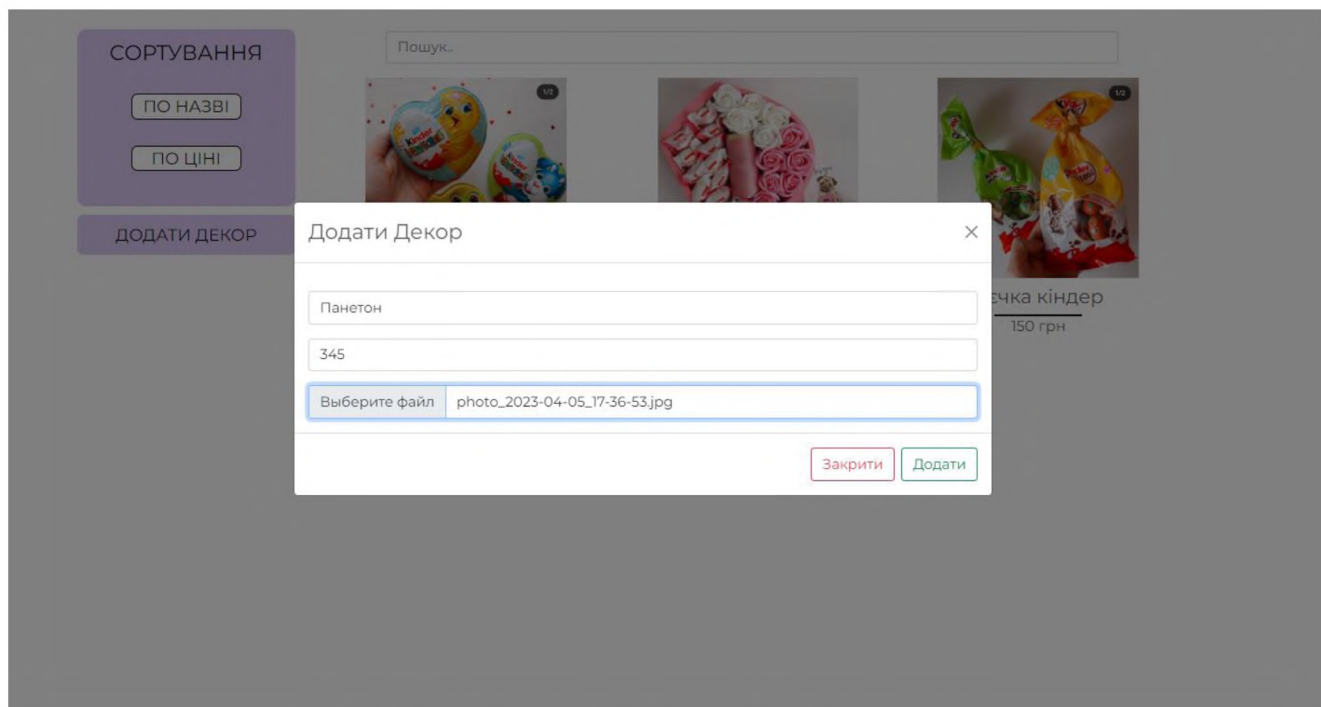


Рис. 2.20. Модальне вікно «Додати декор»

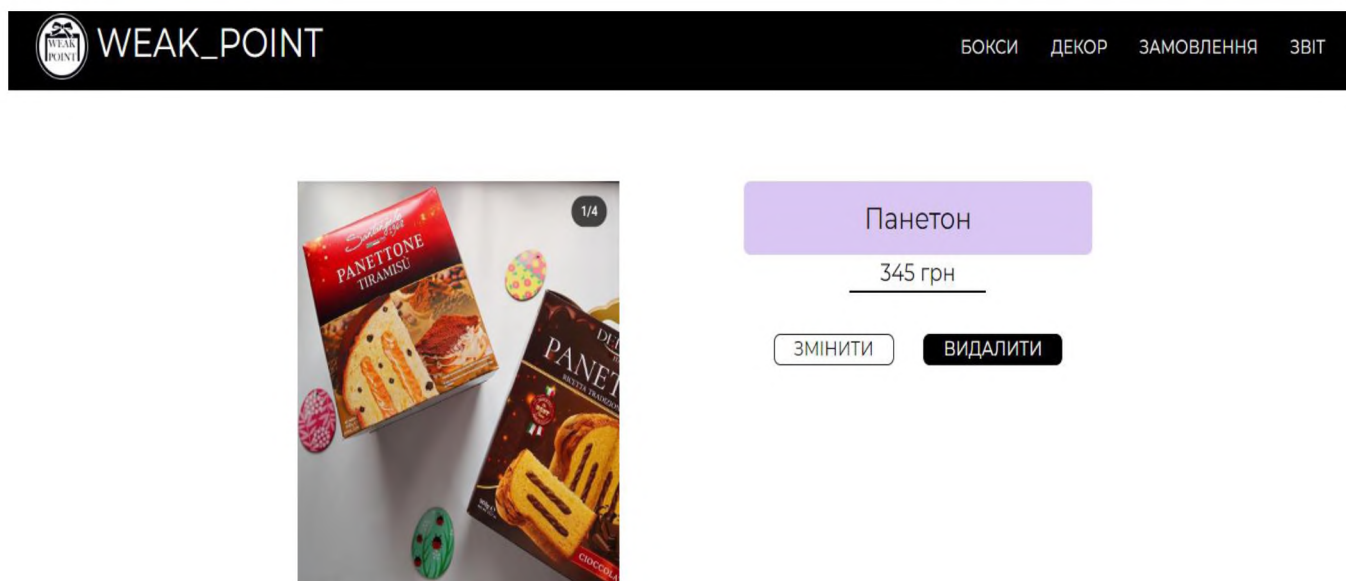


Рис. 2.21. Сторінка нового декору

При створенні нового замовлення відкривається модальне вікно, зображене на рисунку 2.22.

The screenshot shows a web application interface with a modal window titled "Додати замовлення" (Add Order). The background is a dark grey sidebar with the "WEAK_POINT" logo and navigation options like "ПЕРІОД", "ДОДАТИ ЗАМОВЛЕННЯ", "СТАТУС ОПЛАТИ", "ПЕРЕДОПЛАТА", "ОПЛАЧЕНО БЕЗ ДОСТАВКИ", and "НЕОПЛАЧЕНО".

The modal window contains the following elements:

- Buttons: "Додати Бокс" (Add Box) and "Додати Декор" (Add Decor).
- Text: "* - обов'язкові поля" (Required fields).
- Form fields:
 - *Введіть ім'я клієнта (Required)
 - Введіть телефон клієнта
 - Введіть дату відправки
 - Введіть місто доставки
 - Введіть відділення НП
 - Коментар
 - *Введіть ціну (Required)
- Buttons: "Розрахувати" (Calculate).
- Dropdown menus: "Оберіть стан замовлення" (Select order status) and "Оберіть стан оплати" (Select payment status).
- Footer buttons: "Закрити" (Close) and "Додати" (Add).

Рис. 2.22. Модальне вікно додавання замовлення

Для створення замовлення, потрібно заповнити необхідні поля. Обов'язкові поля відзначені символом «*» Також можна обрати існуючі бокси чи декори та додати їх до замовлення. При додаванні складових замовлення, є можливість автоматичного розрахування ціни усього замовлення після натискання відповідної кнопки. В кінці потрібно обрати статус замовлення та статус оплати.

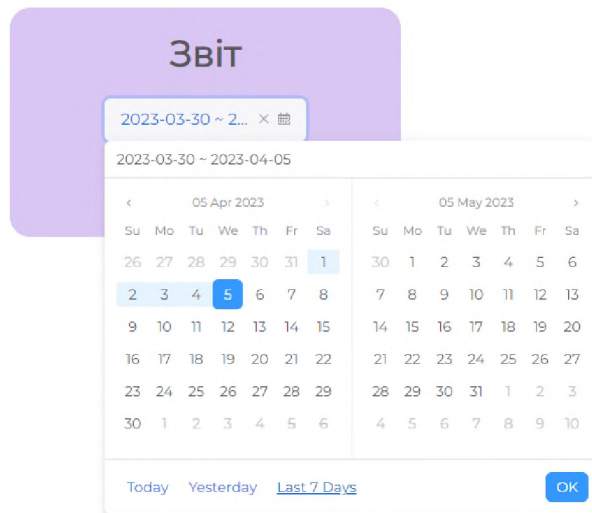


Рис. 2.25. Блок для вибору часового періоду

Після вибору періоду часу який цікавить користувача, та натисканні на кнопку «Показати», система проводить аналіз по замовленнях, які відповідають обраним часовим рамкам.

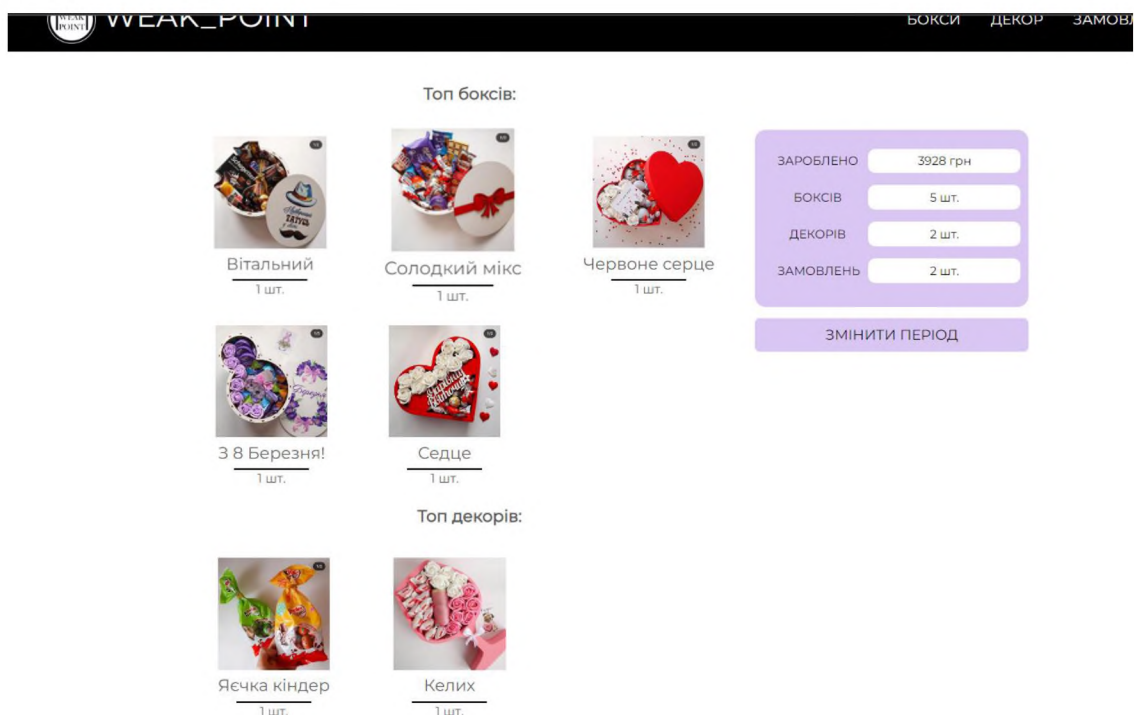


Рис. 2.26 Демонстрація звіту за обраний період

На сторінці відображені секції такі як:

- Топ боксів – відображення усіх боксів, які були додані до замовлень за обраний період, та їхня кількість.

- Топ декорів – відображення усіх декорів, які були додані до замовлень за обраний період, та їхня кількість.

- Блок із загальною інформацією – блок в якому показано інформацію по обраних замовлень, кількість всього зароблених коштів, кількість проданих боксів, кількість проданих декорів, уся кількість замовлень.

Також на сторінці є можливість змінити часовий період, після натискання відповідної кнопки.

РОЗДІЛ 3. ПРОПОЗИЦІЇ ЗА РЕЗУЛЬТАТАМИ ДОСЛІДЖЕННЯ

Веб-орієнтовані інформаційні системи стають все більш популярними серед компаній, які хочуть вдосконалити свою роботу та оптимізувати бізнес-процеси. Розробка веб-орієнтованої інформаційної системи для ведення обліку інтернет-магазину мовою програмування Java Script є однією з можливих рішень для покращення роботи інтернет-магазину та залучення більше клієнтів.

В результаті проведеного дослідження були отримані різноманітні результати, що можуть бути використані для подальшого розвитку інформаційної системи. В цьому розділі розглядається декілька можливих пропозицій, що дозволять вдосконалити роботу інтернет-магазину та залучити більше клієнтів за допомогою розробки нових функцій і модулів.

Ось кілька можливих пропозицій на основі дослідження проєкту розробки веб-орієнтованої інформаційної системи для ведення обліку інтернет-магазину мовою програмування Java Script:

1. Розширення функціональності системи: пропонується розширити функціональні можливості системи, додавши нові функції, такі як підтримка більшого рівня забезпечення безпеки, реалізація розширеного аналізу продажів, підтримка інтеграції з різними сервісами платежів та інші.

2. Оптимізація швидкодії системи: пропонується провести дослідження і оптимізацію швидкодії системи, щоб зменшити час відгуку та час завантаження сторінок, підвищити продуктивність системи та покращити користувацький досвід.

3. Покращення інтерфейсу користувача: пропонується покращити інтерфейс користувача системи, роблячи його більш інтуїтивно зрозумілим, дружелюбним та зручним у використанні для користувачів.

4. Розширення мобільної платформи: пропонується розширити функціональні можливості системи, щоб підтримувати платформи мобільних пристроїв, додавши функції, які дозволяють користувачам здійснювати покупки, переглядати свій історію замовлень та отримувати сповіщення про статус замовлень на свої мобільні пристрої.

5. Підвищення рівня безпеки: пропонується підвищити рівень безпеки системи, використовуючи різні методи інтегрованої захисту, такі як шифрування, автентифікація користувачів, аудит безпеки та моніторинг вразливостей системи, щоб захистити дані користувачів та забезпечити безпечну операційну діяльність інтернет-магазину.

6. Оптимізація використання баз даних: пропонується провести дослідження та оптимізувати використання баз даних, щоб забезпечити швидкий доступ до даних та зменшити час відгуку системи. Оптимізація баз даних допоможе підвищити продуктивність системи та зменшити витрати на її обслуговування.

7. Підвищення масштабованості системи: пропонується розробити систему з урахуванням масштабованості, щоб забезпечити підтримку зростання інтернет-магазину та збільшення обсягів операцій. Підвищення масштабованості допоможе забезпечити стабільну та продуктивну роботу системи, незалежно від обсягу операцій та користувачів.

8. Забезпечення надійності та безперебійності: пропонується забезпечити надійність та безперебійність роботи системи, використовуючи різноманітні методи, такі як відновлення після аварій, дублювання системних компонентів та забезпечення автоматичного моніторингу. Це допоможе зменшити вплив можливих помилок та аварій на користувачів інтернет-магазину, а також забезпечить безперебійну роботу системи.

9. Розробка мобільної версії системи: пропонується розробити мобільну версію системи для забезпечення доступу до інформації про інтернет-магазин зі смартфонів та планшетів. Це допоможе збільшити зручність використання системи та залучити більше користувачів, що може позитивно вплинути на розвиток інтернет-магазину.

10. Інтеграція з іншими системами: пропонується розробити можливість інтеграції системи з іншими відомими системами, наприклад, з системами управління складом, бухгалтерськими програмами, системами електронних платежів та іншими. Це допоможе покращити ефективність управління інтернет-магазином та збільшити автоматизацію бізнес-процесів.

11. Розробка системи звітності: пропонується розробити систему звітності, яка забезпечить можливість генерувати звіти про роботу інтернет-магазину за різними періодами часу та з різних аспектів, наприклад, звіти про продажі, звіти про запаси товарів, звіти про роботу персоналу та інші. Це допоможе підвищити ефективність управління інтернет-магазином та забезпечить можливість аналізу та покращення його роботи.

12. Розробка системи рекомендацій: пропонується розробити систему рекомендацій для користувачів інтернет-магазину, яка забезпечить можливість рекомендаційних пропозицій товарів на основі історії покупок та інших параметрів користувача. Це допоможе покращити ефективність продажу товарів та залучити більше користувачів.

Ці пропозиції можуть бути використані для подальшого розвитку та покращення веб-орієнтованої інформаційної системи для ведення обліку інтернет-магазину мовою програмування Java Script.

ВИСНОВКИ

Під час дослідження була проведена розробка веб-орієнтованої інформаційної системи для ведення обліку інтернет-магазину мовою програмування Java Script. Результати дослідження свідчать про те, що розроблена система може бути ефективним інструментом для автоматизації бізнес-процесів інтернет-магазину, що в свою чергу може привести до збільшення продажів та покращення відносин з клієнтами.

Загальні результати дослідження дозволяють зробити висновок про те, що розробка веб-орієнтованої інформаційної системи для ведення обліку інтернет-магазину мовою програмування Java Script є актуальним і перспективним напрямом вдосконалення роботи бізнесу. Результати дослідження можуть бути використані для подальшого розвитку інформаційної системи та підвищення її ефективності. Під час розробки інформаційної системи було враховано основні вимоги до такої системи, а саме: забезпечення ведення детального обліку товарів та замовлень, автоматизація процесів підтримки клієнтів, забезпечення доступу до системи з будь-якого пристрою з підтримкою веб-браузера. У результаті було розроблено зручний та функціональний інтерфейс користувача, який дозволяє здійснювати всі необхідні операції з ведення обліку товарів та замовлень.

Окрім цього, було проведено тестування розробленої інформаційної системи та виявлено певні недоліки, які було усунуто. Зокрема, було виявлено проблеми зі швидкістю роботи системи при великому обсязі даних. Цю проблему вдалося вирішити шляхом оптимізації деяких алгоритмів та використання додаткових інструментів.

Отже, розробка веб-орієнтованої інформаційної системи для ведення обліку інтернет-магазину мовою програмування Java Script є актуальним та перспективним напрямом вдосконалення роботи бізнесу. Результати дослідження можуть бути використані для подальшого розвитку інформаційної системи та підвищення її ефективності, а також для подальшого дослідження в галузі розробки програмного забезпечення.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Duckett Jon. JavaScript and jQuery: Interactive Front-End Web Development. Wiley, 2014. 640 p.
2. Eloquent JavaScript: A Modern Introduction to Programming. O'Reilly; 2nd edition, 2014. 472 p.
3. Flanagan David. JavaScript: The Definitive Guide: Activate Your Web Pages. O'Reilly Media, 2011. 1096 p.
4. Mozilla Developer Network. JavaScript. Mozilla Developer Network, 2021. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
5. Resig John. Pro JavaScript Techniques: Second Edition. Apress, 2015. 352 p.
6. The Lean Product Playbook. Wiley, 2015. 336 p.
7. W3Schools. JavaScript Tutorial. W3Schools, 2021. URL: <https://www.w3schools.com/js/default.asp>.
8. Алекс Бендерський. React. Найкращі практики / пер. з англ. А. Шамрило. Київ: КМ Академія, 2020. 360 с.
9. Гергель В. Основи HTML та CSS: навч. посіб. Львів: Львівська політехніка, 2017. 336 с.
10. Гончаров О. Веб-технології та веб-дизайн: підруч. Київ: Центр учбової літератури, 2019. 288 с.
11. Ендрю Майклс, Пол Дакетт. JavaScript. Підручник з прикладами / пер. з англ. В. Блинова. Київ: Дім інформаційної книги, 2019. 480 с.
12. Колесник М.В. Веб-дизайн. Курс лекцій: навч. посіб. Київ: Видавничий дім «Сам», 2016. 232 с.
13. Корнієнко В. Python для дітей: навч. посіб. / пер. з англ. Харків: Клуб сімейного дозвілля, 2019. 320 с.
14. Лисенко Т.О. Методи проєктування веб-сайтів: навч. посіб. Київ: Сам, 2018. 232 с.
15. Мельник А.В. Створення веб-сайту: практичний підхід: навч. посіб. Київ: Центр учбової літератури, 2018. 320 с.

16. Ніколас Закас. JavaScript. Нові можливості для професіоналів / пер. з англ. В. Синицького. Київ: Клуб сімейного дозвілля, 2016. 648 с.
17. Стойан Стефанов. JavaScript. Патерни. Програмування / пер. з англ. М. Кравцова. Київ: Клуб сімейного дозвілля, 2017. 320 с.
18. Столяренко Ю.В. Основи програмування на мові Python: навч. посіб. / Київ: КМ Академія, 2020. 364 с.
19. Томас Хант, Марк Харолд. Node.js. Повне керівництво / пер. з англ. В. Синицького. Київ: Дім інформаційної книги, 2019. 504 с.
20. Фредерік Філіпс. Angular. Повне керівництво / пер. з англ. В. Синицького. Київ: Дім інформаційної книги, 2019. 624 с.
21. Шон Лан. Vue.js 2. Розробка професійних веб-додатків / пер. з англ. О. Зінкевича. Київ: Буква, 2019. 304 с.

ДОДАТКИ

client/src/App.js

```
import React, {useContext, useEffect, useState} from "react";
import {BrowserRouter} from "react-router-dom";
import AppRouter from "../components/AppRouter";
import MyNavBar from "../components/MyNavBar";
import {observer} from "mobx-react-lite";
import {Context} from "../index";
import {check} from "../http/userApi";
import {Spinner} from "react-bootstrap";
import styles from "../App.css"
```

```
const App = observer(() => {
  const {user} = useContext(Context)
  const [loading, setLoading] = useState(true)

  useEffect(()=>{
    check().then(data=>{
      user.setIsAuth(true)
      user.setUser(user)
    }).finally(()=> setLoading(false))
  },[])
  if(loading){
    return <Spinner animation={"grow"}/>
  }

  return (
    <BrowserRouter >
      <MyNavBar/>
      <AppRouter/>
    </BrowserRouter>
  )
})
```

```
);  
})
```

```
export default App;
```

client/src/index.js

```
import React, {createContext} from 'react';  
import ReactDOM from 'react-dom';  
import App from './App';  
import UserStore from './store/UserStore';  
import BoxStore from './store/BoxStore';  
import PostCardStore from './store/PostCardStore';  
import OrderStore from './store/OrderStore';
```

```
export const Context = createContext(null)
```

```
ReactDOM.render(  
  <Context.Provider value={{  
    user: new UserStore(),  
    box: new BoxStore(),  
    postCard: new PostCardStore(),  
    order: new OrderStore()  
  }}>  
    <App />  
  </Context.Provider>,  
  document.getElementById('root')  
);
```

client/src/routes.js

```
import Main from './pages/Main';  
import {
```

```

    BOXPAGE_ROUTE, DECOR_ROUTE,
    LOGIN_ROUTE,
    MAIN_ROUTE,
    ORDER_ROUTE,
    POSTCARD_ROUTE,
    REPORT_ROUTE
} from "./utils";
import BoxPage from "./pages/BoxPage";
import Order from "./pages/Order";
import Auth from "./pages/Auth";
import OrderPage from "./pages/OrderPage";
import ReportOrder from "./pages/ReportOrder";
import PostCardPage from "./pages/PostCardPage";
import PostCard from "./pages/PostCard";

export const authRoutes = [
  {
    path:MAIN_ROUTE,
    Component: <Main/>
  },
  {
    path:BOXPAGE_ROUTE +('/:id',
    Component: <BoxPage/>
  },
  {
    path:POSTCARD_ROUTE +('/:id',
    Component: <PostCardPage/>
  },
  {
    path:DECOR_ROUTE,
    Component: <PostCard/>
  }
];

```

```

    },
    {
      path:ORDER_ROUTE,
      Component: <Order/>
    },
    {
      path:ORDER_ROUTE + '/:id',
      Component: <OrderPage/>
    },
    {
      path:REPORT_ROUTE,
      Component: <ReportOrder/>
    }
  ]
export const publicRoutes = [
  {
    path:LOGIN_ROUTE,
    Component: <Auth/>
  }
]

```

server/controller/userController.js

```

const {User} = require('../models/models')
const ApiError = require('../error/ApiError')
const jwt = require('jsonwebtoken')
const bcrypt = require('bcrypt')

class userController {
  async create(req, res,next) {
    try {
      const {email, password} = req.body
      const candidate = await User.findOne({where: {email}})

```

```

    if(candidate){
        return next(ApiError.badRequest("Користувач з таким Email уже існує"))
    }
    const hashPassword = await bcrypt.hash(password,5)

    const user = await User.create({email, password:hashPassword})
    const token = jwt.sign(
        {
            id:user.id,email:user.email,},
        process.env.SECRET_KEY,
        {expiresIn: '24h'}
    )
    return res.json({token})
} catch (e){
    next(ApiError.badRequest(e.message))
}
}
async login(req, res,next) {
    const {email, password} = req.body
    const user = await User.findOne({where:{email}})
    if(!user){
        return next(ApiError.badRequest("Користувача з таким Email не існує"))
    }
    // let comparePassword = bcrypt.compareSync(password,user.password)
    // console.log(user.password)
    // if(!comparePassword){
    //     return next(ApiError.badRequest("Не правильний пароль"))
    // }
    const token = jwt.sign(
        {
            id:user.id,email:user.email,

```

```

    },
    process.env.SECRET_KEY,
    {expiresIn: '1h'}
  )
  return res.json({token})
}
async check(req, res,next) {
  const token = jwt.sign(
    {
      id:req.user.id,email:req.user.email,
    },
    process.env.SECRET_KEY,
    {expiresIn: '1h'}
  )
  return res.json({token})
}
}

module.exports = new userController()
server/middleware/authMiddleware.js
const jwt = require('jsonwebtoken')

module.exports = function (req,res,next){

  if(req.method === "OPTIONS"){
    next()
  }
  try {
    const token = req.headers.authorization.split(' ')[1]

    if(!token){
      res.status(401).json({message:"Не авторизованный"})
    }
  }
}

```

```
    }  
    const decoded = jwt.verify(token, process.env.SECRET_KEY)  
    req.user = decoded  
    next()  
  } catch (e) {  
    res.status(401).json({ message: "Не авторизован" })  
  }  
}
```