

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач відділення

комп'ютерних технологій

Наталія Стефурак / _____/

підпис

«__»_____2023 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проєкту

освітньо-кваліфікаційного рівня «молодший спеціаліст»

зі спеціальності 122 «Комп'ютерні науки»

на тему: «Веб-сайт для продуктового магазину “Левас”»

Студент групи КН-41

Дживра Н.Б.

(підпис)

Керівник проєкту

Чубей О.О.

(підпис)

Консультанти:

з техніко-економічного
обґрунтування

Меленчук Л.І.

(підпис)

нормоконтролер

Гавришків Н.Г.

(підпис)

Тернопіль – 2023

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ЗАТВЕРДЖУЮ

Завідувач відділення

комп'ютерних технологій

Наталія Стефурак / _____/

підпис

«__» _____ 2022 р.

ЗАВДАННЯ

на дипломне проєктування
на здобуття освітньо-кваліфікаційного рівня «молодший спеціаліст»

студенту _____
(прізвище, ім'я та по-батькові студента)

1. Тема проєкту _____

затверджена наказом по коледжу від “16” грудня _____ 2022 р., № 1193-н

2. Термін здачі студентом завершеного проєкту “__” _____ 2023 р.

3. Вихідні дані до проєкту _____

4. Перелік питань, які повинні бути розроблені в проєкті:

а) основна частина _____

б) техніко-економічне обґрунтування _____

5. Перелік графічного матеріал _____

6. Консультанти проекту: _____

Розділ	Консультанти	Підпис, дата	
		Завдання видано	Завдання прийнято
З техніко-економічного обґрунтування	<div></div> <div>(вчена ступень, звання П.І.Б. консультанта)</div>		

КАЛЕНДАРНИЙ ПЛАН дипломного проектування

№ п/п	Найменування етапу	Терміни	
		початку	завершення
1.	Вибір теми, ознайомлення з вимогами до дипломного проектування.	28.11.2022 р.	12.12.2022 р.
2.	Огляд типових рішень та написання відповідного розділу ПЗ	13.12.2022 р.	25.12.2022 р.
3.	Дослідження технологій реалізації та написання відповідного розділу ПЗ	23.01.2023 р.	13.02.2023 р.
4.	Розробка функціональних вимог до проекту та робота над структурою програмного продукту. Написання відповідного розділу ПЗ	14.02.2023 р.	24.02.2023 р.
5.	Встановлення на налаштування середовища реалізації	1.03.2023 р.	23.03.2023 р.
6.	Проектування програмного засобу (функціоналу, інтерфейсу, бази даних продукту)	27.03.2023 р.	10.04.2023 р.
7.	Реалізація та налаштування програмного засобу	11.04.2023 р.	05.05.2023 р.
8.	Доопрацювання модулів	08.05.2023 р.	29.05.2023 р.
9.	Тестування на налагодження продукту	30.05.2023 р.	05.06.2023 р.
10.	Опрацювання економічного розділу	06.06.2023 р.	12.06.2023 р.
11.	Робота над оформленням пояснювальної записки	12.06.2023 р.	17.06.2023 р.
12.	Попередній захист дипломного проекту, доопрацювання	15.06.2023 р.	
13.	Підготовка до захисту дипломного проекту	16.06.2023 р.	25.06.2023 р.
14.	Захист дипломного проекту	26.06.2023 р.	27.06.2023 р.

7. Дата видачі завдання “___” _____ 2022 р.

Керівник _____/

Завдання прийняв до виконання _____/

Реферат

Дипломний проєкт. Тема: «Веб-сайт для продуктового магазину “Левас”» містить сторінок – 69, рисунків - 51, таблиць - 4, джерел - 14, додатків – 1.

Метою даного дипломного проєкту є створення веб-сайту для продуктового магазину “Левас”.

Об’єктом дослідження для подальшої розробки є інтернет магазин Gastro Freshbaza та Delikatto.

Головним завданням є побудова веб-сайту для продуктового магазину “Левас”.

Результатом дипломного проєктування було розроблено сайт для продуктового магазину “Левас”. Цей сайт орієнтований для широкого застосування в мережі Інтернет. Завдяки йому, користувачі зможуть переглядати товари, дізнатись про акції та робити покупки через мережу інтернет.

Для розробки інтерфейсу веб-сайту було використано такі технології як React та Scss. Основною особливістю розробленого сайту є адаптивність, тобто можливість відображення на мобільних пристроях та планшетах. Для розробки клієнтської частини використовується бібліотека React.

React — це JavaScript-бібліотека, з використанням якої розробляються інтерфейси. В цій бібліотеці інтерфейси пишуться за допомогою такого інструмента, як компоненти, які являють з себе окремі блоки коду.

Для зберігання даних також використовується СУБД MongoDB. MongoDB – це документно-орієнтована система управління базами. Дані в ній зберігаються у вигляді JSON, тобто об’єктів мови JavaScript.

HTML+CSS, JAVASCRIPT, АНАЛІЗУВАННЯ, СИСТЕМА, ІНТЕРНЕТ, МАГАЗИН, ПРОДУКТОВИЙ «ЛЕВАС»

Abstract

Diploma project. Topic: "Web site for the grocery store "Levas"" contains 70 pages, 51 figures, 4 tables, 14 sources, 1 appendices.

The goal of this diploma project is to create a website for the Levas grocery store.

The object of research for further development is the online store Gastro Freshbaza and Delikatto.

The main task is to build a website for the Levas grocery store.

As a result of the diploma project, a website was developed for the Levas grocery store. This site is intended for wide use on the Internet. Thanks to it, users will be able to view products, find out about promotions and make purchases via the Internet.

Technologies such as React and Scss were used to develop the website interface. The main feature of the developed site is adaptability, that is, the ability to display on mobile devices and tablets. The React library is used to develop the client part.

React is a JavaScript library that is used to develop interfaces. In this library, interfaces are written using such a tool as components, which are separate blocks of code.

MongoDB DBMS is also used for data storage.

MongoDB is a document-oriented database management system.

Data in it is stored in the form of JSON, that is, objects of the JavaScript language.

HTML+CSS, JAVASSRIPT, ANALYSIS, SYSTEM, INTERNET, SHOP, PRODUCT "LEVAS"

ЗМІСТ

Вступ.....	8
1 Аналіз існуючих рішень і постановка завдань.....	9
1.1 Аналітичний огляд існуючих рішень	9
1.2 Система обґрунтування доцільності.....	13
2 Проєктування програмної системи	16
2.1 Огляд основних інструментів для розробки веб сайтів.....	16
2.2 Постановка задачі на розробку програмного забезпечення.....	25
2.3.....	Оп
ис та обґрунтування вибору методу організації вхідних та вихідних даних	25
2.4 Структура та огляд сайту.....	30
2.5 Розробка структури бази даних сайту	34
2.6 Програмування сайту	37
3 Реалізація та тестування програмної системи.....	41
3.1 Найменування та область застосування.....	41
3.2 Вимоги до програмного забезпечення та документації.....	42
3.3 Техніко-економічні показники.....	42
3.4 Стадії та етапи розробки.....	43
3.5 Тестування сайту	43
4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ.....	57
4.1.....	Виз
начення стадій технологічного процесу та загальної тривалості проведення робіт	57
4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи	58
4.3 Розрахунок витрат на електроенергію.....	61

					ДП.КН.23.504.02.000 ПЗ			
Змн	Арк	№ документа	Підпис	Дата	<div>Розробка проєкту веб-сайту для продуктового магазину "Левас"</div> <div>Пояснювальна записка</div>			
Розроб.		Джигира Н.Б.						
Перевірів		Чубей О.О.						
Реценз.		Посвятовська						
Н. Контр		Гавришків						
Зав.від.		Стефурак Н.А.			<div>Літ.</div> <div>Арк.</div> <div>Аркушів</div> <div>ГФК. ВЧ. ЦК. ІКД. КН-41</div>			

4.4 Розрахунок суми амортизаційних відрахувань	61
4.5 Обчислення накладних витрат	62
4.6 Складання кошторису витрат та визначення собівартості робіт	62
4.7 Розрахунок ціни робіт	63
4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень	63
Висновки	66
Перелік джерел посилань	67
Додатки	69
Додаток А. Текст скрипта для перевірки стану авторизації користувача	69

АНОТАЦІЯ

Головною метою даного дипломного проєкту є створення веб-сайту для продуктового магазину “Левас”.

Дипломний проєкт складається з п’яти розділів: в першому загальному розділі проведено аналітичний огляд існуючих рішень, вказано на доцільність роботи та розроблено технічне завдання.

Другий – основний розділ, містить матеріал по вибору засобів для написання текстів сайту, а також огляд структури бази даних та тестування сайту.

Третій, спеціальний розділ містить інструкції з установки програмного забезпечення, необхідного для розробки, розміщення сайту на хостинг та його експлуатація.

Четвертий, економічний розділ містить розрахунок собівартості розробки веб-сайту.

В п’ятому розділі розглянуті питання охорони праці, вентиляція виробничих приміщень, умови самозаймання речовин та організація служби охорони праці.

Графічна частина проєкту складає чотири плакати формату А1.

Дипломний проєкт носить практично-орієнтований характер і як частину містить повноцінно-функціонуючий сайт, розміщений в мережі Інтернет за наступною адресою: <https://mydiplomlevas.herokuapp.com/> .

					ДП.КН.23.504.02.000 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Інтернет розвивається з шаленою швидкістю. Стрімко зростає число публікацій в Інтернеті, що віщує його проникнення навіть в ті галузі нашого життя, які не зв'язані з технологіями. Інтернет тепер більше схожий на величезне джерело різного роду інформації, ніж на засіб виключно для інтелектуальних людей.

Інтернет з кожним днем все більше входить в різні сфери людського життя, включаючи комерційну діяльність. Найкращим прикладом такої діяльності є інтернет-магазин.

Інтернет-магазин – це сайт в мережі інтернет, на якому проходить торгівля товарами. Цей сайт дозволяє користувачам зробити замовлення якихось товарів, оплатити його, та вказати метод доставки онлайн, в браузері або через мобільний додаток. Продаж при цьому проводиться дистанційно, що накладає певні обмеження.

Сайти дуже популярні серед покупців, так як вони дозволяють оглянути, порівняти та купити товари через мережу. Власнику це також вигідно, так як не потрібно оплачувати роботу персоналу, та орендувати приміщення для ведення своєї діяльності.

Метою даного дипломного проєкту є розробка веб-сайту для продуктового магазину “Левас”. Інтернет-магазин дозволить зробити процес покупки продуктів зручнішим та швидшим. Також веб-сторінка зацікавить більшу кількість покупців. На сайті можна буде зробити замовлення, з можливістю передоплати з використанням банківської карти, переглянути його статус та інші дані. Також буде реалізована система відгуків, що дозволить керівництву магазину оцінити якість товару та попит на нього.

					ДП.КН.23.504.02.000 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ І ПОСТАНОВКА ЗАВДАНЬ

1.1 Аналітичний огляд існуючих рішень

У наш час, інтернет все глибше впроваджується у всі сфери нашого життя, тим самим формує, так звану, інформаційну економіку.

Перехід до інформаційної економіки стає стратегічною ціллю кожної сучасної країни, так як в ній науковому розвитку, масовій цифровізації і електронному керуванню приділяється особлива увага. В таких умовах стає очевидно, що інтернет-технології сприяють формуванню, та розвитку економіки. Внаслідок цього все більша кількість підприємств старається вийти на інтернет-ринок.

Інтернет-торгівля це явище, якому притаманна поява власних бізнес-моделей, без яких стає неможливим саме її існування. Можливість інтерактивної взаємодії між продавцем та клієнтом, дає широкі можливості, такі як можливість мати доступ до підприємства будь-коли та будь-де. Завдяки цьому, продавцю стає набагато легше задовільнити потреби покупців, а також швидко адаптуватись до змін в ринку.

Сьогодні, головні тенденції в розвитку світовою економіки задаються формуванням інтернет-торгівлі. Це зв'язано з певними факторами: забезпечується розширене виробництво на новій матеріальній основі, представляється вільний доступ до технологій та різного роду інформації, які згодом перетворюються на послуги та товари. Підвищується цінність капіталу, так як максимізація продукту проходить без будь-якого виробництва, що є особливістю електронної комерції.

Все вищесказане лише доводить той факт, що теперішнє формування економічних відносин проходить в умовах нового, постіндустріального рівня.

В наш час інтернет-торгівля стає все поширенішою.

Інтернет-торгівля – це така сфера економіки, яка включає в себе проведення фінансових операцій, з використанням комп'ютерної мережі, а

					ДП.КН.23.504.02.000 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

також бізнес-процеси, які з ними зв'язані. Сьогодні інтернет є важливим інструментом для ведення успішного бізнесу.

Інтернет став потужним інструментом для розвитку торгівлі, а також ведення бізнесу завдяки тому що він забезпечує дистанційну взаємодію, не маючи при цьому територіальних чи часових обмежень.

Близько 95% всіх користувачів мережі інтернет використовують його для пошуку інформації, замовлення товарів чи послуг, для роботи, дому, підприємництва і багато іншого. Електронна комерція все більше входить в наше життя. Більшість компаній вже займаються веденням бізнесу в інтернеті.

У 1991 році Інженер Тімоті Джон Бернерс-Лі створив перший веб-сайт. Це була примітивна HTML-сторінка, яка була розроблена для того аби показати як виглядає гіпертекст. Відтоді пройшло багато часу і на сьогоднішній день веб-сторінки виглядають зовсім інакше. При їх розробці використовується широкий спектр програмних засобів і різноманітних мов програмування.

Для написання дипломної роботи було проаналізовано інтернет-магазини Gastro Freshbaza та Delikatto. За допомогою цих сайтів вивчивчено особливості онлайн-продажу продуктів харчування та товарів для ресторанного бізнесу, процес замовлення, оплати та доставки, а також з системи знижок та умови співпраці. Аналіз цих сайтів надав корисну інформацію.

Gastro Freshbaza - це український інтернет-магазин, який спеціалізується на продажу продуктів харчування та товарів для ресторанного бізнесу.

Сайт має простий та зрозумілий дизайн, що дозволяє з легкістю знайти необхідні товари. Головна сторінка містить інформацію про акції та пропозиції, а також розділи з продуктами для приготування страв, напоїв та кондитерських виробів (Рисунок 1.1).

					ДП.КН.23.504.02.000 ПЗ	Арк.
						10
Зм.	Арк.	№ докум.	Підпис	Дата		

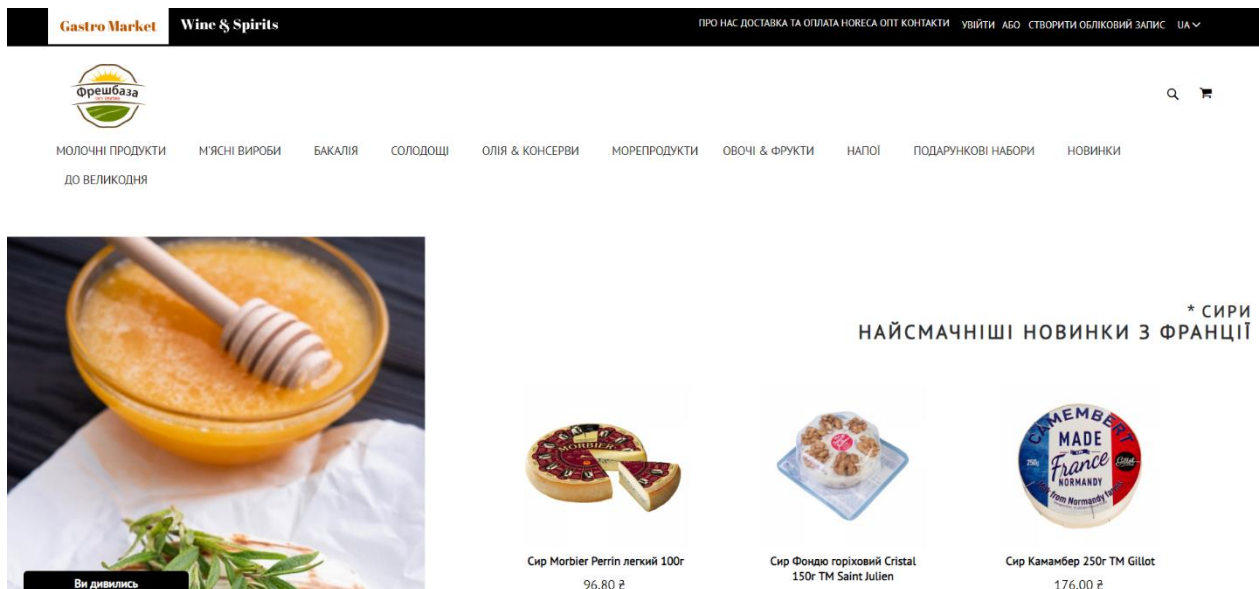


Рисунок 1.1 – Gastro Freshbaza

Кожен товар супроводжується детальним описом, фото та ціною. Зручною функцією є можливість зберігати товари в "Кошику" для зручності пізнішого оформлення замовлення.

Для зручності покупців сайт має систему онлайн-замовлень та оплати товарів. Доставка може бути здійснена по всій Україні, а також самовивіз з точок видачі, які розташовані в різних містах.

Також, на сайті є можливість зареєструватися як партнер та отримувати спеціальні ціни та умови співпраці для ресторанного бізнесу.

Узагалі, Gastro Freshbaza є зручним та надійним місцем для придбання продуктів харчування та товарів для ресторанного бізнесу з доставкою по всій Україні та зручним процесом замовлення та оплати.

Delikatto - це український інтернет-магазин, який пропонує широкий асортимент продуктів харчування та товарів для дому.

Сайт має чіткий та зручний дизайн, який дозволяє з легкістю знайти необхідні товари. На головній сторінці розміщено актуальні пропозиції, новинки та спеціальні пропозиції. В меню сайту можна вибрати необхідну категорію товарів, такі як "Харчові продукти", "Вино та алкоголь", "Товари для дому", "Краса та здоров'я" тощо (Рисунок 1.2).

					ДП.КН.23.504.02.000 ПЗ	Арк.
						11
Зм.	Арк.	№ докум.	Підпис	Дата		

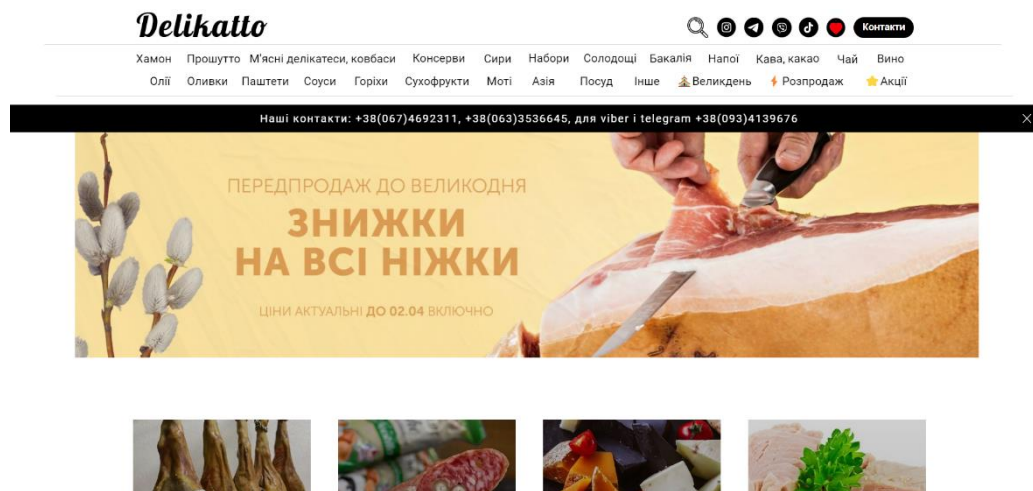


Рисунок 1.2 – Delikatto

Кожен товар супроводжується детальним описом, фото та ціною. Зручною функцією є можливість зберігати товари в "Обраному" для зручності пізнішого вибору та замовлення. Крім того, на сайті можна скористатися пошуком товарів за назвою, брендом чи категорією.

Для зручності покупців сайт має систему оформлення замовлень та оплати товарів в режимі онлайн. Доставка може бути здійснена по всій Україні, а також самовивіз з точок видачі, які розташовані в різних містах.

Також, сайт має програму лояльності для постійних покупців, яка дозволяє зекономити на покупках та отримувати різні бонуси та пропозиції.

Узагалі, Delikatto є зручним та надійним місцем для онлайн-шопінгу з великим вибором товарів та добре організованим процесом замовлення та доставки.

Отже, для успішної розробки сайту, необхідно розробити як адміністративну, так і клієнтську частину з різноманітними функціями. Адміністративна частина повинна містити можливості для додавання товарів, адміністрування замовлень та користувачів, а також додавання новин. Клієнтська частина повинна надавати можливість оформлення замовлення, залишення відгуків та фільтрування товарів під час пошуку. Забезпечення наявності цих функцій допоможе підвищити користувацький комфорт та забезпечити ефективне функціонування сайту.

					ДП.КН.23.504.02.000 ПЗ	Арк.
						12
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2 Система обґрунтування доцільності

Система обґрунтування доцільності включає розробку логічного аргументу, який підтримує необхідність або користь певної дії, рішення або проєкту. Ця система допомагає переконати інших людей у правомірності та ефективності певної ініціативи, а також допомагає вам самому зрозуміти і оцінити можливі наслідки вашого вибору.

Основні кроки системи обґрунтування доцільності включають:

- Визначення мети: Формулювання мети, яку потрібно досягти або проблеми, яку потрібно вирішити. Мета повинна бути специфічною, вимірюваною, досяжною, релевантною та масштабованою (SMART-критерії).
- Збір і аналіз інформації: Збір необхідної інформації, яка підтверджує доцільність вашої ініціативи. Це можуть бути дані, статистика, наукові дослідження, експертні оцінки тощо. Аналіз інформації, щоб виявити переваги та можливі недоліки вашого підходу.
- Визначення переваг: Визначення переваг, які приносить ініціатива. Це можуть бути економічні вигоди, соціальна користь, покращення ефективності, рішення проблеми, забезпечення конкурентного переваги тощо. Аргументування важливості і доцільності переваг.
- Оцінка ризиків: Розгляд можливих ризиків і недоліків ініціативи. Ідентифікація можливих проблем, які можуть виникнути, і з'ясування, шляхів вирішення або зменшення їх впливу. Складання резервних планів та альтернативних рішень для демонстрації, належного аналізу ситуації.
- Логічні аргументи: Побудова логічних аргументів, які пояснюють зв'язок між ініціативою і досягненням мети. Використання фактів, доказів, аналізу, аналогій або прикладів для підтримки ствердження. Особлива увага на потенційні запитання або сумніви, які можуть виникнути і включення в аргумент відповідні пояснення.

					ДП.КН.23.504.02.000 ПЗ	Арк.
						13
Зм.	Арк.	№ докум.	Підпис	Дата		

– Вплив на стейкхолдерів: Розгляд впливу ініціативи на різні зацікавлені сторони або стейкхолдерів. Розгляд можливих об'єкцій стейкхолдерів.

– Викладення аргументу: Представлення аргументу у зрозумілій, лаконічній та переконливій формі. Використання чіткого мовного стилю, структурування аргументації логічно і систематично. Візуалізація даних або інформації, коли це можливо, за допомогою графіків, діаграм або ілюстрацій.

Система обґрунтування доцільності для веб-сайту продуктового магазину включає наступні деталі та тести:

Аналіз ринку та конкурентів:

– Детальне дослідження ринку продуктових магазинів та їх присутності в Інтернеті.

– Аналіз конкурентів, їхніх веб-сайтів та функціональних можливостей.

– Виявлення прогалин у присутності магазину "Левас" в Інтернеті та можливостей для вдосконалення.

Покращення доступності та зручності:

– Пояснення необхідності наявності веб-сайту для залучення нових клієнтів та збереження існуючих.

– Аналіз переваг онлайн-покупок та зручності отримання інформації про товари через веб-сайт.

– Розробка зручної навігації, швидкого пошуку товарів та простого процесу замовлення.

Підвищення продажів та залучення клієнтів:

– Демонстрація впливу веб-сайту на збільшення охоплення цільової аудиторії та залучення нових клієнтів.

– Переконливе обґрунтування можливості збільшення обсягу продажів через зручність та доступність покупок онлайн.

					ДП.КН.23.504.02.000 ПЗ	Арк.
						14
Зм.	Арк.	№ докум.	Підпис	Дата		

– Використання стратегій маркетингу та промоційних акцій через веб-сайт для привертання уваги клієнтів.

Тестування та оцінка ефективності:

– Розробка тестового середовища для перевірки функціональності веб-сайту.

– Проведення тестів користувацького досвіду та здійснення замовлень для перевірки працездатності та зручності процесу.

– Зібрання та аналіз отриманих даних з тестування для оцінки ефективності веб-сайту та виявлення можливих покращень.

Оцінка ризиків та недоліків:

– Ідентифікація потенційних ризиків, пов'язаних з розробкою та використанням веб-сайту, наприклад, проблеми з безпекою, технічні неполадки тощо.

– Аналіз можливих недоліків та шляхи їх усунення для мінімізації негативного впливу на роботу магазину та задоволення клієнтів.

В цілому, система обґрунтування доцільності включає широкий спектр аргументів, деталей та тестів, які підтверджують необхідність та користь розробки веб-сайту для продуктового магазину "Левас". Це дозволяє переконати стейкхолдерів в важливості та доцільності проєкту.

					ДП.КН.23.504.02.000 ПЗ	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

2.1 Огляд основних інструментів для розробки веб сайтів

Сьогодні розробку веб-сторінок поділяють на два етапи:

- Frontend відповідає за інтерфейс. Він забезпечує користувачу можливість взаємодіяти з сайтом через різні засоби, такі як кнопки, таблиці, текстові поля та інше.

- Backend обробляє отримані запити, і відсилає якусь інформацію.

Для розробки інтерфейсу використовують багато різних засобів та мов програмування.

HTML (Hypertext Markup Language — мова гіпертекстової розмітки) використовується для розмітки сайту, та зберігається як текстовий файл, з розширенням *.html та *.htm. Вміст веб-сторінки описується у вигляді HTML-тегів. З їх допомогою здійснюється форматування тексту, вставлення таблиць, зображень, посилань на інші документи в мережі інтернет та інші елементи сайту.[7]

HTML надає засоби для:

- публікації електронних документів з текстом різного стилю, фотографіями, таблицями і т.д.
- завантаження певної інформації при натисканні на гіпертекстове посилання.
- розробки форм для проведення платежів, пошуку інформації, фільтрування і т.д.
- введення таблиць, зображень, відеоматеріалів та ін. безпосередньо в документ.

Приклад HTML-документу наведено у лістингу 2.1.

Лістинг 2.1 – Приклад HTML-документу

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

```

<title>Example</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>
  <nav>
    <ul>
      <li><a href="#">Головна</a></li>
      <li><a href="#">Про нас</a></li>
      <li><a href="#">Контакти</a></li>
    </ul>
  </nav>
  <h1>Ласкаво просимо на мій веб-сайт!</h1>
  <p>Це перший абзац мого тексту. Я дуже радий поділитись
з вами моїми думками та ідеями.</p>
</body>
</html>

```

HTML-документ складається з елементів. Елементи використовуються для зображення вмісту сторінки певним чином, та для надання йому певних властивостей. Кожен елемент складається з тегів, атрибутів та вмісту. Теги являють собою назву елемента, записану латинськими літерами в дужках (< > та </ >). Атрибути записуються в дужках після початкового тегу та використовуються для введення додаткової інформації про елемент. Вміст – це вся інформація, яка виводиться на екран.[8]

Структуру HTML-елементу зображено на рисунку 2.1.

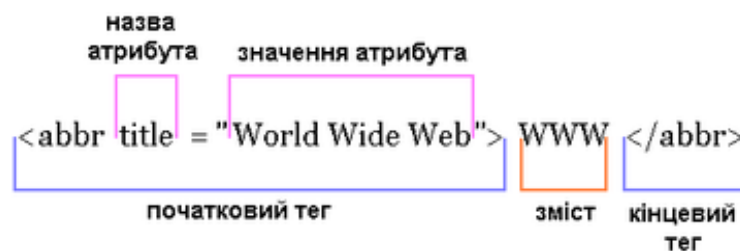


Рисунок 2.1 – Структура HTML-елементу

HTML використовується для визначення структури веб-сторінки, а для форматування використовується CSS.

CSS (Каскадні таблиці стилів) – це спеціальна мова, яка використовується для полегшення процесу розробки веб-сторінок.

CSS відповідає за зовнішній вигляд сторінки. З його допомогою можна вказувати шрифт тексту, його розмір, стиль, колір, розмір зображень, відступи, фон, вид на різних пристроях і т.д.

Зазвичай опис стилів виноситься в окремий файл з розширенням *.css. Це дозволяє краще сконцентруватись на різних задачах, так як в HTML-файлі програміст описує тільки структуру сайту, а в CSS-файлі удосконалює його вигляд.[9]

CSS-стиль складається з селектора, та певної кількості властивостей та значень.

Приклад структури CSS-стилів зображено на рисунку 2.2.



Рисунок 2.2 – Приклад структури CSS-стилів

Приклад структури CSS-файлу наведено у лістингу 2.2.

Лістинг 2.2 – Приклад структури CSS-файлу

```
/* Стили для заголовка первого уровня */
h1 {
  color: #007bff;
  font-size: 2.5rem;
  margin-bottom: 1.5rem;
}

/* Стили для абзацев */
p {
  color: #6c757d;
  font-size: 1.25rem;
  line-height: 1.5;
  margin-bottom: 1.5rem;
}
```

Можливості CSS можна розширити використовуючи Scss.

Scss (Синтаксично досконалі таблиці стилів) – метамова, яка компілюється в звичайні CSS-стилі.

Scss робить процес розробки стилів швидшим, та зручнішим, завдяки введенню нових механізмів, доступних в різних мовах програмування, але відсутніх в CSS. Серед них: змінні, наслідування, вкладені правила та функції.[10]

Приклад використання змінних наведено у лістингу 2.3.

Лістинг 2.3 – Приклад структури CSS-файлу

```
/* Оголошення змінних */
$primary-color: #007bff;
$secondary-color: #6c757d;

/* Стили для заголовка першого рівня */
h1 {
  color: $primary-color;
  font-size: 2.5rem;
  margin-bottom: 1.5rem;
}
/* Стили для абзаців */
p {
  color: $secondary-color;
  font-size: 1.25rem;
  line-height: 1.5;
  margin-bottom: 1.5rem;
}
```

Приклад використання вкладених правил наведено у лістингу 2.4.

Лістинг 2.4 – Приклад використання вкладених правил

```
/* CSS */
.container {
  padding: 1rem;
  background-color: #f2f2f2;
}

.container h1 {
  color: #007bff;
  font-size: 2.5rem;
  margin-bottom: 1.5rem;
}

.container p {
  color: #6c757d;
  font-size: 1.25rem;
  line-height: 1.5;
  margin-bottom: 1.5rem;
}
/* SCSS */
.container {
  padding: 1rem;
```

					ДП.КН.23.504.02.000 ПЗ	Арк.
						19
Зм.	Арк.	№ докум.	Підпис	Дата		

```

background-color: #f2f2f2;

h1 {
  color: #007bff;
  font-size: 2.5rem;
  margin-bottom: 1.5rem;
}

p {
  color: #6c757d;
  font-size: 1.25rem;
  line-height: 1.5;
  margin-bottom: 1.5rem;
}
}

```

Для обробки сценаріїв, в HTML-документах використовується мова програмування JavaScript.

JavaScript – це реалізація стандарту EcmaScript. Вона використовується як в frontend, так і на backend. В той час як HTML і CSS описують структуру і стилі, JavaScript забезпечує інтерактивні елементи на сторінці, а також зв'язок з сервером та базами даних. На JavaScript також програмують backend (Node.js), використовують для розробки ігор (GDevelop, Impact.js), комп'ютерних та мобільних додатків (Electron.js, React Native), подій в прикладному ПЗ (наприклад, Adobe Creative Suite), в PDF-документах і тд.

Для розробки клієнтської частини використовується бібліотека React.

React — це JavaScript-бібліотека, з використанням якої розробляються інтерфейси. В цій бібліотеці інтерфейси пишуться за допомогою такого інструмента, як компоненти, які являють з себе окремі блоки коду. Завдяки React, процес розробки інтерактивних інтерфейсів стає набагато простішим та швидшим. Перевагою є те що у разі зміни коду певного компоненту, React оновить тільки його. React дуже простий в освоєнні, так як в ньому немає великої кількості концепцій.[13]

Приклад React-компоненту наведено у лістингу 2.5.

Лістинг 2.5 – Приклад React-компоненту

```

import React from 'react';
function Greeting(props) {

```

					ДП.КН.23.504.02.000 ПЗ	Арк.
						20
Зм.	Арк.	№ докум.	Підпис	Дата		

```

    return (
      <div>
        <h1>Hello, {props.name}!</h1>
        <p>Welcome to my website.</p>
      </div>
    );
  }

  export default Greeting;

```

Також популярною бібліотекою для написання клієнтської частини є Vue.js.

Vue.js - це прогресивна бібліотека мови JavaScript, яку використовують для написання інтерфейсів. Основною перевагою цього фреймворку є його простота та швидкість вивчення. Вже просто знаючи HTML, CSS, JavaScript можна писати коди з використанням Vue.js. Він одночасно виступає в ролі бібліотеки та повноцінного фреймворку. Це є основною ідеєю Vue.js – комбінування основних переваг React та Angular.[15]

Так само як і React, в Vue.js інтерфейси пишуться з використанням компонентів.

Приклад компоненту Vue.js наведено у лістингу 2.6.

Лістинг 2.6 – Приклад компоненту Vue.js

```

<template>
  <div>
    <h2>{{ title }}</h2>
    <ul>
      <li v-for="item in items" :key="item.id">{{ item.name }}</li>
    </ul>
  </div>
</template>

<script>
export default {
  name: 'MyComponent',
  props: {
    title: String,
    items: Array
  }
}
</script>
<style>

```

					ДП.КН.23.504.02.000 ПЗ	Арк.
						21
Зм.	Арк.	№ докум.	Підпис	Дата		

```
/* CSS стилі для компонента */  
</style>
```

Для розробки серверної частини використовується Node.js.

Node.js представляє собою одноядерне, крос-платформне середовище, яке виконує код на серверній частині сайту. На відміну від інших рішень, не потребує сторонніх серверів, таких як Apache чи Nginx, так як саме є сервером.

Основою Node.js служить рушій V8, розроблений компанією Google, який використовується в браузерах Chromium.

Node.js – це подійно-орієнтована система, в якій модель виконання входу-виходу виконується таким чином, що ніколи не блокується основний процес. А відсутність єдиного блокуемого потоку виконання дозволяє робити додатки ефективними та високопродуктивними.

Його головною перевагою є можливість писати і клієнтську, і серверну частину з використанням однієї мови – JavaScript.

Приклад коду Node.js наведено у лістингу 2.7.

Лістинг 2.7 – Приклад коду Node.js

```
require('dotenv').config()  
const express = require('express')  
const cors = require('cors')  
const mongoose = require('mongoose')  
const cookieParser = require('cookie-parser')  
const path = require('path');  
const stripe = require("stripe")(process.env.stripe_private)  
const router = require("./routers/index.js")  
const errorHandler = require("./middleware/errorHandlingMiddleware")  
const app = express()  
const port = process.env.PORT || 8000;  
  
app.use(cors());  
app.options('*', cors());  
app.use(express.json());  
app.use(cookieParser());  
app.use(express.static(path.join(__dirname,  
"../../client/build")));  
app.use("/api", router)  
app.use(errorHandler)  
start()
```

					ДП.КН.23.504.02.000 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

До відомих технологій для написання серверної частини сайту відносять PHP.

PHP – це мова, розроблена для обробки HTML-сторінок на стороні серверу. Його особливістю є те, що він включає в себе HTML. Код, написаний з використанням PHP зазвичай знаходиться в тезі `<?php і ?>` в HTML файлі, або в файлі з розширенням `.php`.

Різниця між PHP та JavaScript і іншими мовами програмування полягає в тому, що на сервері PHP-скрипти генерують HTML, який передається на клієнтську частину сайту. Користувач отримує лише результат скрипту, тобто гіпертекстовий документ, а при певній настройці буде неможливо визначити чи використовувався PHP взагалі.

PHP дуже легко вивчити, завдяки хорошій документації, яка знаходиться на офіційному сайті

Приклад використання PHP наведено у лістингу 2.8.

Лістинг 2.8 – Приклад використання PHP

```
<!DOCTYPE html>
<html>
<head>
    <title>PHP</title>
</head>
<body>
    <?php
        // виводимо повідомлення
        echo "<h1>Привіт, світ!</h1>";
    ?>
</body>
</html>
```

Дані сайту теж потрібно якимось чином зберігати. Найпопулярнішим рішенням для цього є СУБД MySQL.

MySQL – це система управління реляційними базами даних. Дані в ній зберігаються у вигляді таблиць. MySQL спочатку був повністю безкоштовним, тому код є у відкритому доступі, але за отримання послуг підтримки потрібно заплатити. В MySQL передбачена сумісність з всіма операційними системами.

Приклад зберігання даних в СУБД MySQL зображено на рисунку 2.3.

					ДП.КН.23.504.02.000 ПЗ	Арк.
						23
Зм.	Арк.	№ докум.	Підпис	Дата		

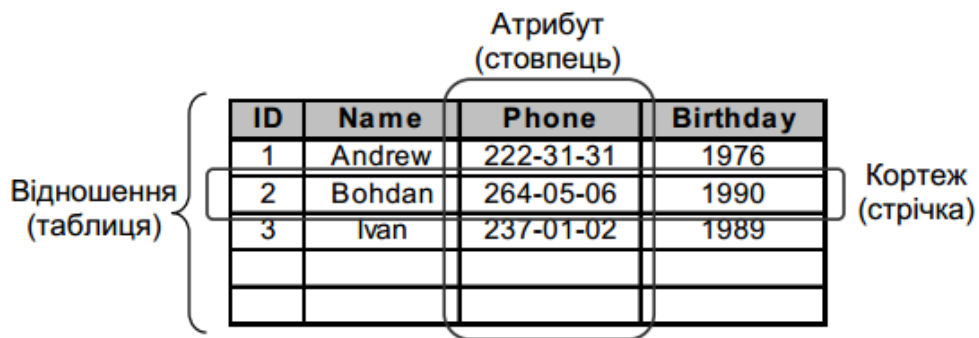


Рисунок 2.3 – Зберігання даних в СУБД MySQL

Для зберігання даних також використовується СУБД MongoDB.

MongoDB - це документно-орієнтована система управління базами. Дані в ній зберігаються у вигляді JSON, тобто об'єктів мови JavaScript. Вона є платною.

Відмінності MongoDB та MySQL показано на рисунку 2.4.

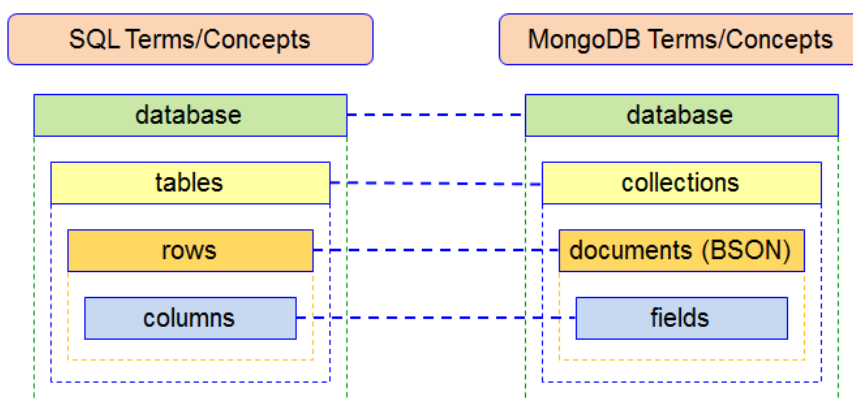


Рисунок 2.4 – Відмінності MongoDB та MySQL

Для написання сайту, використовуючи описані вище технології необхідно встановити Node Package Manager, для завантаження різних бібліотек.

Отже, результатом проведення аналітичного огляду існуючих рішень, є вибір стеку React.js, Node.js для написання коду сайту, а також СУБД MongoDB для зберігання даних.

2.2 Постановка задачі на розробку програмного забезпечення

Основне завдання цього дипломного проєкту – це розробка сайту для продуктового магазину «Левас». Сайт повинен виконувати наступні функції:

1) Функціонування та відображення головного меню. Переключення між пунктами меню та виділення поточного пункту. У підпунктах головного меню користувач має можливість здійснити такі операції:

- обрати категорію товару;
- створення аккаунту покупця;
- забезпечення входу у створений аккаунт;
- отримання інформації про вибрані товари у "Корзині".

2) Забезпечення пошуку товарів по сайту. Створення фільтрів пошуку, які допоможуть уточнити пошук.

3) Виведення інформації про наявність "гарячих" пропозицій та акцій покупцеві.

Також сайт повинен бути швидким та гнучким.

Структурна схема проєкту зображена на листі 2022.ДП.123.406.12.00.00 СС.

2.3 Опис та обґрунтування вибору методу організації вхідних та вихідних даних

Для досягнення мети, поставленої в технічному завданні було вирішено використати наступні засоби.

1) CSS – каскадна таблиця стилів. Він використовується для розділення процесу написання гіпертексту, та стилів. CSS зазвичай виноситься в окремий файл з розширенням .css.

CSS може під'єднуватись до документу такими способами:

- Через тег <link> з вказанням ссылки на документ, який має розширення .css.

					ДП.КН.23.504.02.000 ПЗ	Арк.
						25
Зм.	Арк.	№ докум.	Підпис	Дата		

- Через тег <style> в голові документу.
- Через атрибут style в тезі.

В цьому проєкті буде використовуватись Scss.

Scss – предпроцесор CSS. Він відрізняється від CSS тим, що в ньому присутні функції скриптових мов програмування. Sass компілюється в CSS.

Його головними перевагами є хороша документація, велика база користувачів та наявність багатьох засобів. За час свого існування зарекомендував себе як хороше рішення.

З його допомогою можна використовувати:

- Змінні.
- Вкладені правила.
- Змішані стилі.
- Функції.

Різницю між CSS та SCSS наведено у лістингу 2.9.

Лістинг 2.9 – Різниця між CSS та SCSS

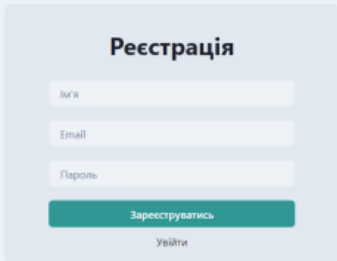
```
/* SCSS */
section {
  height: 100px;
  width: 200px;
  .class-one {
    height: 100px;
    width: 200px;
    .button {
      color: #fff;
    }
  }
}
/* CSS */
section {
  height: 100px;
  width: 200px;
}
section .class-one {
  height: 100px;
  width: 200px;
}
section .class-one .button {
  color: #fff;
}
```

У лістингу 2.10 наведено приклад використання SCSS на сторінці з формою реєстрації.

Лістинг 2.10 – Приклад використання SCSS на сторінці з формою реєстрації

```
#form {  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
  flex: 1;  
  margin: 0 60px;  
  max-width: 600px;  
  padding: 50px;  
  background-color: rgb(236, 236, 236);  
  border-radius: 10px;  
  
  & input {  
    margin: 10px 0;  
    border: 1px solid gray;  
    height: 2rem;  
    border-radius: 5px;  
    width: 100%;  
  }  
}
```

Форму реєстрації зображено на рисунку 2.5.



The image shows a registration form titled "Регістрація" (Registration) centered on a light blue background. The form is contained within a light gray rounded rectangle. It features three input fields: "Ім'я" (Name), "Email", and "Пароль" (Password). Below the password field is a teal button labeled "Зареєструватись" (Register). At the bottom of the form is a link labeled "Увійти" (Login).

Рисунок 2.5 – Форма реєстрації

2) React.js – бібліотека мови JavaScript, яка використовується для написання веб-інтерфейсів.

До його переваг відносять:

- Швидкість.
- Простота.
- Можливість інтеграції з іншими фреймворками.
- Використання JSX-синтаксису.

JSX – це скорочення від JavaScript XML. Це файл, який вміщає в собі коди JavaScript та гіпертекст HTML. Він робить додаток надійнішим та швидшим.

У лістингу 2.11 наведено JSX синтаксис на сторінці з товарами.

Лістинг 2.11 – JSX синтаксис на сторінці з товарами

```
return (  
  <>  
  <Container    p="30px"    maxWidth="1500px"    display="flex"  
    flexDir="column" alignItems="center">  
    <Button ref={btnRef} colorScheme='teal' onClick={onOpen}  
      marginBottom="15px" w="200px">  
      Фільтри  
    </Button>  
    <SimpleGrid    templateColumns={`repeat(${getColumns()},  
250px)`}    justifyItems="center"    minChildWidth='250px'  
    spacing='20px'>  
    {goodsList.map((good) => (  
    <ProductCard key={good.id} param={good} />  
    ))}  
    </SimpleGrid>  
  </Container>  
  <Drawer  
    isOpen={isOpen}  
    placement='left'  
    onClose={onClose}  
    finalFocusRef={btnRef}>  
    <DrawerOverlay />  
    <DrawerContent>  
    <DrawerCloseButton />  
    <DrawerHeader>Фільтр</DrawerHeader>  
    <DrawerBody><Box>  
    <Select color="black" placeholder='Наявність' onChange={(e)  
=> setState(e.target.value)}>  
    {is.map((isfilter) => <option  
key={isfilter}>{isfilter}</option>)}  
    </Select></Box>
```

					ДП.КН.23.504.02.000 ПЗ	Арк.
						28
Зм.	Арк.	№ докум.	Підпис	Дата		

3) Node.js – середовище JavaScript для розробки серверної частини сайту, написане на рушії V8.

Його переваги:

- Кросплатформність.
- Величезна кількість безкоштовних пакунків.
- Оптимальна швидкість роботи додатків.
- Можливість повторного використання.
- Можливість писати і клієнтську, і серверну частину сайту однією мовою програмування.

Швидкість Node.js зумовлена виростанням рушія V8, написаного на мові програмування C++.

Приклад Node.js коду наведено у лістингу 2.12.

Лістинг 2.12 – Приклад Node.js коду

```
require('dotenv').config()
const express = require('express')
const cors = require('cors')
const mongoose = require('mongoose')
const cookieParser = require('cookie-parser')
const path = require('path');
const stripe = require("stripe")(process.env.stripe_private)
const router = require("./routers/index.js")
const errorHandler = require("./middleware/errorHandlingMiddleware")
const app = express()
const port = process.env.PORT || 8000;
app.use(cors());
app.options('*', cors());
app.use(express.json());
app.use(cookieParser());
app.use(express.static(path.join(__dirname,
"../../client/build")));
app.use("/api", router)
app.post('/api/stripe', async (req, res) => {
  const price = Number(req.body.price) * 100;
  const intent = await stripe.paymentIntents.create({
    amount: Math.round(price * 100) / 100,
    currency: 'usd'
  });
  res.json({ client_secret: intent.client_secret,
intent_id: intent.id });
});
```

					ДП.КН.23.504.02.000 ПЗ	Арк.
						29
Зм.	Арк.	№ докум.	Підпис	Дата		

2.4 Структура та огляд сайту

Для розробки інтерфейсу веб-сайту було використано такі технології як React та Scss. Основною особливістю розробленого сайту є адаптивність, тобто можливість відображення на мобільних пристроях та планшетах.

Головна сторінка сайту зображена на рисунку 2.6. На ній розміщений банер магазину (Рисунок 2.7), слайдер для відображення інформації про акції (Рисунок 2.8), та акційні товари (Рисунок 2.9).

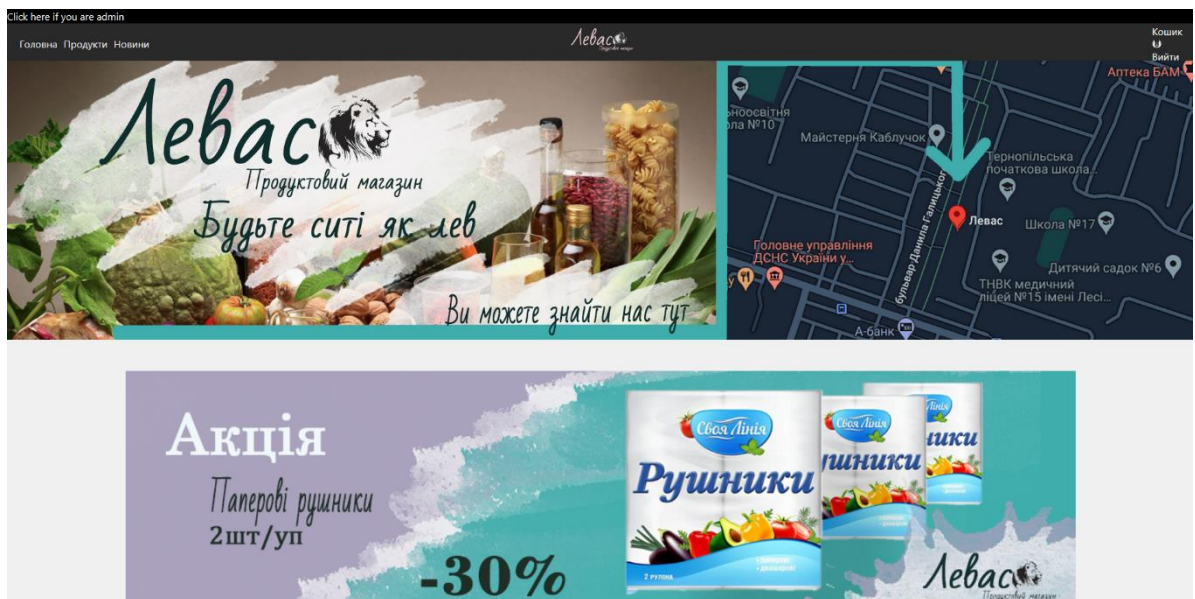


Рисунок 2.6 – Головна сторінка сайту



Рисунок 2.7 – Банер магазину

					ДП.КН.23.504.02.000 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

На цьому банері зображено логотип, гасло та місце розміщення магазину.



Рисунок 2.8 – Слайдер для відображення інформації про акції

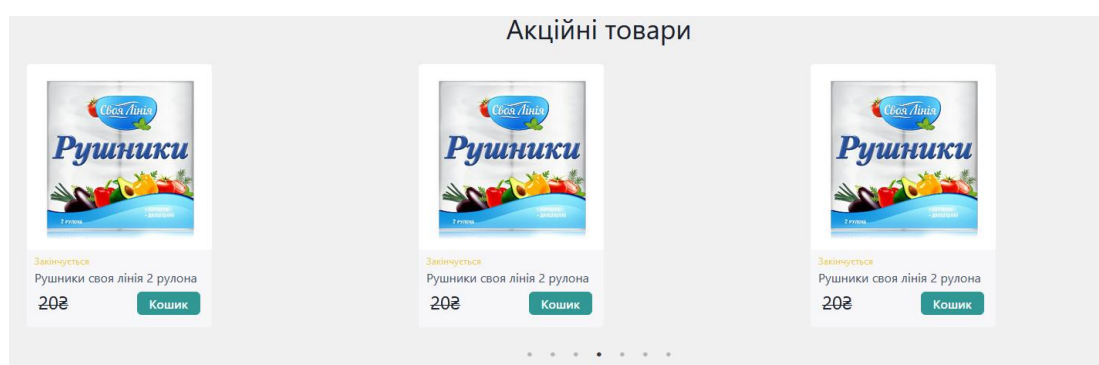


Рисунок 2.9 – Акційні товари

Зверху на сайті знаходиться навігаційна панель, завдяки якій ми можемо перейти на сторінку продуктів, або сторінку новин.

Сторінка продуктів зображена на рисунку 2.10. На ній є кнопка для фільтрації (Рисунок 2.11), при натисканні на яку відкривається меню фільтрації (Рисунок 2.12), а також список товарів (Рисунок 2.13).

					ДП.КН.23.504.02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

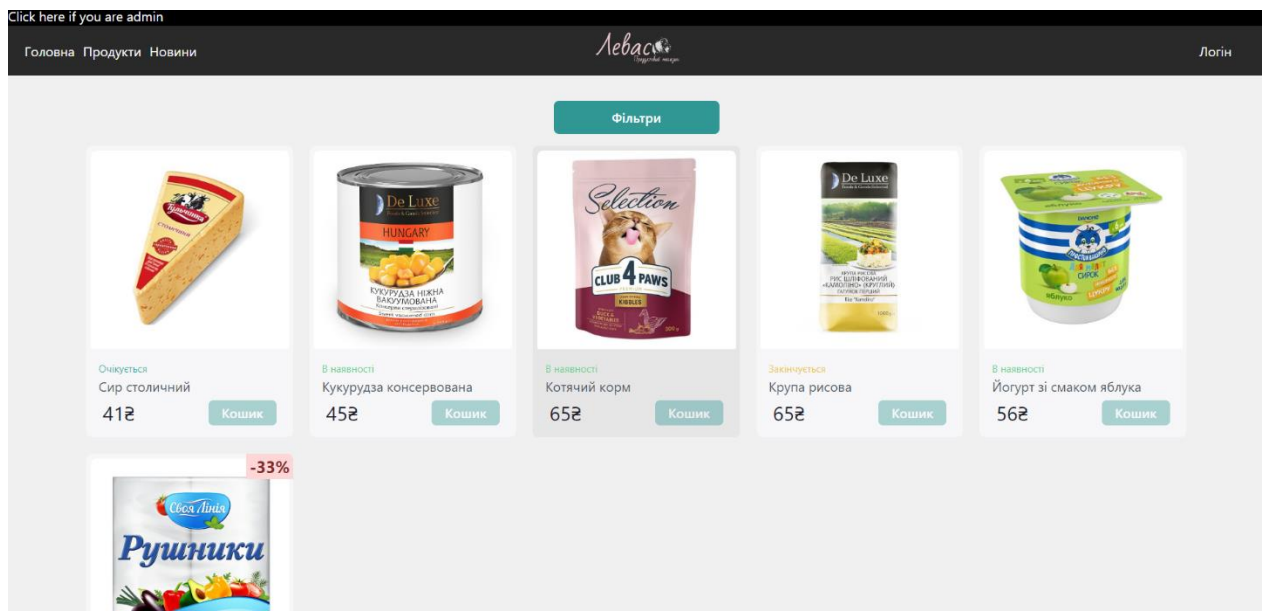


Рисунок 2. 10 –Сторінка продуктів

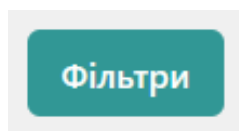


Рисунок 2.11 – Кнопка для фільтрації

Фільтр
×

Наявність
▼

☐ Сіль
☐ Вода
☐ Сир
☐ Котячий корм
☐ Рис
☐ Масло
☐ Кукурудза
☐ Рушники
☐ Йогурт
☐ Морозиво

Гарячі пропозиції
☐ Акційний товар

Скинути фільтри

Скасувати

Підтвердити

Рисунок 2.12 –Меню фільтрації

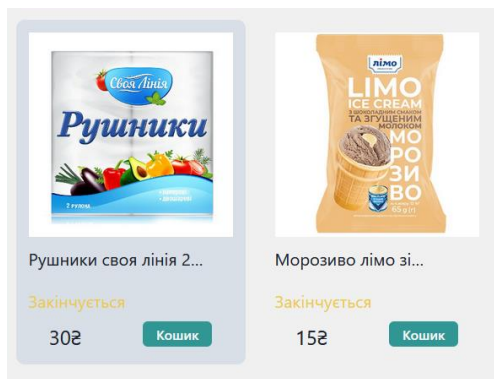


Рисунок 2.13 – Список товарів

Для детального ознайомлення з товаром та проведення покупки необхідно натиснути на товар, який цікавить, після чого відкриється його сторінка (Рисунок 2.14). На ній можна переглянути фотографію, назву та характеристики. Також можна залишити відгук про товар, а також поставити оцінку. Для купівлі товару, потрібно натиснути кнопку “Додати в кошик”, після чого перейти на сторінку кошику (рис 2.15) та натиснути кнопку “Купити”.

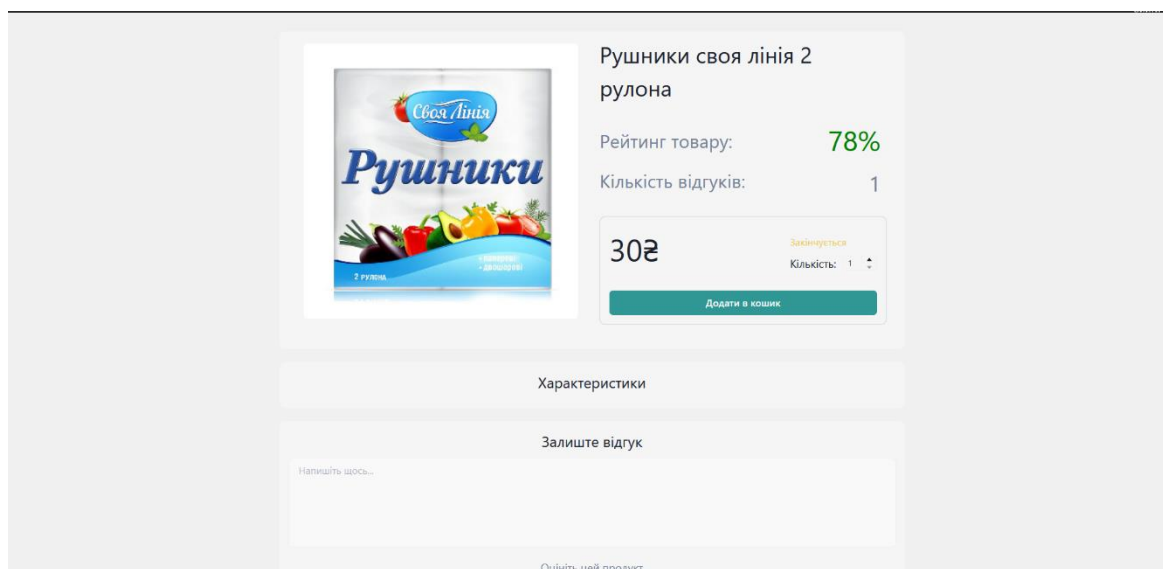


Рисунок 2.14 – Сторінка товару

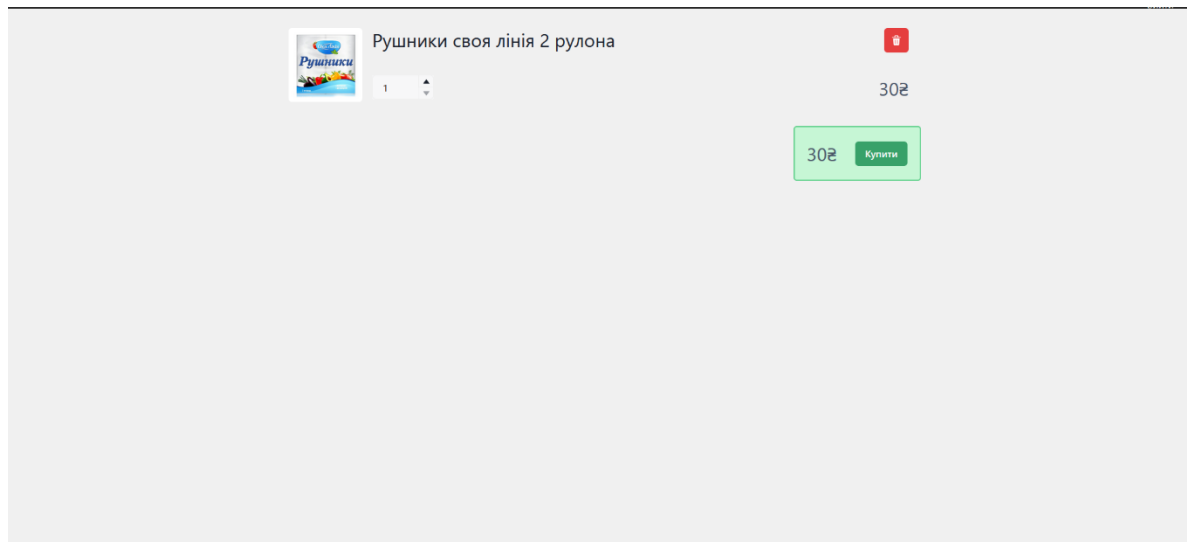


Рисунок 2.15 – Сторінка кошику

2.5 Розробка структури бази даних сайту

База даних – це програма, яка дозволяє зберігати та обробляти певну інформацію в структурованому вигляді. Вона необхідна для зберігання даних сайту. Для цієї задачі була вибрана СУБД MongoDB.

MongoDB – документо-орієнтована база даних. Це означає що дані в ній зберігаються у вигляді JSON-файлів, тобто об'єктів мови JavaScript. Самі ж JSON-файли зберігаються в колекціях. Такий підхід є дуже зручним, так як немає чіткої схеми зберігання, тому кожен об'єкт може складатись з будь-якої кількості полів.

Головний сайт MongoDB зображено на рисунку 2.16.

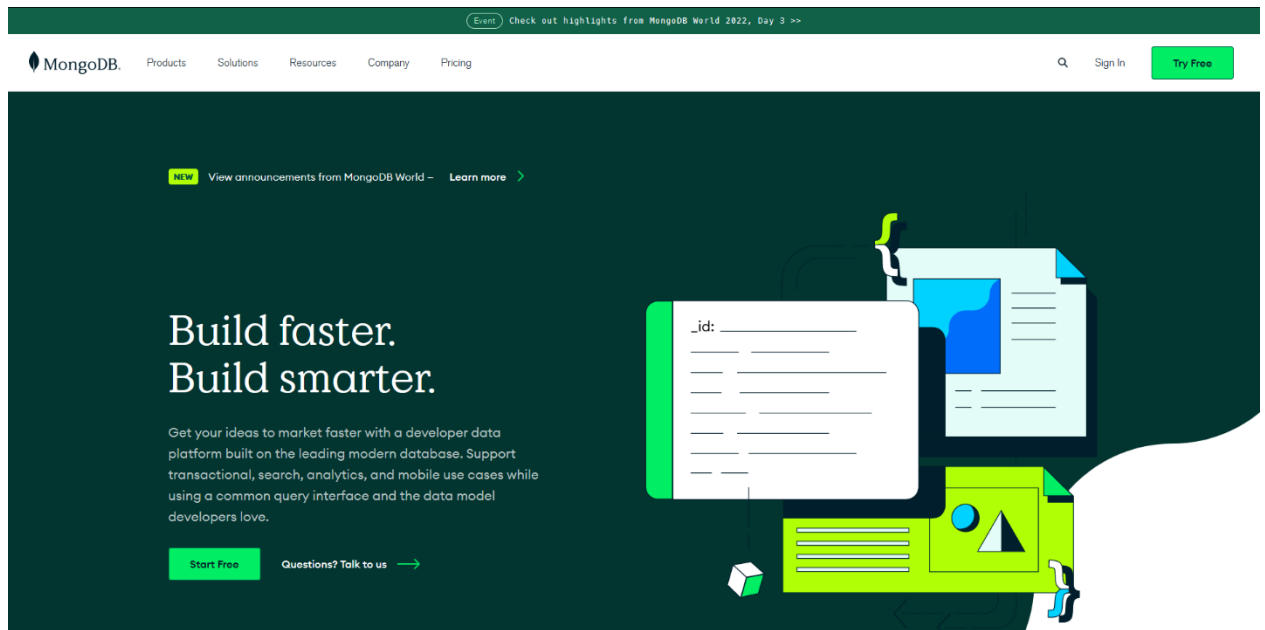


Рисунок 2.16 – Головний сайт MongoDB

Список колекцій зображено на рисунку 2.17.

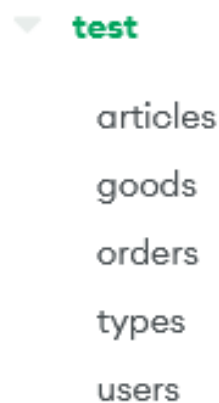


Рисунок 2.17 – Список колекцій

Колекція “articles” містить дані про статті, а саме:

- ID статті.
- Заголовок.
- Текст.
- Дату створення.

Колекція “goods” містить дані про товари, а саме:

- ID товару.

- Назву.
- ID категорії.
- Ціну.
- Акційну ціну, у випадку акції.
- Відгуки, включаючи ім'я користувача, текст, оцінку та дату створення.
- Характеристики товару, включаючи назву і значення.
- Посилання на зображення.
- Стан наявності товару.

Колекція “orders” містить дані про замовлення, а саме:

- ID замовлення.
- Ціна.
- Товари.
- Час замовлення.

Колекція “types” містить дані про категорії замовлення, а саме:

- ID категорії товару.
- Назва категорії.

Колекція “users” містить дані про користувачів, а саме

- ID користувача.
- Ім'я.
- Пошта.
- Пароль.
- Роль.
- Корзина.
- Замовлення.

Поля для колекцій задаються схемами, приклад яких продемонстровано у лістингу 2.13.

Лістинг 2.13 – Поля для колекцій задаються схемами

```
const goodsSchema = new Schema({
  id: { type: String, required: true, unique: true },
  name: { type: String, required: true },
```

					ДП.КН.23.504.02.000 ПЗ	Арк.
						36
Зм.	Арк.	№ докум.	Підпис	Дата		

```

typeID: { type: String, required: true },
price: { type: Number, required: true, min: 1 },
isDiscount: { type: Boolean, required: true, default: false
},
priceWithDiscount: { type: Number, min: 0 },
comments: [{
name: { type: String },
      text: { type: String },
      time: { type: Date },
      grade: { type: Number }
    }],
params: [{
  title: { type: String },
  description: { type: String }
}],
image: { type: String, required: true },
state: { type: String, enum: ["В наявності",
"Закінчується", "Закінчився", "Очікується"], default:
"Очікується" }
})

```

2.6 Програмування сайту

Скрипт — це маленька програма, яка лінійно виконує список однотипних задач. Якщо говорити простіше, то скрипт просто автоматизує якийсь багаторазово повторювальний процес. Прикладом скриптів можуть бути:

- Заміна одних символів на інші.
- Автоматична генерація якогось меню.
- Підготовка файлів для відправки на сервер.
- Перевірка даних.
- Звернення до бази даних.

Правильне використання скриптів робить сайт швидшим, завдяки чому сайту легше просуватись в пошукових системах. Скрипти дуже зручні для користувачів, так як дозволяють використовувати анімації та динамічні об'єкти, такі як слайдери.

До їхніх переваг відносять:

					ДП.КН.23.504.02.000 ПЗ	Арк.
						37
Зм.	Арк.	№ докум.	Підпис	Дата		

- Простота внесення змін. Скрипти ділять програму на велику кількість блоків коду і у випадку якоїсь помилки її стає набагато легше локалізувати та виправити без нанесення шкоди іншим компонентам.
- Прискорення процесу рішення завдань. Скрипт дозволяє зразу зосередитись на рішенні якоїсь задачі, замість того щоб розбирати величезну програму.
- Багатоплатформність. Сценарії, для прикладу, легко адаптувати до браузерів, так як всі вони інтерпретують JavaScript.

На сайті є сторінки, вміст яких доступний лише зареєстрованим користувачам. Такими сторінками є “Кошик” та “Мої замовлення”. Спеціально для цього в дипломному проєкті був реалізований скрипт, який перед запитом до бази даних перевіряє чи користувач існує. Протягом перевірки розшифровується токен, та звіряються дані користувача з базою даних.

Розглянемо 4 випадки його відповіді на прикладі сторінки з кошиком покупця.

1) Якщо користувач зареєстрований, та має дійсний токен, тобто не підроблений чи прострочений, тоді він отримає всі дані про товари у своєму кошику (Рисунок 2.18).

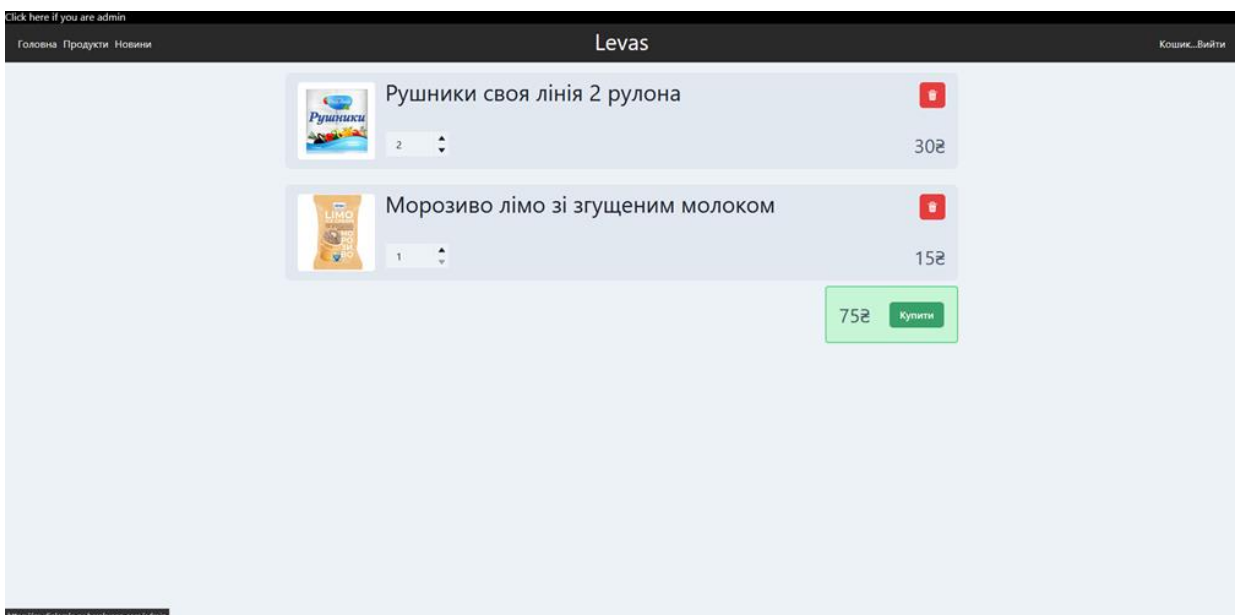


Рисунок 2.18 – Успішне виконання скрипта

					ДП.КН.23.504.02.000 ПЗ	Арк.
						38
Зм.	Арк.	№ докум.	Підпис	Дата		

2) Якщо користувач по якійсь причині не має токена, тоді він отримає помилку “Unauthorized1” (Рисунок 2.19). В результаті він не отримає даних про товари у кошику.

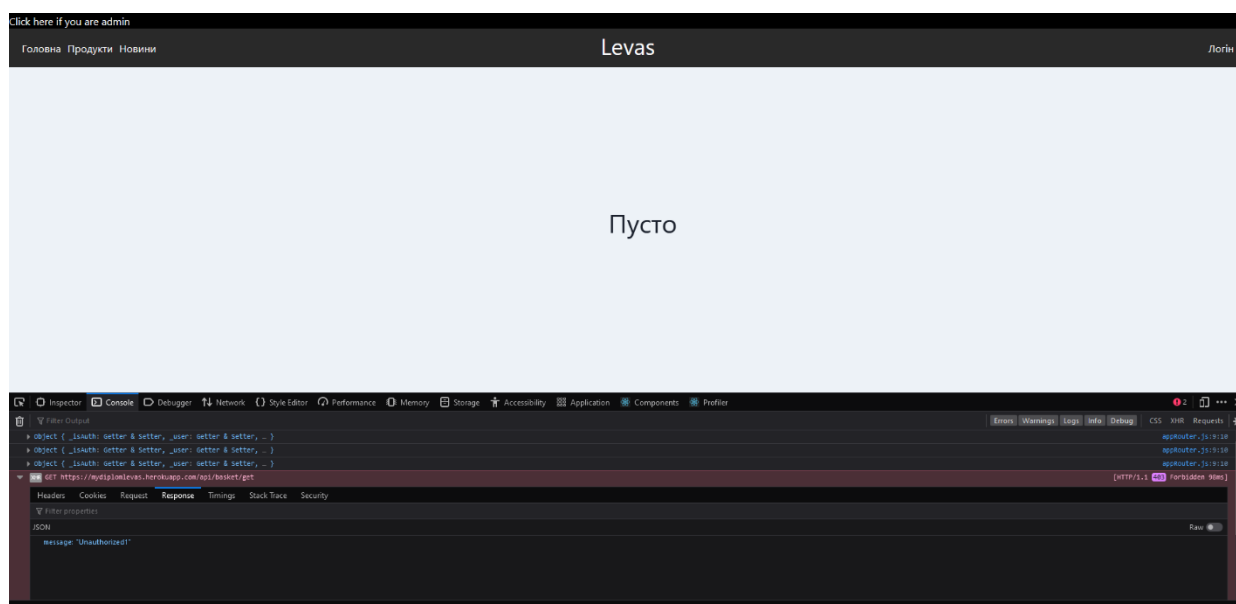


Рисунок 2.19 – Помилка “Unauthorized1”

3) Якщо у користувача є токен, але при розшифруванні було виявлено, що користувача немає в базі даних, тобто токен підроблений, або користувача видалено, тоді він отримає помилку “Unauthorized2” (Рисунок 2.20).

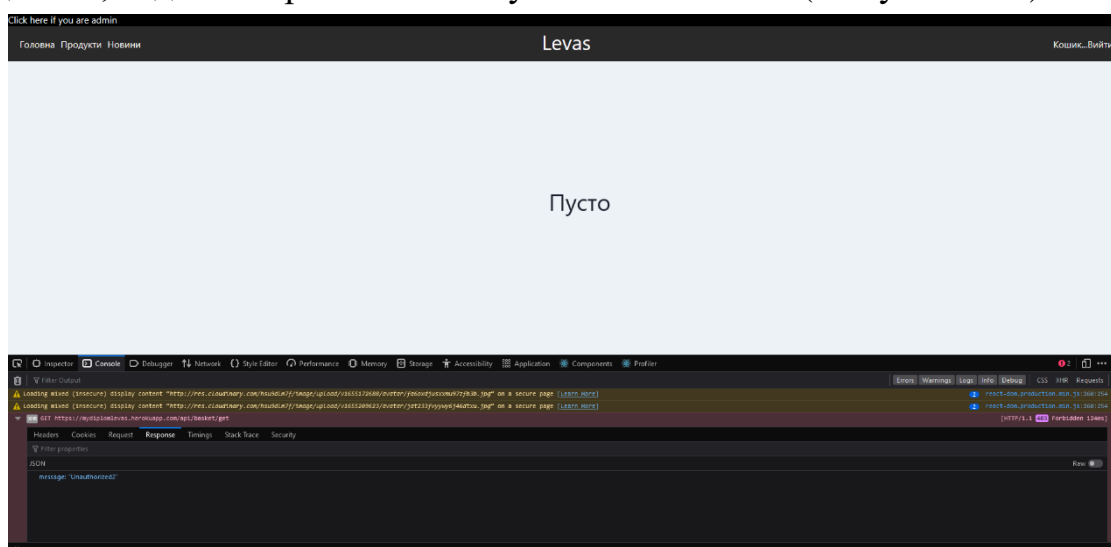


Рисунок 2.20 – Помилка “Unauthorized2”

4) Якщо у користувача неправильний токен, тоді він отримає помилку “Unauthorized3” (Рисунок 2.21).

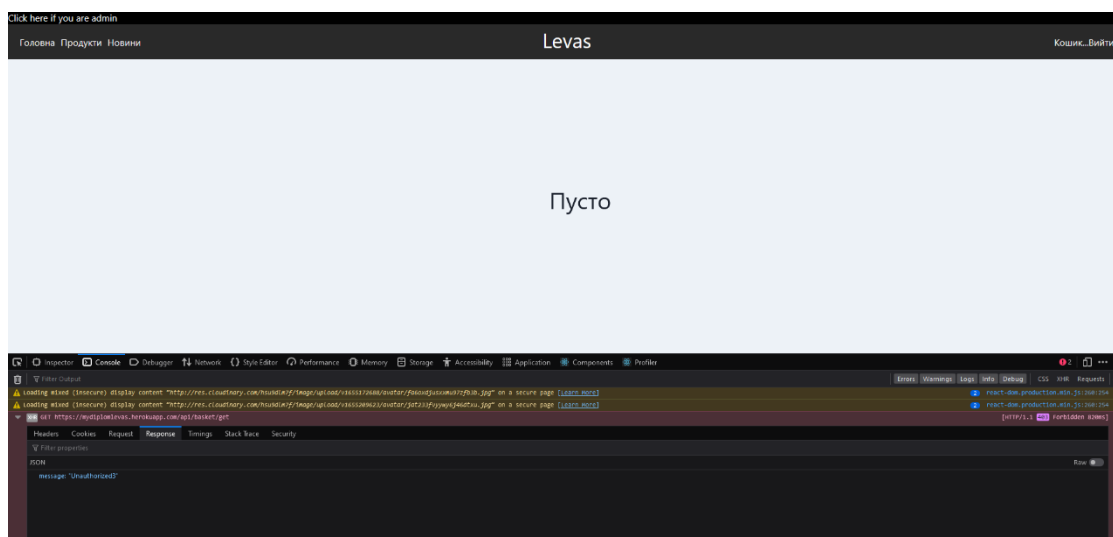


Рисунок 2.21 - Помилка “Unauthorized3”

Тільки після цієї перевірки є можливість провести запит. Цей скрипт значно пришвидшує роботу сайту, так як перед роботою з базою даних відсіюються потенційно зловмисні та неправильні запити.

Код скрипта наведено в Додадку А.

Детальніше з алгоритмом роботи скрипта можна ознайомитись завдяки блок-схемі, зображеній на плакаті 2022.ДП.123.406.12.00.00 БС.

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

3.1 Найменування та область застосування

Завданням дипломного проєкту є створення веб-сторінки для продуктового магазину “Левас”.

Сайт повинен містити:

- Каталог продукції магазину, їх зовнішній вигляд та характеристики.
- Акції.
- Розділ з новинами.

Дизайн повинен бути зручним та сучасним.

Левас – це продуктовий магазин, що знаходиться за адресою бульвар Данила Галицького, 5, Тернопіль. В ньому торгують продуктами.

Розробка веб-сторінки продуктового магазину призначена для продажу продуктів, через інтернет. Наявність інтернет-магазину дозволить покупцям вибирати та оплачувати товари ще до приходу в магазин. Після отримання замовлення, персонал збирає необхідні продукти і видає їх покупцю без черги, що дозволить зробити процес покупки товарів набагато зручнішим та швидшим.

Також користувач зможе порівнювати продукцію та слідкувати за знижками та новими товарами чи послугами, які надає магазин через мережу інтернет. Також користувач зможе залишати та переглядати оцінку про товари. Це одночасно є перевагою як покупців, які зможуть оцінити якість певної продукції та переглянути загальну оцінку, так і для керівництва, так як воно зможе зробити висновки про популярність товару. Також на сайті буде розділ з новинами, де магазин зможе поширювати певну інформацію.

					ДП.КН.23.504.02.000 ПЗ	Арк.
						41
Зм.	Арк.	№ докум.	Підпис	Дата		

3.2 Вимоги до програмного забезпечення та документації

Загальні державні вимоги до веб-сторінок затверджені в українських (ДСТУ) та міжнародних (ISO/IEC/IEEE, RUP, SWEBOOK і т.д.) стандартах. В них прописані рекомендації щодо етапів розробки, написання технічного завдання, розробки програмного забезпечення та подальшого використання.

Вимоги до дипломного проєкту:

- Зручний інтерфейс.
- Відсутність критичних помилок.
- Правильний вхід та вихід даних.

Для написання сайту вибрано мову програмування Javascript і стек технологій MongoDB, Express.js, React.js, Node.js.

Після завершення розробки сайту, замовникові передається певний пакет документів, який містить основні відомості про структуру сайту, яка може допомогти в обслуговуванні сайту та усуненні помилок у роботі сайту.

Комплект цих документів в повинен включати:

- загальний опис структури системи і її функціональних можливостей;
- структуру зв'язків між сторінками та модулями веб-сайту;
- інструкції з налаштування необхідних установок для робочих станцій.

3.3 Техніко-економічні показники

Для розробки сайту для магазину Левас обрано наступні програмні і технічні засоби:

Програмні засоби:

1. Браузер: Firefox 91.0.
2. Пакетний менеджер: NPM v8.1.2.
3. Відправник запитів: Postman v9.20.0.
4. Visual Studio Code 1.68.0.

Технічні засоби:

					ДП.КН.23.504.02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

1. Операційна система: Windows 10.
2. Процесор: Intel Core I5 4570.
3. Відеокарта: Nvidia GTX 1050.
4. Оперативна пам'ять: Kingston 8GB DDR3-1666.

3.4 Стадії та етапи розробки

При розробці проєкту веб-сайту весь обсяг робіт можна розділити на декілька етапів, що складають основу процесу створення веб-проєкту. Кожен з цих етапів має свої особливості та вимоги, що потребують відповідального підходу з боку розробників та команди проєктування. Основні етапи розробки веб-сайту включають:

- 1) Визначення тематики та основної мети проєкту.
- 2) Розробка технічного завдання.
- 3) Написання коду сайту.
- 4) Наповнення сайту контентом.
- 5) Тестування сайту.
- 6) Здача готового проєкту.

Після завершення розробки проєкту інтернет-магазину створюється спеціальна комісія для перевірки дієздатності та ефективності даного сайту. Дана комісія повинна складатись із представників двох сторін: керівництва підприємства і представників розробника. Створена комісія має перевірити надійність та функціональність сайту, стійкість до несанкціонованого доступу, доцільність використання ресурсів, виділених на створення даного проєкту.

3.5 Тестування сайту

Перед встановленням сайту на хостинг необхідно провести ряд тестів, які допоможуть знайти та виправити помилки у дизайні, верстці чи зв'язку з

					ДП.КН.23.504.02.000 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		43

сервером. Проєкту буде дуже складно набрати популярність серед користувачів без цього.

Завдяки тестуванню можна перевірити правильність виконання технічного завдання, швидкість та зручність.

Функціональне тестування можна назвати найбільш тривалим, так як його процес зводиться до перевірки заявленого функціоналу:

- Перевірки роботи всіх обов'язкових функцій сайту.
- Тестування працездатності форм на сайті.
- Перевірка посилань, пошук неробочих посилань.
- Перевірка завантаження файлів на сервер.

Для перевірки були здійснені тестування всіх функцій, таких як додавання товару в кошик, замовлення, системи відгуків і т.д.

Також були протестовані форми реєстрації та входу.

Був проведений тест на наявність неробочих посилань, з використанням сервісу за адресою <http://www.brokenlinkcheck.com>. Сайт успішно пройшов цей тест(Рисунок 3.1).

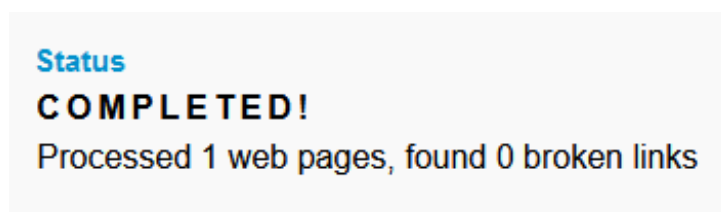


Рисунок 3.1 – Результат тесту на наявність неробочих посилань

Для перевірки зв'язку з сервером було здійснене додавання товару на сервер. Товар успішно добавився.

Для тестування верстки була здійснена перевірка адаптивності сайту. При проведенні тесту використовувались інструменти розробника браузера Firefox. Перевірялась сторінка з товарами у розширеннях 412x915 (Мобільна версія) та 810x1080 (Планшетна версія).

Як видно на рисунку 3.2, сайт пройшов перевірку адаптивності для мобільних телефонів.

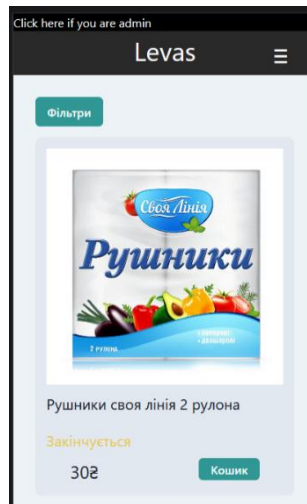


Рисунок 3.2 – Результат тесту верстки для телефонів

Сайт також успішно пройшов перевірку адаптивності для планшетів (Рисунок 3.3).

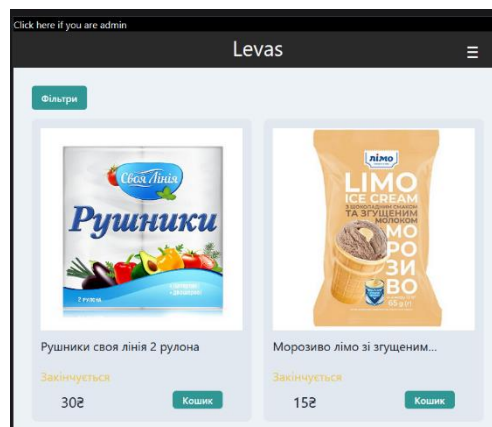


Рисунок 3.3 – Результат тесту верстки для планшетів

Тестування продуктивності проводиться для перевірки швидкодії сайту.

Було проведено два тести на швидкодію:

1) Тестування з використанням сервісу Gtmetrix (Рисунок 3.4).

					ДП.КН.23.504.02.000 ПЗ	Арк.
						45
Зм.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.4 – Тест Gtmetrix

В цьому тесті сайт отримав хорошу оцінку.

2) Тестування з використанням сервісу Pagespeed (Рисунок 3.5).

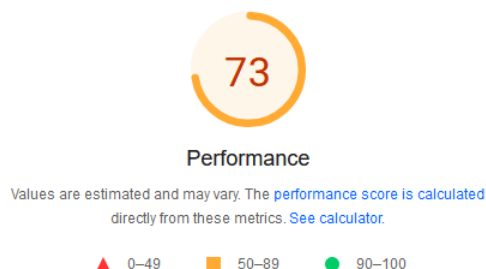


Рисунок 3.5 – Тест Pagespeed

За результатами цього тестування сайт теж отримав хорошу оцінку.

Інсталяція програмного забезпечення, необхідного для розробки проекту здійснюється наступними кроками:

1. Переходимо на офіційний сайт Visual Studio Code, вибираємо свою операційну систему та завантажуємо інсталятор. Сайт Visual Studio Code зображено на рисунку 3.6.

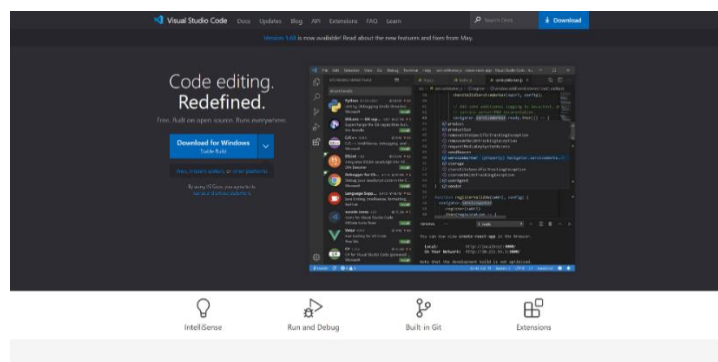


Рисунок 3.6 – Сайт Visual Studio Code

2. Переходимо на офіційний сайт NPM (Node Package Manager) та завантажуємо інсталятор. Сайт NPM зображено на рисунку 3.7.

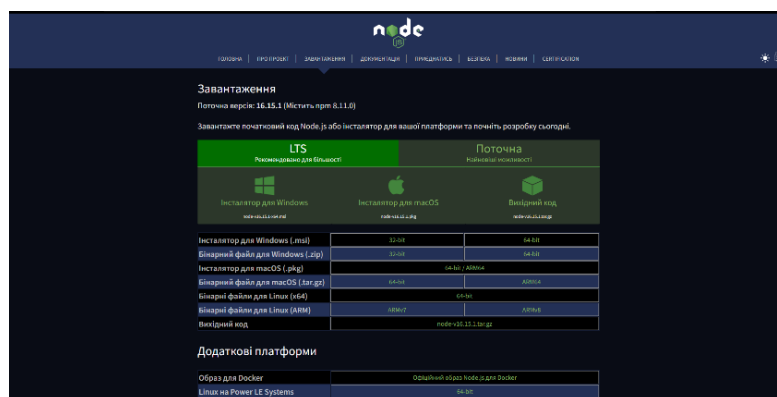


Рисунок 3.7 – Сайт NPM

Перевіряємо встановлення в командному рядку командами, зображеними на рисунку 3.8.

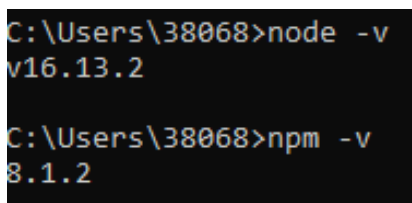


Рисунок 3.8 – Команди для перевірки встановлення

Якщо виводиться помилка, то вручну вписуємо NPM в шляхи ОС, так як зображено на рисунку 3.9.

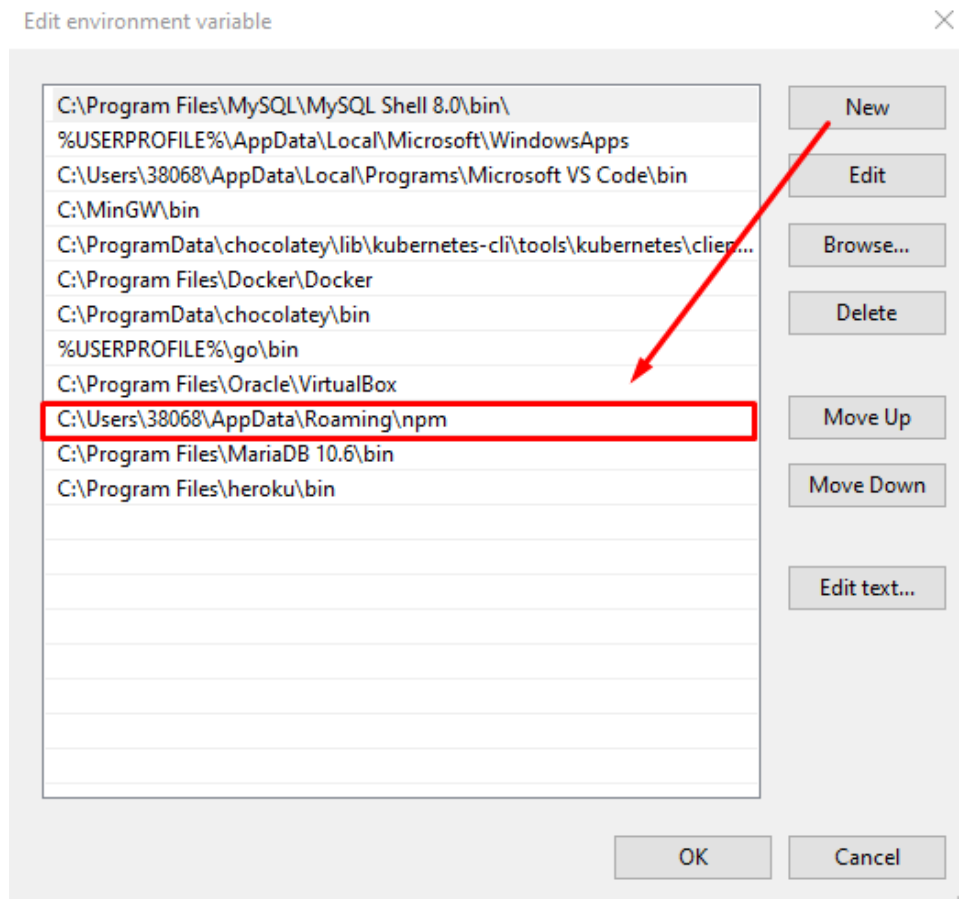


Рисунок 3.9 – Приклад наявності NPM в шляхах ОС

3. В папку з проектом необхідно встановити певні пакети, використовуючи NPM. Для цього в командному рядку вводимо команду `npm i < назва пакета >`. Приклад встановлення пакунків зображено на рисунку 3.10.

```
C:\Users\38068\Desktop\test2>npm i nodemon
up to date, audited 210 packages in 4s
27 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
```

Рисунок 3.10 – Приклад встановлення пакетів

Повний список пакетів, необхідних для розробки:

- Bcrypt.
- Cloudinary.

- Cors.
- Dotenv.
- Express.
- Jsonwebtoken.
- Multer.
- Mongoose.
- Nodemon.
- Path.
- Uuid.
- @chakra-ui/icons.
- @chakra-ui/react.
- Axios.
- cyrillic-to-translit-js.
- framer-motion.
- Iconv.
- iconv-lite.
- jwt-decode.
- Mobx.
- mobx-persist-store.
- mobx-react-lite.
- node-sass.
- raw-body.
- react-dropzone.
- react-image-crop.
- Sass.

Для хосту сайту буде використовуватись сервіс Нероки. Цей сервіс відомий своєю зручністю, простотою та швидкою технічною підтримкою.

Для того, щоб заhostити сайт необхідно слідувати наступним крокам:

					ДП.КН.23.504.02.000 ПЗ	Арк.
						49
Зм.	Арк.	№ докум.	Підпис	Дата		

1) Заходимо на клієнтську частину сайту, в змінну та виконуємо команду `npm run build` (Рисунок 3.11).

```
PS C:\Users\38068\Desktop\New folder (2)\DiplomLevas\client> npm run build
npm WARN config global '--global', '--local' are deprecated. Use '--location=global' instead.

> client@0.1.0 build
> react-scripts build

Creating an optimized production build...
```

Рисунок 3.11 – Виконання команди `npm run build`

Результатом виконання даної команди є створення папки `build`, де зберігається версія сайту, яка готова до хостингу.

2) Заходимо в файл `index.js` і пишемо в ньому код, продемонстрований у лістингу 3.1, та скачуємо необхідні пакети.

Лістинг 3.1 – Поля для колекцій задаються схемами

```
require('dotenv').config()
const express = require('express')
const cors = require('cors')
const mongoose = require('mongoose')
const cookieParser = require('cookie-parser')
const path = require('path');
const stripe = require("stripe")(process.env.stripe_private)
const router = require("./routers/index.js")
const errorHandler =
require("./middleware/errorHandlingrMiddleware")
const app = express()
const port = process.env.PORT || 8000;
app.use(cors());
app.options('*', cors());
app.use(express.json());
app.use(cookieParser());
app.use(express.static(path.join(__dirname,
"../../client/build")));
app.use("/api", router)
app.get("/*", (_, res) => { res.sendFile(path.join(__dirname,
"../../client/build/index.html")) });
app.use(errorHandler)
```

3) Ініціалізуємо репозиторій на Github.

					ДП.КН.23.504.02.000 ПЗ	Арк.
						50
Зм.	Арк.	№ докум.	Підпис	Дата		

4) Реєструємось на HeroKu, реєструємо акаунт та створюємо проєкт (Рисунок 3.12).

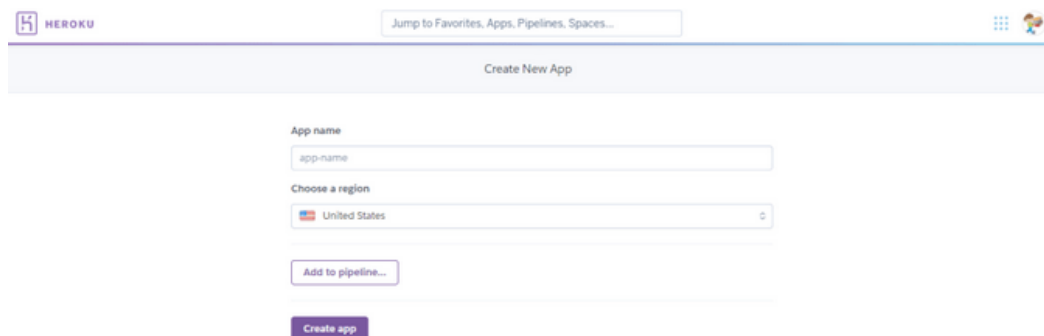


Рисунок 3.12 – Створення проєкту Heroku

5) Прив'язуємо Github, та на сторінці проєкту вибираємо репозиторій з проєктом.

6) Натискаємо кнопку “Deploy branch”, та чекаємо поки додаток завантажиться (Рисунок 3.13).

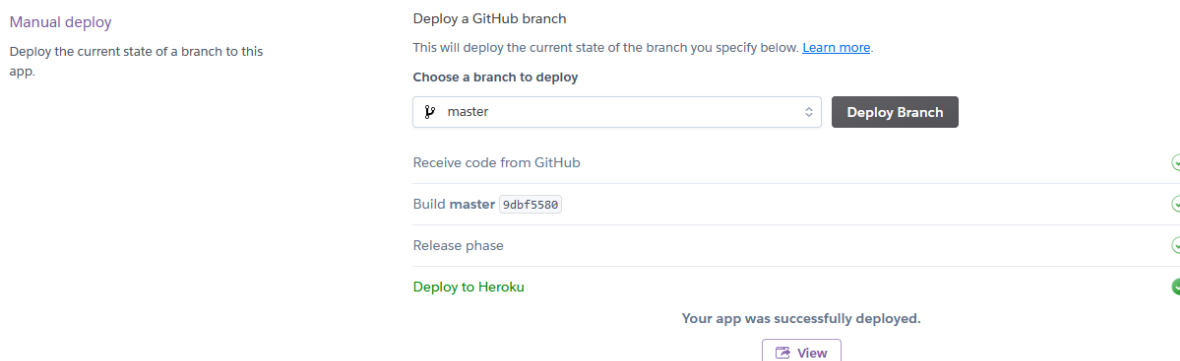


Рисунок 3.13 – Завантаження сайту на Heroku

7) Отримуємо наш сайт, при натисненні на кнопку “View”.

Перед здачею, кожен сайт повинен пройти тестування. Це невід’ємна частина розробки сайтів, так як вона дозволяє виявити та виправити помилки

					ДП.КН.23.504.02.000 ПЗ	Арк.
						51
Зм.	Арк.	№ докум.	Підпис	Дата		

в кодi. На даний момент є безліч способів тестування сайтiв, які можна згрупувати у 3 категорії:

- Тести інтерфейсу.
- Сервісні тести.
- Модульні тести.

Разом вони утворюють так звану “піраміду тестування”.

Вона була вперше представлена в книзі Майкла Кона “Scrum: Зручна розробка програмного забезпечення”. Піраміда виступає чудовою метафорою, яка наводить на думку, що тестування буває різнорівневе.

Сама “Піраміда тестування” зображена на рисунку 3.14.

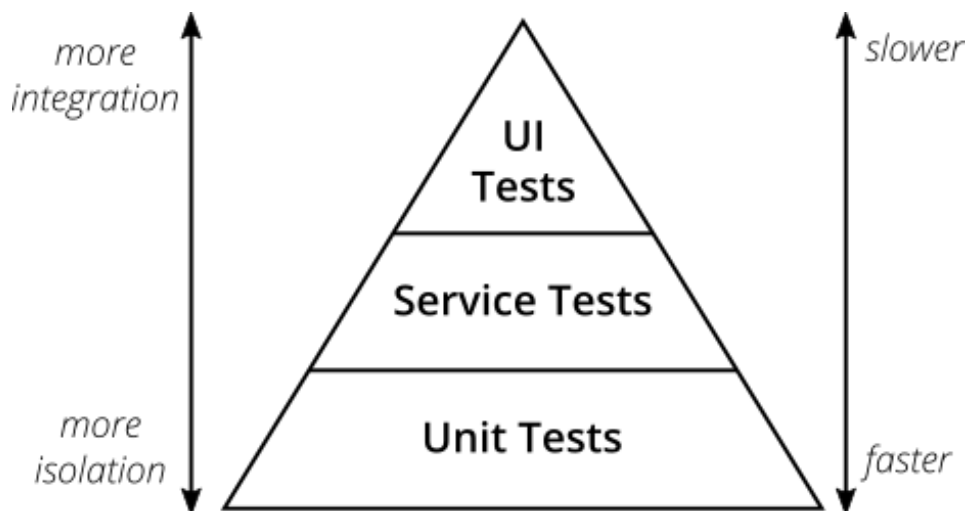


Рисунок 3.14 – “Піраміда тестування” Майкла Кона

Ширина кожного рівня піраміди показує скільки тестів має бути на кожному з них.

На найвищому рівні розташовуються автоматичні тести користувацького інтерфейсу. Їх повинно бути найменше.

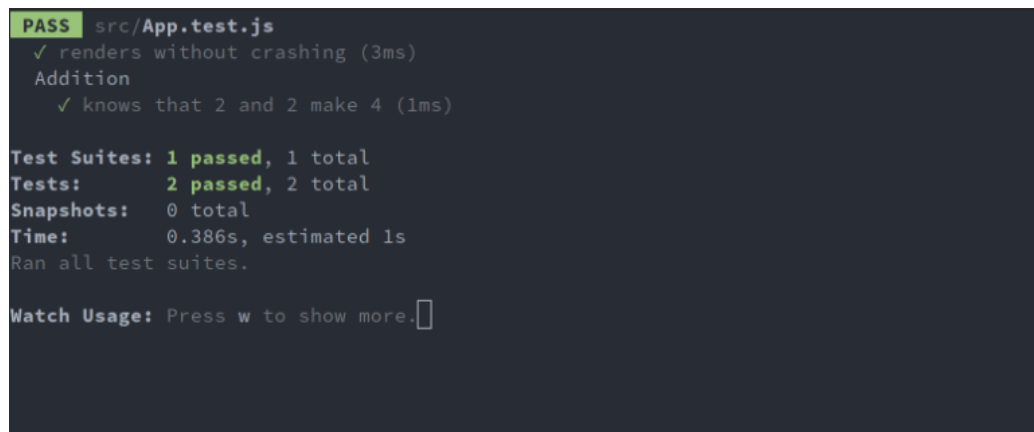
На середньому рівні розташовуються тести функціоналу сайту. Їх ще називають API-тестами. Таких тестів деколи потрібно багато, але завжди менше чим юніт-тестів.

На найнижчому рівні розташовуються модульні тести. Їх кількість завжди найбільша, так як тестується кожен модуль.

Для тестування було вирішено провести модульні тести, так як вони проводяться легко та швидко. Бібліотека React, яка була використана для написання клієнтської частини сайту, пропонує утиліту для модульного тестування Jest.

Jest – утиліта для тестування коду JavaScript, розроблена компанією Meta (Facebook). Її не потрібно додатково завантажувати, так як вона входить у вбудовані компоненти React. Для тестування утилітою Jest необхідно створити новий файл типу <Назва компонента>.test.js.

Для прикладу, протестуємо компонент App.js. Результат зображено на рисунку 3.15.



```
PASS src/App.test.js
✓ renders without crashing (3ms)
  Addition
    ✓ knows that 2 and 2 make 4 (1ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        0.386s, estimated 1s
Ran all test suites.

Watch Usage: Press w to show more.
```

Рисунок 3.15 – Результат тесту компонента App.js

Цей компонент пройшов тестування.

Суть модульного тестування полягає в перевірці кожного компоненту. Цього може бути недостатньо для покриття всього сайту, але для проведення високорівневого тестування необхідно звернутись до тестувальників.

В панелі адміністратора здійснюються такі модифікації як:

- Додавлення, видалення та редагування товару.
- Редагування та видалення користувачів.
- Додавлення категорій товару.
- Додавлення, редагування та видалення статей.

Для того щоб увійти в панель адміністратора нам необхідно натиснути на посиланні з текстом “Click here if you are admin” (Рисунок 3.16).

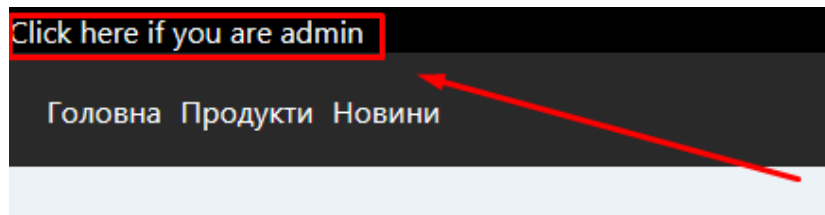


Рисунок 3.16– Посилання на адмін-панель

У разі, якщо користувач зазначений в базі даних як адміністратор, він перейде на адмін-панель у розділ продукти.

У розділі “Продукти” можна переглянути, створити та видалити товар та додати категорію товарів (Рисунок 3.17).

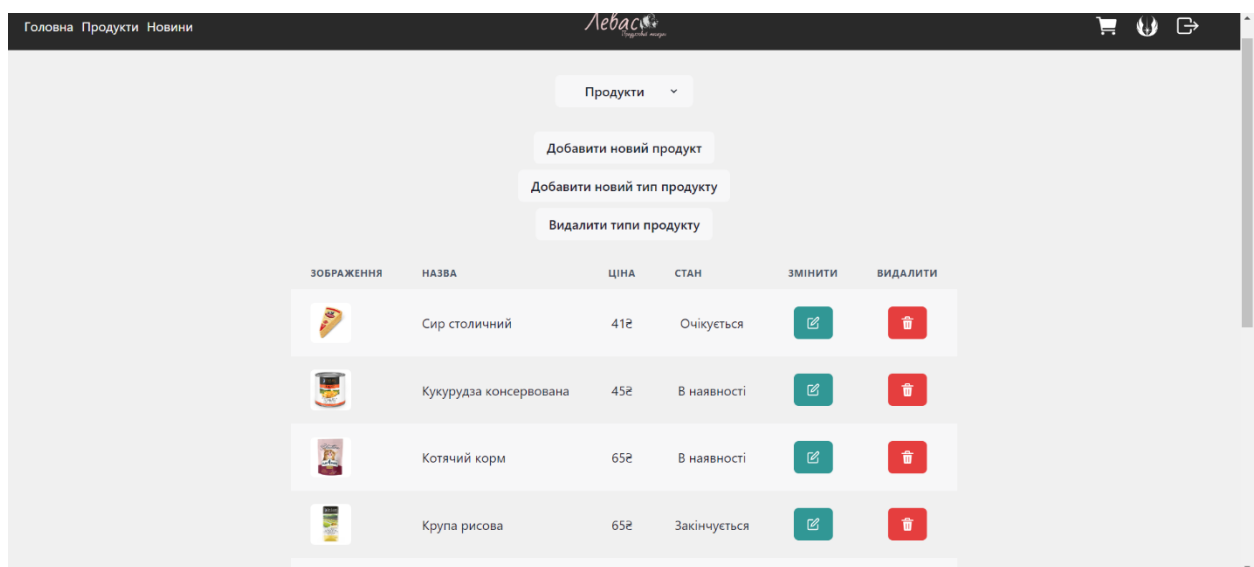



Рисунок 3.17 - Розділ “Продукти”

Додати товар можна натиснувши на кнопку “Додати новий продукт”. Висвітиться модальне вікно, у якому необхідно вказати дані товару і натиснути кнопку “Save” (Рисунок 3.18).

Фото

Browse... catalog_product_gal_1726.jpg



Назва

Вода "Полана Квасова" 1.5л

Ціна

35

Тип

Вода

Наявність на складі

В наявності

Властивості

Добавити властивості

Виробник

Своя лінія

Об'єм

1.5

Рисунок 3.18 – Додавання товару

Для перегляду та проведення дій з користувачами необхідно перейти в розділ “Користувачі” (Рисунок 3.19).

Click here if you are admin



Головна Продукти Новини

Levas

Кошик Вийти

Користувачі

Список користувачів

ІДЕНТИФІКАТОР	ІМ'Я	EMAIL	РОЛЬ	ВИДАЛИТИ
cd2d2e9a-6910-4ab9-90c6-191670ceb4c3	test	test@gmail.com	admin	
a82317c-af3b-4b73-ba72-bcd5d6f2d72a	Adam	test2@gmail.com	customer	
ІДЕНТИФІКАТОР	ІМ'Я	EMAIL	РОЛЬ	ВИДАЛИТИ

Список усіх користувачів

Рисунок 3.19 – Розділ “ Користувачі”

У цьому розділі можна поміняти роль користувача, міняючи значення селектора та видалити його натисканням кнопки.

У розділі “Замовлення” можна переглядати та змінювати статус замовлення, міняючи значення селектора (Рисунок 3.20).

Click here if you are admin

ГоловнаПродуктиНовини

Levas

Кошик...Вийти

Замовлення

ІДЕНТИФІКАТОР	ЦІНА	СТАН	ДАТА ТА ЧАС
26566d4b-2029-4119-a93a-e533fc480bb9	500₴	Відправлено	2022-06-10T16:01:09.940Z
0b159d4d-016c-4670-b83c-5b65545a3b19	500₴	В обробці	2022-06-10T16:01:20.084Z
8237e882-8280-4719-a3df-9a361e17c03c	5009₴	Очікує отримання	2022-06-10T16:09:43.000Z
ІДЕНТИФІКАТОР	ЦІНА	СТАН	ДАТА ТА ЧАС

Список усіх користувачів

Рисунок 3.20 – Розділ “Замовлення”

У розділі “Статті” можна додати, редагувати, та видалити статтю натисканням відповідних кнопок (Рисунок 3.21).





Click here if you are admin				
Головна		Продукти	Новини	Levas
Корзина... Вийти				
Статті				
Додати статтю				
ІДЕНТИФІКАТОР	ЗАГОЛОВОК	УРІВОК ТЕКСТУ	ЗМІНИТИ	ВИДАЛИТИ
921293bc-a870-4c7b-9d13-3720a1ecb7b6	2132132	zxczcdddd		
f58cbcd8-f8dc-4af3-8f5c-4467309195d0	Test	Test		
ІДЕНТИФІКАТОР	ЗАГОЛОВОК	УРІВОК ТЕКСТУ	ЗМІНИТИ	ВИДАЛИТИ
Список усіх користувачів				

Рисунок 3.21 – Розділ “Статті”

4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ

Темою даного дипломного проєкту є “Розробка проєкту веб-сайту для продуктового магазину “Левас”. Мета економічної частини в дипломному проєкті - це здійснення економічних розрахунків, спрямованих на визначення економічної ефективності розробки сайту і прийняття рішення про його подальший розвиток і впровадження або ж недоцільність проведення відповідної роботи.

Об’єктом розробки є сайт.

Розрахунок вартості розробки виконується в декілька етапів:

- Опис технологічного процесу розробки із зазначенням трудомісткості кожної операції;
- визначення суми витрат на оплату праці основного і допоміжного персоналу, включаючи відрахування на соціальні заходи;
- визначення суми матеріальних затрат;
- обчислення витрат на електроенергію для науково-виробничих цілей;
- розрахунок транспортних витрат;
- нарахування суми амортизаційних відрахувань;
- визначення суми накладних витрат;
- складення кошторису та визначення собівартості робіт;
- розрахування ціни робіт;
- визначення економічної ефективності та терміну окупності.

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення робіт

Базуючись на даних сформованого технічного завдання виділено 9 стадій розробки сайту.

					ДП.КН.23.504.02.000 ПЗ	Арк.
						57
Зм.	Арк.	№ докум.	Підпис	Дата		

Дані про витрати часу на проведення робіт наведені в таблиці 4.1.

Таблиця 4.1 – Середній час виконання робіт по обслуговуванню та стадії (операції) технологічного процесу.

№ п/п	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
1.	Постановка задачі та формування технічного завдання	Керівник проєкту	4
2.	Розробка дизайну сайту	Програміст	7
3.	Затвердження дизайну	Керівник проєкту	2
4.	Розробка frontend та backend	Програміст	30
5.	Тестування сайту	Лаборант	2
6.	Документування	Лаборант	8
7.	Здача сайту	Програміст	3
		Керівник проєкту	3
Разом		-	59

Сумарний час виконання операцій технічного процесу становить 59 годин.

4.2 Визначення витрат на оплату праці та відрахувань на соціальні заходи

Заробітна плата, відповідно до Закону України “Про оплату праці”, – це “винагорода, обчислена, як правило, у грошовому виразі.

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та господарської діяльності підприємства. Заробітна плата складається з основної та додаткової оплати праці.

Основна заробітна плата нараховується на виконану роботу за тарифними ставками, відрядними розцінками чи посадовими окладами і не залежить від результатів господарської діяльності підприємства.

Додаткова заробітна плата – це складова заробітної плати працівників, до якої включають витрати на оплату праці, не пов'язані з виплатами за фактично відпрацьований час. Нараховують додаткову заробітну плату залежно від досягнутих і запланованих показників, умов виробництва, кваліфікації виконавців. Джерелом додаткової оплати праці є фонд матеріального стимулювання, який створюється за рахунок прибутку.

Основна заробітна плата розраховується за формулою:

$$Z_{осн.} = T_c \cdot K_z, \quad (4.1)$$

де: T_c – тарифна ставка, грн.;

K_z – кількість відпрацьованих годин.

Отже основна заробітна плата для:

- Керівника проекту $Z_{осн1} = 100 \cdot 9 = 900$ грн.
- програміст $Z_{осн2} = 200 \cdot 40 = 8000$ грн.
- лаборанта $Z_{осн3} = 70 \cdot 10 = 700$ грн.

Сумарна основна заробітна плата становить:

$$Z_{осн} = 900 + 8000 + 700 = 9800 \text{ грн.}$$

Додаткова заробітна плата становить 10 – 15 % від суми основної заробітної плати.

$$Z_{дод.} = Z_{осн.} \cdot K_{додл.}, \quad (4.2)$$

де $K_{додл.}$ – коефіцієнт додаткових виплат працівникам, 0,1–0,15.

Отже додаткова заробітна плата по категоріях працівників становить:

- Керівника проекту $Z_{дод1} = 900 \cdot 0,15 = 135$ грн.

					ДП.КН.23.504.02.000 ПЗ	Арк.
						59
Зм.	Арк.	№ докум.	Підпис	Дата		

- інженера $З_{\text{доп2}} = 8000 * 0,15 = 1200$ грн.
- лаборанта $З_{\text{доп3}} = 700 * 0,15 = 105$ грн.

Загальна додаткова заробітна плата становить:

$$З_{\text{доп}} = 135 + 1200 + 105 = 1440 \text{ грн.}$$

Звідси загальні витрати на оплату праці ($V_{\text{о.п.}}$) визначаються за формулою:

$$V_{\text{о.п.}} = З_{\text{осн.}} + З_{\text{доп.}}, \quad (4.3)$$

$$V_{\text{о.п.}} = 9800 + 1400 = 11200 \text{ грн}$$

Крім того, слід визначити відрахування на соціальні заходи:

- єдиний соціальний внесок – 22%

Отже, сума відрахувань на соціальні заходи буде становити:

$$V_{\text{с.з.}} = \text{ФОП} \cdot 0,22, \quad (4.4)$$

де ФОП – фонд оплати праці, грн.

$$V_{\text{с.з.}} = 11\,200 * 0,22 = 2464 \text{ грн.}$$

Проведені розрахунки витрат на оплату праці зведемо у таблицю 4.2.

Таблиця 4.2 - Зведені розрахунки витрат на оплату праці

№ п/п	Категорія працівників	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	Нарахув. на ФОП, грн.	Всього витрати на оплату праці, грн. $6=3+4+5$
		Тарифна ставка, грн.	К-сть від- працьов. год.	Фактично нарах. з/пл., грн.			
1	Кер.проекту	100	9	900	135	-	-
2	Програміст	200	40	8000	1200	-	-
3	Лаборант	70	10	700	105	-	-
Разом				9800	1400	2464	13664

Отже, загальні витрати на оплату праці становлять 13664 грн.

4.3 Розрахунок витрат на електроенергію

Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$Z_e = W \cdot T \cdot S, \quad (4.5)$$

де: W – необхідна потужність, кВт;

T – кількість годин роботи обладнання;

S – вартість кіловат-години електроенергії.

Час роботи ПК над даним проектом становить 59 годин, споживана потужність - 0.5 кВт/год.

$$Z_e = 0.5 \cdot 59 \cdot 1,68 = 49,56 \text{ грн.}$$

Загальні витрати на електроенергію становлять $Z_e = 49,56$ грн.

4.4 Розрахунок суми амортизаційних відрахувань

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення.

Амортизація на комп'ютери та оргтехніку нараховується лише в випадку, якщо їх вартість перевищує 2500 грн., а мінімально допустимі строки їх корисного використання 2 роки.

Для визначення амортизаційних відрахувань застосовуємо формулу:

$$A = \frac{B_B \cdot H_A}{100\%}, \quad (4.6)$$

де: A – амортизаційні відрахування за звітний період, грн.;

B_B – балансова вартість групи основних фондів на початок звітного періоду, грн.;

H_A – норма амортизації, %.

					ДП.КН.23.504.02.000 ПЗ	Арк.
						61
Зм.	Арк.	№ докум.	Підпис	Дата		

Враховуючи, що ПК працює над даним проєктом 61 год.,

$$A_{\text{ПК}} = \frac{7600 \cdot 0,04}{150} \cdot 59 = 119,57 \text{ грн.}$$

4.5 Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 20 – 60 % від суми основної та додаткової заробітної плати працівників.

$$H_v = B_{o.l.} \cdot 0,2 \dots 0,6, \quad (4.7)$$

де H_v – накладні витрати.

$$H_v = 11200 \cdot 0,3 = 3360 \text{ грн.}$$

4.6 Складання кошторису витрат та визначення собівартості робіт

Результати проведених розрахунків витрат зведемо у таблицю 4.4.

Таблиця 4.3 - Кошторис витрат.

Зміст витрат	Сума, грн.	В % до загальної суми
Витрати на оплату праці (основну і додаткову заробітню плату)	11200	65,14
Відрахунки на соціальні заходи	2464	14,33
Витрати на електроенергію	49,56	0,28
Амортизаційні відрахування	119,57	0,69
Накладні витрат	3360	19,54
Собівартість	17193,13	100

Собівартість (C_B) НДР розраховуємо за формулою:

$$C_B = B_{o.n.} + B_{c.z.} + Z_{m.v.} + Z_e + T_e + A + H_e, \quad (4.8)$$

Отже, собівартість дорівнює $C_B = 17193,13$ грн.

4.7 Розрахунок ціни робіт

Ціну робіт можна визначити за формулою:

$$Ц = C_B \cdot (1 + P_{рен}) + K \cdot B_{и.и.} \cdot (1 + ПДВ) \cdot K \quad (4.9)$$

де: $P_{рен}$ – рівень рентабельності, 30 %; K – кількість замовлень, од.;

$B_{и.и.}$ – вартість носія інформації, грн.;

ПДВ – ставка податку на додану вартість, (20 %).

$$Ц = 17193,13 \cdot (1 + 0,3) \cdot (1 + 0,2) = 26821,28$$

4.8 Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Для визначення ефективності продукту розраховують чисту теперішню вартість, можна визначити за формулою 4.13 та термін окупності ($T_{ок}$), який можна визначити за формулою 4.14.

					ДП.КН.23.504.02.000 ПЗ	Арк.
						63
Зм.	Арк.	№ докум.	Підпис	Дата		

$$ЧТВ = -K_B + \sum_{i=1}^t \frac{\Gamma_{\Pi}}{(1+i)^t} \quad (4.10)$$

де K_B – затрати на проєкт;

Γ_{Π} – грошовий потік за t – ий рік;

t – відповідний рік проєкту;

i – величина дисконтної ставки (10...15%).

Якщо $ЧТВ \geq 0$, то проєкт може бути рекомендований до впровадження.

$$ЧТВ = -17193,13 + \frac{8100}{(1+0,1)} + \frac{8100}{(1+0,1)^2} + \frac{8100}{(1+0,1)^3} = 3559,06 \text{ грн}$$

Термін окупності визначається за формулою:

$$T_{OK} = T_{ПВ} + \frac{H_B}{\Gamma_{ПР}}, \quad (4.11)$$

де $T_{ПВ}$ – період до повного відшкодування витрат, років;

H_B – невідшкодовані витрати на початок року, грн.;

$\Gamma_{ПР}$ – грошовий потік на початок року, грн.

$$T_{OK} = 2 + 3135,1/8100 = 2,4$$

Всі дані розрахунків внесемо в зведену таблицю 4.4 техніко-економічних показників.

4.9 Висновки

Таблиця 4.4 - Техніко-економічні показники розробки програми

№ п/п	Показник	Значення
1.	Собівартість, грн.	17193,13
2.	Плановий прибуток, грн.	8100
3.	Ціна, грн.	26821,28
4.	Чиста теперішня вартість, грн.	3559,06
5.	Термін окупності, рік	2,4

В економічному розділі описано фінансову доцільність проведення науково-дослідних робіт.

В результаті проведених розрахунків можна зробити висновок: оскільки собівартість даного проекту становитиме 22017,36 грн. при очікуваному плановому прибутку 9250 грн. то термін окупності даного сайту складатиме більше ніж два роки.

					ДП.КН.23.504.02.000 ПЗ	Арк.
						65
Зм.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У цьому дипломному проєкті було розроблено сайт для продуктового магазину “Левас”. Цей сайт орієнтований для широкого застосування в мережі Інтернет. Завдяки йому, користувачі зможуть переглядати товари, дізнатись про акції та робити покупки через мережу інтернет.

У процесі розробки був проведений аналіз сучасних веб-технологій, з допомогою яких розробляють веб-сайти.

Для стилізації сайту була використана бібліотека Chakra, яка спростила процес написання інтерфейсу та допомогла забезпечити адаптивність сайту.

Структура веб-сайт для магазину «Левас» побудована використовуючи мову програмування JavaScript, а саме бібліотеку React, яка була використана для написання інтерфейсу сайту та взаємодії з користувачем на клієнтській частині, та середовища Node.js яка забезпечила обробку запитів та взаємодію з базою даних.

Дипломний проєкт містить економічну частину, де розраховується вартість написання даного сайту, а також розділ, де описуються питання з охорони праці, техніки безпеки та екологічних вимог.

Економічна доцільність розробки та впровадження даного сайту підтверджена розрахунками, проведеними в економічному розділі.

Подальше вдосконалення веб-сайту можливе у вигляді доопрацювання дизайну, та добавлення нового функціоналу

					ДП.КН.23.504.02.000 ПЗ	Арк.
						66
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Вступ в HTML [Електронний ресурс] – Режим доступу до ресурсу: https://www.w3schools.com/html/html_intro.asp – (Дата звернення: 13.04.2023р.)
2. Основні поняття мови HTML та структура документів [Електронний ресурс] – Режим доступу до ресурсу: <https://sites.google.com/site/vivcaemowebdizajndistancijno/html/lekcija-3-osnovni-ponatta-movi-html-ta-struktura-dokumentiv> – (Дата звернення: 13.04.2023р.)
3. Застосування CSS-технології в сайті, основні поняття [Електронний ресурс] – Режим доступу до ресурсу: http://antibotan.com/file.html?work_id=529477 – (Дата звернення: 13.04.2023р.)
4. Керівництво по SASS. Як верстати сайти вдвічі швидше? [Електронний ресурс] – Режим доступу до ресурсу: <https://tokar.ua/read/6672> – Дата доступу: 13.04.2023р.
5. SASS [Електронний ресурс] – Режим доступу до ресурсу: <https://sass-lang.com/> – (Дата звернення: 13.04.2023р.)
6. Поняття про мови веб-скриптів способи їх використання [Електронний ресурс] – Режим доступу до ресурсу: http://comscienceatschool.blogspot.com/p/26_17.html - (Дата звернення: 13.04.2023р.)
7. React [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.reactjs.org/> – (Дата звернення: 13.04.2023р.)
8. ReactJS курс [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tutorialspoint.com/reactjs/index.htm> – (Дата звернення: 13.04.2023р.)
9. Vue.js | вступ та інсталяція [Електронний ресурс] – Режим доступу до ресурсу: <https://www.geeksforgeeks.org/vue-js-introduction-installation/> – (Дата звернення: 13.04.2023р.)
10. Документація до Node.js [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.org/uk/docs/> – (Дата звернення: 13.04.2023р.)

					ДП.КН.23.504.02.000 ПЗ	Арк.
						67
Зм.	Арк.	№ докум.	Підпис	Дата		

11. MongoDB [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/> – (Дата звернення: 13.04.2023р.)

12. Види тестування [Електронний ресурс] – Режим доступу до ресурсу: <https://redstone.media/> – (Дата звернення: 13.04.2023р.)

13. Піраміда тестів на практиці [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/post/358950/> – (Дата звернення: 13.04.2023р.)

14. Jest [Електронний ресурс] – Режим доступу до ресурсу: <https://jestjs.io/uk/> – (Дата звернення: 13.04.2023р.)

					ДП.КН.23.504.02.000 ПЗ	Арк.
						68
Зм.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

Додаток А. Текст скрипта для перевірки стану авторизації користувача

```
/*Імпорт бібліотек та необхідних файлів*/
const jwt = require('jsonwebtoken');
const err = require("../errors/err")
const user = require("../models/userModel")

module.exports = async function checkAuth(req, res, next) {
  try {
    /*Перевірка чи метод запиту не OPTIONS */
    if (req.method === "OPTIONS") {
      next()
    }

    /*Токен дістається з заголовку та присвоюється змінній*/
    const token = req.header('Authorization')?.replace('Bearer ', ' ').trim();

    /*Якщо токена немає, тоді користувач отримує помилку*/
    if (!token) {
      return next(err.unauthorized("Unauthorized1"));
    }

    /*Розшифрування токена*/
    const decoded = jwt.verify(token, process.env.secret)

    /*Пошук його даних у базі даних*/
    const candidate = await user.findOne({ "id": decoded.id, "name":
    decoded.name, "email": decoded.email, "role": decoded.role }).exec()

    /*Якщо не знайдено, тоді користувач отримує помилку */
    if (!candidate) {
      return next(err.unauthorized("Unauthorized2"));
    }
  }
}
```

					ДП.КН.23.504.02.000 ПЗ	Арк.
						69
Зм.	Арк.	№ докум.	Підпис	Дата		

```

    }
    /*Якщо користувача знайдено, тоді дані передаються в наступну
функцію */
    req.role = candidate.role;
    req.user = candidate;
    next();
  } catch (e) {
    /*Якщо виникає якась непередбачувана ситуація, тоді користувач
отримує помилку, яка додатково виводиться в консоль*/
    console.log(e);
    return next(err.unauthorized("Unauthorized3"));
  }
}

```

					ДП.КН.23.504.02.000 ПЗ	Арк.
						70
Зм.	Арк.	№ докум.	Підпис	Дата		

РЕЦЕНЗІЯ
на дипломний проєкт
студента відділення комп'ютерних технологій
Галицького фахового коледжу імені В'ячеслава Чорновола

IV курсу групи КН-41

Дживри Назара Богдановича
(прізвище та ініціали)

Спеціальність 122 „Комп'ютерні науки”

Обсяг дипломного проєкту:

сторінок – 69, рисунків - 51, таблиць - 4, джерел - 14, додатків– 1.

Тема: «Веб-сайт для продуктового магазину “Левас”»

1. Актуальність теми: популяризація магазину та поширення відомостей про нього в багато чому залежить від інформаційної присутності в глобальній мережі та якості подання цієї інформації, тим самим досягається збільшення цільової аудиторії

2. Практична або теоретична цінність опрацьованих питань розроблений сайт створений відповідно до поставленого завдання та з урахуванням сучасних вимог створення сайтів подібної тематики, особливістю розробленого сайту є адаптивність, тобто можливість відображення на мобільних пристроях та планшетах.

3. Недоліки роботи

до недоліків можна віднести слабе посилання в тексті пояснювальної записки на використані джерела, в окремих моментах не дотримано вимог до оформлення та зустрічаються стилістичні неточності, що в загальному не позначається на функціонуванні реалізованого сайту та проєкту в цілому

4. Загальний висновок _____

дипломний проєкт виконаний згідно вимог і заслуговує на позитивну оцінку, а студент – присвоєння кваліфікації “техніка-програміста”

Рецензент О.Б. Посвятовська (прізвище та ініціали рецензента)

«__» __06__ 2023р. _____ -

ВІДГУК
дипломний проект
відділення комп'ютерних технологій
Галицького фахового коледжу імені В'ячеслава Чорновола

студента IV курсу групи КН-41

Джіври Назара
(прізвище та ініціали)

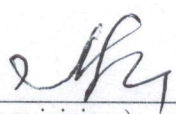
Спеціальність 122 „Комп'ютерні науки”

Керівник ДП Чубей О.О.

Тема: Розробка веб-сайту продуктового магазину «Левас»

1. Загальна характеристика студента: *студент проявив себе, як старанний, відповідальний та наполегливий. На достатньому рівні володіє засобами проектування. Робота проводилась із дотриманням усіх відповідних термінів.*
2. Практична або теоретична цінність опрацьованих питань: *Робота полягала у проектуванні вебсайту продуктового магазину. Отож, тема є актуальною для маркетингу та бізнес-комерції. Детально проаналізовано проблематику та здійснено огляд існуючих рішень. Даний проект готовий до застосування за призначенням, оскільки є нескладним у використанні.*
3. Недоліки роботи: *робота не містить UML-діаграм, які візуалізують процеси та роботу досліджувальної системи.*
4. Загальний висновок: *дипломний проект виконано згідно вимог і заслуговує на позитивну оцінку.*

Керівник дипломного проекту


(прізвище та ініціали)

Ім'я користувача:
Василь Кузик

Дата перевірки:
16.06.2023 00:40:35 EEST

Дата звіту:
16.06.2023 00:58:37 EEST

ID перевірки:
1015620284

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100012366

Назва документа: Дживра

Кількість сторінок: 59 Кількість слів: 7707 Кількість символів: 57350 Розмір файлу: 9.12 MB ID файлу: 1015267429

17.5% Схожість

Найбільша схожість: 8.86% з джерелом з Бібліотеки (ID файлу: 1015267100)

14.4% Джерела з Інтернету

868

Сторінка 61

9.37% Джерела з Бібліотеки

18

Сторінка 67

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

2