

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ
ОДЕСЬКОГО НАЦІОНАЛЬНОГО ТЕХНОЛОГІЧНОГО
УНІВЕРСИТЕТУ»**

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-06

Дипломний проект
здобувача освіти денної форми навчання
РП.06.18.000.ДП

***Прокопчука Андрія
Михайловича***

**м. Одеса
2023 р.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОДЕСЬКОГО
НАЦІОНАЛЬНОГО ТЕХНОЛОГІЧНОГО УНІВЕРСИТЕТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-06

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи) на тему:

Розробка інтернет-магазину з продажу комп'ютерної техніки


Проектний матеріал складається з пояснювальної записки на 77 сторінках та графічного (презентаційного) матеріалу на 10 аркушах (слайдах).

Дипломник  (Прокопчук А.М.)

Керівник  (Шувалова І.О.)

Консультанти:

з економічної частини  (Копайгородська Т.Г.)

з охорони праці  (Чорновол Н.І.)

з дотримання вимог ЄСКД  (Петрашова В.І.)

старший консультант  (Кунуп Т.В.)

До захисту допущений

Голова циклової комісії  (Кривченко Ю.В.)

Завідувач відділення  (Скорнякова О.В.)

Захист «23» 06 2023 р.

Протокол ДКК № 2

Оцінка ДКК 4 (за бр.)

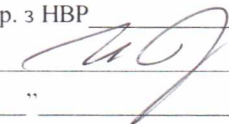
Секретар ДКК 

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНАХТ»

Відділення комп'ютерних систем Комісія КТ та ПІ
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Розробка програмного забезпечення»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР


“ ” 2023 р.

ЗАВДАННЯ

на дипломний проект (роботу)

Здобувачеві (здобувачці) освіти Прокопчука Андрія Михайловича
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка інтернет-магазину з продажу комп'ютерної техніки

затверджена наказом по коледжу від “ 17 ” жовтня 2022 р. № 235-A2-ОД

2. Термін здачі закінченого проекту (роботи) 09.серпня 2023

3. Вихідні данні до проекту (роботи) Microsoft Visual Studio, .Net, C#, MS SQL Server, JavaScript
LINQ, API, HTTP, Angular, CSS, Microsoft Edge, Postman, MySQL, СУБД

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

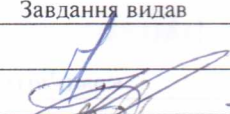
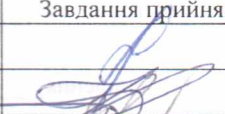
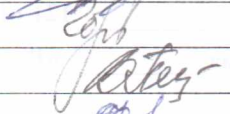
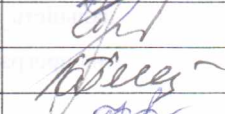
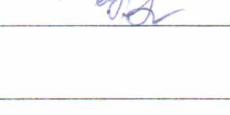
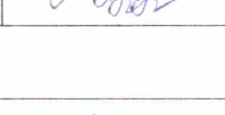


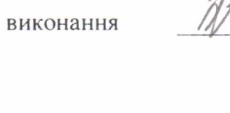
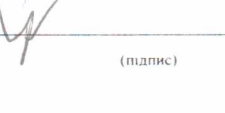
Вступ. 1. Технологічний розділ. 1.1. Поняття веб-розробки. 1.1.1. Інтернет та веб-сайти.
1.1.2 Огляд існуючих рішень. 1.1.3 Огляд структури веб-додатку. 1.1.4 Вибір складу
технічних і програмних засобів. 1.1.5. Огляд мов програмування для використання в
додатку. 1.2 Розробка веб-магазину. 2. Економічна частина 3. Охорона праці. Висновки

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)

Презентація (10 слайдів): 1. Вступний слайд. 2. Мета проекту. 3. Веб-сторінка та її складові. 4. Планування та побудова проекту. 5-10. Структура веб-додатку.


11. Висновки

6. Консультанти по проекту (роботі), із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Основний	Шувалова І.О.		
Економічний	Копайгородська Т.Г.		
Охорона праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		
Старший консультант	Кунуп Т.В.		

7. Дата видачі завдання 22.05.2023

Керівник


(підпис)

Завдання прийняв до виконання


(підпис)


КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1	Розділ 1. Технологічний розділ	22.05.2023	виконав
2	Розділ 2. Економічний розрахунок	22.05.2023	виконав
3	Розділ 3. Охорона праці	22.05.2023	виконав
4	Розробка презентації до дипломної роботи	25.05.2023	виконав
5	Чистове оформлення пояснювальної записки	27.05.2023	виконав
6	Підготовка доповіді до захисту	04.06.2023	виконав
7	Отримання рецензії, відповіді на зауваження Рецензента	05.06.2023	виконав
8	Захист роботи	09.06.2023	виконав

Дипломник


(підпис)

Керівник


(підпис)

ЗМІСТ

ВСТУП	7
1 ТЕХНОЛОГІЧНИЙ РОЗДІЛ.....	9
1.1 Поняття веб-розробки.....	9
1.1.1 Інтернет та веб-сайти.....	9
1.1.2 Огляд існуючих рішень	11
1.1.3 Огляд структури веб-додатку	13
1.1.4 Вибір складу технічних і програмних засобів	13
1.1.5 Огляд мов програмування для використання у додатку	15
1.2 Розробка інтернет-магазину	16
1.2.1 Розробка структури сайту	16
1.2.2 Розробка серверної частини сайту.	25
1.2.3 Розробка клієнтської частини сайту.	27
1.2.4 Розробка системи авторизації/реєстрації користувачів.....	29
1.2.5 Документація для користувача.....	30
2 ЕКОНОМІЧНА ЧАСТИНА	38
3 ОХОРОНА ПРАЦІ.....	46
3.1 Гігієнічні вимоги до виробничого середовища	46
3.1.1 Вимоги до приміщення.....	46
3.1.2 Вимоги до організації робочого місця користувача ПК.....	47
3.1.3 Освітлення робочого місця	49
3.1.4 Мікроклімат.....	50
3.2 Пожежна безпека	51
ВИСНОВКИ	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	55

					РП 06.18.000 ДП ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

ВСТУП

У сучасному світі технології впливають на майже кожен аспект нашого життя. Вони змінюють спосіб, яким ми спілкуємося, працюємо, навчаємося і, безумовно, як ми купуємо. Електронна комерція, або інтернет-торгівля, стала домінуючим методом покупки товарів і послуг для мільйонів людей по всьому світу. Вона надає зручність, широкий вибір і можливість порівняти ціни, що робить її незамінною для багатьох споживачів.

Одним з секторів, який активно використовує електронну комерцію, є ринок комп'ютерної техніки. Цей ринок характеризується високим рівнем конкуренції та постійним оновленням асортименту, що вимагає від інтернет-магазинів гнучкості, ефективності та високої якості обслуговування.

Відповідно до цих вимог, важливість розробки ефективних, зручних і надійних інтернет-магазинів зростає. Необхідність розробки таких ресурсів привела до того, що тема цієї дипломної роботи стала "Розробка інтернет-магазину з продажу комп'ютерної техніки".

Мета цієї роботи - розробити повноцінний інтернет-магазин для продажу комп'ютерної техніки, використовуючи сучасні технології та підходи. Для цього ми використовуватимемо ASP.NET для розробки серверної частини, Angular для клієнтської частини, а також HTML та CSS для розробки інтерфейсу користувача.

Для досягнення цієї мети необхідно:

1. Проаналізувати вимоги до функціональності та дизайну інтернет-магазину.
2. Розробити серверну частину за допомогою ASP.NET, що буде обробляти запити від клієнтів, керувати даними користувача та обробляти замовлення.
3. Розробити клієнтську частину за допомогою Angular, що буде відповідальною за інтерфейс користувача та взаємодію з сервером.

					РП 06.18.000 ДП ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

4. Створити інтерфейс користувача за допомогою HTML та CSS, включаючи головну сторінку, каталог товарів, сторінку товару, корзину та систему авторизації та реєстрації користувачів.

5. Реалізувати систему авторизації та реєстрації користувачів для забезпечення безпечного доступу до особистих даних та управління замовленнями.

Провести тестування всієї системи, щоб переконатися в її коректної роботі та відсутності помилок.

У рамках цієї роботи ми плануємо розробити основні сторінки інтернет-магазину, такі як головна сторінка, каталог товарів, сторінка товару, корзина, а також систему авторизації та реєстрації користувачів. Крім того, ми плануємо розробити серверну частину, яка буде обробляти запити від клієнта, керувати даними користувача і обробляти замовлення. Amigos сприяють успіху проекту.

					<i>РП 06.18.000 ДП ПЗ</i>	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

1 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

1.1 Поняття веб-розробки

1.1.1 Інтернет та веб-сайти

Сайт (від англ. Website: web - «павутина, мережа» і site - «місце», буквально «місце, сегмент, частина в мережі») - сукупність електронних документів (файлів) приватної особи або організації в комп'ютерній мережі, об'єднана під однією адресою (доменним ім'ям або IP-адресою). За замовчуванням мається на увазі, що сайт розташовується в мережі Інтернет.

Всі сайти в сукупності складають Всесвітню павутину, де комунікація (павутина) об'єднує сегменти інформації світової спільноти в єдине ціле - базу даних і комунікації планетарного масштабу. Для прямого доступу клієнтів до сайтів на серверах був спеціально розроблений протокол НТТР.

НТТР - протокол прикладного рівня передачі даних (спочатку у вигляді гіпертекстових документів). Основою НТТР є технологія «клієнт-сервер», тобто передбачається існування споживачів (клієнтів), які ініціюють з'єднання і надсилають запит, і постачальників (серверів), які очікують з'єднання для отримання запиту, виробляють необхідні дії і повертають назад повідомлення з результатом. НТТР в даний час повсюдно використовується у Всесвітній павутині для отримання інформації з веб-сайтів.

					<i>РП 06.18.001 ДП ПЗ</i>	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.1 – Структура мережі інтернет

Сторінки сайтів - це файли з текстом, розмічених мовою HTML. Ці файли, будучи завантаженими відвідувачем на його комп'ютер, обробляються браузером і виводяться на його засіб відображення (монітор, екран КПК, принтер). Мова HTML дозволяє формувати текст, розрізняти в ньому функціональні елементи, створювати гіпертекстові посилання (гіперпосилання) і вставляти в сторінку зображення, звукозаписи і інші мультимедійні елементи. Відображення сторінки можна змінити додаванням в неї таблиці стилів на мові CSS або сценаріїв на мові JavaScript.

Сторінки сайтів можуть бути простим статичним набором файлів або створюватися спеціальною комп'ютерною програмою на сервері - так званім движком сайту. Движок може бути або зроблений на замовлення для окремого сайту, або бути готовим продуктом, розрахованим на певний клас сайтів. Деякі з движків можуть забезпечити власнику сайту можливість гнучкої настройки структуризації і виведення інформації на веб-сайті. Такі двигуни називаються системами управління змістом (CMS).

1.1.2 Огляд існуючих рішень

Сучасні засоби проектування (СП) можна розділити на великі категорії. Першу складають CASE-системи (як незалежні (upper CASE), так і інтегровані з СУБД), що забезпечують проектування БД та додатків у комплексі з інтегрованими засобами розробки додатків "клієнт-сервер" (наприклад, Westmount I-CASE+Uniface, Designer/2000) +Developer/2000). Їхня основна перевага полягає в тому, що вони дозволяють розробляти всю ІС цілком (функціональні специфікації, логіку процесів, інтерфейс з користувачем і базу даних), залишаючись в одному технологічному середовищі. Інструменти цієї категорії, як правило, мають суттєву складність, широку сферу застосування та високу гнучкість.

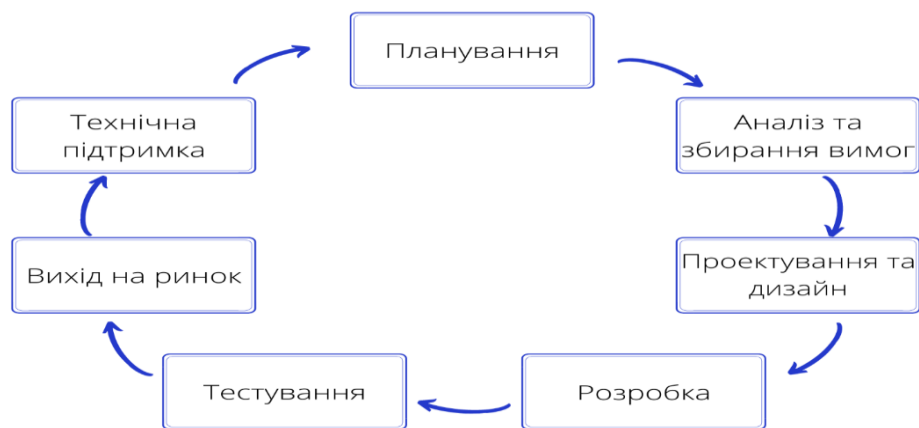


Рисунок 1.2 – Планування та побудова проекту для ринку.

Другу категорію складають власне кошти проектування БД, що реалізують ту чи іншу методологію, як правило, "сутність-зв'язок" ("entity-relationship") і розглядаються в комплексі із засобами розробки додатків. До засобів цієї категорії можна віднести такі, як SILVERRUN+JAM, ERwin/ERX+PowerBuilder та ін.

Крім зазначених категорій, СП можна класифікувати за такими ознаками:

- ступеня інтегрованості: (окремі локальні засоби, набір частково інтегрованих засобів, що охоплюють більшість етапів життєвого циклу ІС та повністю інтегровані засоби, пов'язані загальною базою проектних даних – репозиторієм);
- Застосовуваним методологіям і моделям систем та БД;
- Ступені інтегрованості з СУБД;
- Ступеня відкритості;
- Доступним платформам.

До розряду СП потрапляють як відносно дешеві системи для персональних комп'ютерів (ПК) з дуже обмеженими можливостями, так і дорогі системи для неоднорідних обчислювальних платформ та операційних середовищ. Так, сучасний ринок програмних засобів налічує близько 300 різних CASE-систем, найпотужніші з яких так чи інакше використовуються практично всіма провідними західними фірмами.

Застосування СП вимагає від потенційних користувачів спеціальної підготовки та навчання. Досвід показує, що впровадження СП здійснюється повільно, однак у міру набуття практичних навичок та загальної культури проектування ефективність застосування цих засобів різко зростає, причому найбільша потреба у використанні СП випробовується на початкових етапах розробки, а саме на етапах аналізу та специфікації вимог. Це тим, що помилок, допущених на початкових етапах, кілька порядків перевищує ціну помилок, виявлених більш пізніх етапах розробки.

На сьогоднішній день ринок програмного забезпечення має в своєму розпорядженні наступні найбільш розвинені СП.

Наведений перелік не претендує на повноту. Крім того, на ринку постійно з'являються нові (для вітчизняних користувачів) системи, так і нові версії, модифікації перерахованих систем

					<i>РП 06.18.001 ДП ПЗ</i>	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

1.1.3 Огляд структури веб-додатку

Структура інтернет-магазину складається з таких функціональних елементів:

- каталог товарів;
- користувальницький кошик;
- реєстраційна форма;
- форма відправлення замовлення.

Каталог товарів є складною і багаторівневою структурою даних, яка повинна простим і зрозумілим способом проводити впорядкування товарів. Найпростіше такий каталог подати у вигляді дерева об'єктів, верхній рівень якого складається зі списку розділів. Розділи можуть містити підрозділи чи посилання конкретний товар тощо. Таке впорядкування просто необхідне для зручного та швидкого пошуку та замовлення товарів.

Користувальницький кошик є деяким масивом даних, який служить для зберігання замовленого користувачем товару.

Реєстраційна форма служить введення персональних даних користувачів. Надалі ця інформація використовується для їхньої ідентифікації між сеансами роботи з інтернет-магазином. Ця інформація може зберігатись як на стороні сервера, так і на стороні клієнта.

1.1.4 Вибір складу технічних і програмних засобів

Для створення даного сайту будуть використані:

- мова гіпертекстової розмітки - HTML;
- мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки -CSS;
- бібліотека роботи с JavaScript-jQuery;
- серверна мова програмування – C#
- середовище програмування - Notepad ++;
- графічний растровий редактор - Adobe Photoshop;

					<i>РП 06.18.001 ДП ПЗ</i>	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

– використовується для тестування браузер - Google Chrome, Mozilla Firefox, Opera.

В якості мови гіпертекстової розмітки був обраний HTML -Стандартний мову розмітки документів в Інтернеті. Більшість веб-сторінок створюються за допомогою мови HTML. Мова HTML інтерпретується браузером і відображається у вигляді документа, в зручній для людини формі.

Підставою вибору HTML є його менша критичність до коректності структури сторінки, велика поширеність, і, отже, найкраща підтримка браузерами.

Для оформлення HTML сторінок використовується CSS - формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки. З його допомогою можна додати елементам сторінки абсолютно будь-який вид, що дозволяє зробити дизайн сайтаменее залежним від різного роду зображень.

Була обрана бібліотека роботи с JavaScript - jQuery - бібліотека JavaScript, що фокусується на взаємодії JavaScript і HTML. Бібліотека jQuery допомагає легко отримувати доступ до будь-якого елементу

Вибір середовища розробки в першу чергу повинен бути обумовлений зручністю написання коду, а також доступність програми. Notepad ++ - вільний текстовий редактор з відкритим вихідним кодом для Windows, з підсвічуванням синтаксису, великої кількості мов програмування і розмітки. Notepad ++ забезпечує наступний функціонал:

- підсвічування синтаксису;
- згортання коду;
- авто доповнення;
- закладки;
- регулярні вирази для пошуку і заміни;
- запис і відтворення макросів;
- порівняння файлів;

					<i>РП 06.18.001 ДП ПЗ</i>	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

- перевизначення будь-яких гарячих клавіш;
- резервне копіювання файлів, що зберігаються (включається в настройках);
- підтримка і конвертація кодувань ANSI, UTF-8 та UCS-2;
- блокове виділення тексту, одночасне виділення декількох різних місць (з CTRL);
- мульти рядкове редагування (з використанням Alt).

Переваги браузера Google Chrome - зручний і ненав'язливий інтерфейс.

Одне з головних переваг цього браузера - швидкість.

Форма відправлення замовлення служить для введення контактної інформації замовника та надсилання її та замовлення на електронну пошту організації.

1.1.5 Огляд мов програмування для використання у додатку

JavaScript - це мова написання сценаріїв, розроблена компанією Netscape Communications, для створення клієнтських та серверних додатків. Netscape Navigator створено для інтерпретації сценаріїв JavaScript, запроваджених у веб-сторінки.

JavaScript – об'єктно-орієнтована скриптова мова програмування. Є діалектом мови ECMAScript.

JavaScript зазвичай використовується як вбудована мова для програмного доступу до об'єктів додатків. Найширше застосування знаходить у браузерах як мову сценаріїв для надання інтерактивності веб-сторінок.

					<i>РП 06.18.001 ДП ПЗ</i>	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.3 – Рейтинг мов програмування у 2023 році

Основні архітектурні риси: динамічна типізація, слабка типізація, автоматичне керування пам'яттю, прототипне програмування, функції як об'єкти першого класу. Популярність у галузі побудови web-сайтів визначається наявністю великого набору вбудованих засобів для розробки web-додатків. Основні з них:

- автоматичне вилучення POST та GET-параметрів, а також змінних оточення web-сервера в певні масиви;
- файлові функції успішно обробляють як локальні, і віддалені файли;
- автоматичне відправлення HTTP-заголовків;
- робота з cookies та сесіями;
- обробка файлів, що завантажуються на сервер;
- робота з HTML заголовками та HTTP авторизацією.

1.2 Розробка інтернет-магазину

1.2.1 Розробка структури сайту

Головна сторінка будь-якого сайту - це основний елемент, на який перш за все звертають увагу користувачі. Тому при розробці макету головної сторінки важливо забезпечити зручний інтерфейс та привабливий вигляд.

					РП 06.18.001 ДП ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

Структура сайту зображена на рис 1.4

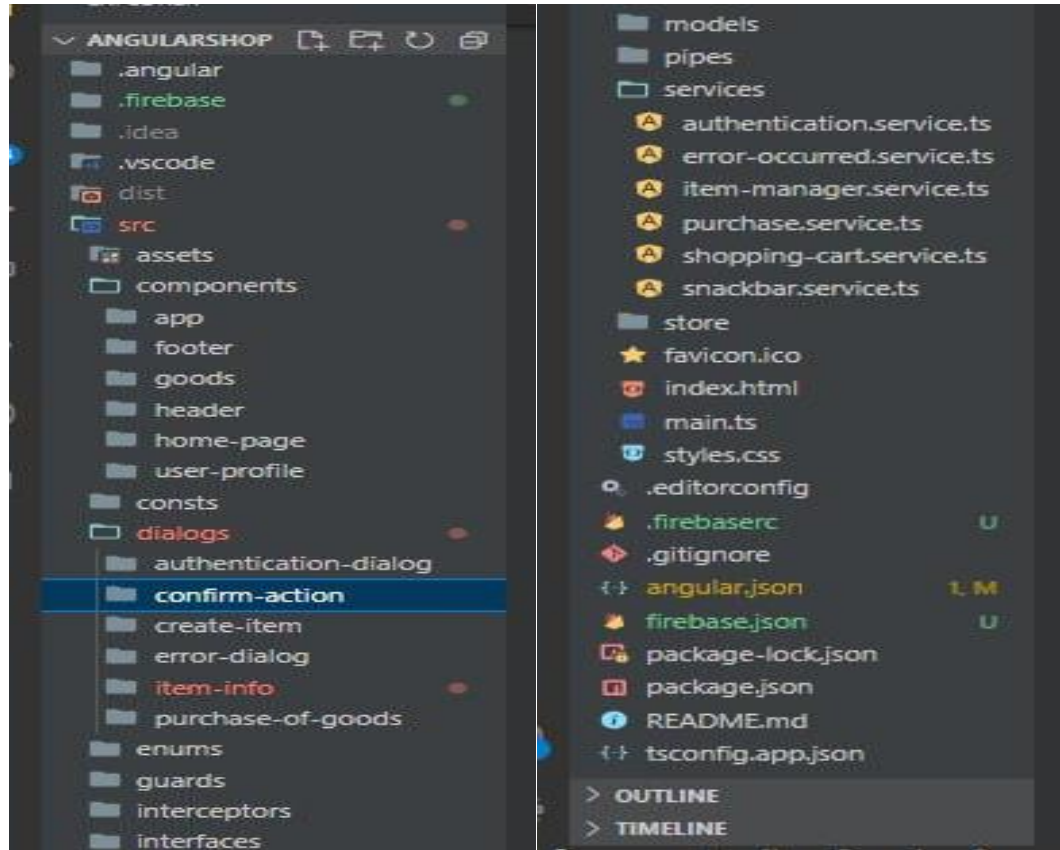


Рис 1.4 – Структура сайту

Структура бази даних зображена на рис. 1.5 та 1.6

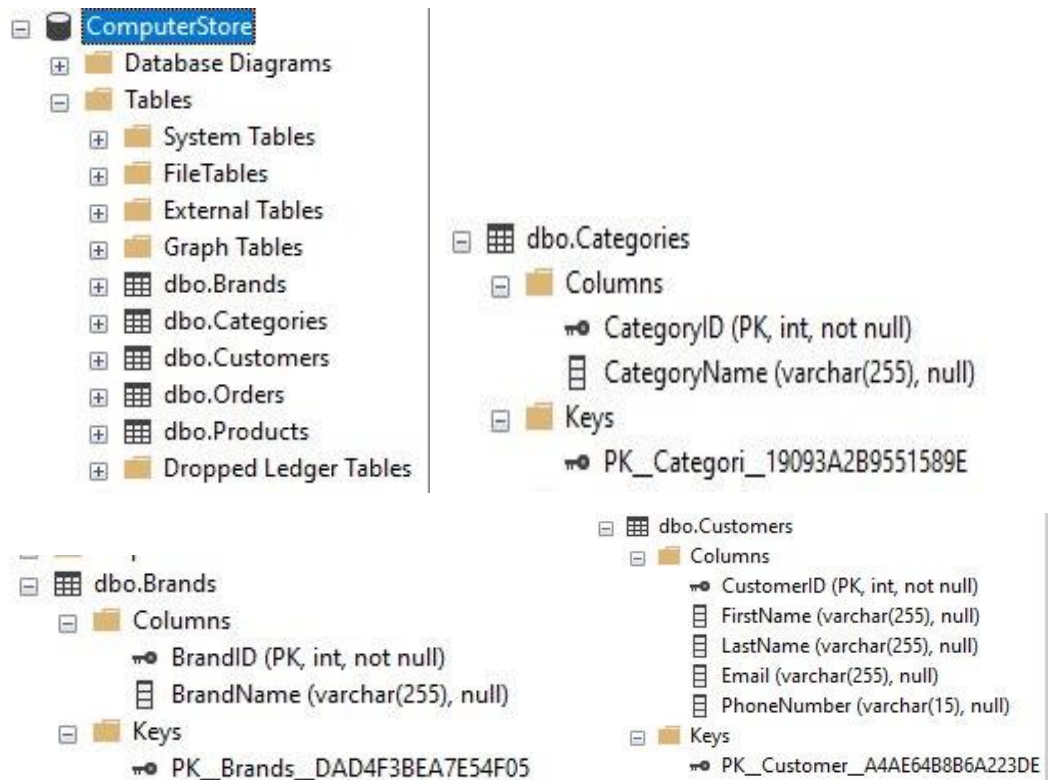


Рис. 1.5 – Структура БД



Рис 1.6 – Структура БД

Структура HTML-документа для головної сторінки може включати такі основні елементи: шапку сайту (logo, меню навігації), основний контент (наприклад, список популярних товарів або акцій), та підвал сайту (інформацію про компанію, контактні дані).

Стилізація сторінки виконується за допомогою CSS. Важливо врахувати, що дизайн має бути одночасно привабливим і функціональним, а також коректно відображатися на різних пристроях і в різних браузерах.

Нижче наведений приклад простого HTML-коду головної сторінки:

```

<!DOCTYPE html>
<html>
<head>
  <title>Назва інтернет-магазину</title>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
  <header>
    <h1>Logo</h1>
    <nav>
      <ul>
        <li><a href="#">Каталог</a></li>
        <li><a href="#">Контакти</a></li>
      </ul>
    </nav>
  </header>
  <main>
    <!-- Тут можуть бути різні блоки з товаром, акціями тощо -->
  </main>
  <footer>
    <p>© 2023, Назва інтернет-магазину</p>
  </footer>

```

Також важливо розробити адаптивний дизайн, що коректно відображається на різних типах пристроїв - комп'ютерах, планшетах, мобільних телефонах. Для цього можна використовувати медіа-запити в CSS.

Після розробки макету головної сторінки варто провести його тестування - перевірити коректність відображення в різних браузерах та на різних пристроях. Важливо звернути увагу на швидкість завантаження сторінки, бо це впливає на загальний користувацький досвід і SEO.

Також варто залучити до тестування реальних користувачів, щоб з'ясувати, наскільки зручним і зрозумілим є інтерфейс. Вони можуть виявити недоліки, які не були помічені під час розробки.

Отже, розробка макету головної сторінки - це не тільки створення естетично привабливого дизайну, а й забезпечення зручності користування, швидкості завантаження та коректного відображення на всіх типах пристроїв.

Головна сторінка буде мати такий вигляд: шапка сайту з лого «AngularShop»(назва лише для прикладу, у подальшому може змінитися), та матиме каталог товару. Фрагмент коду:

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>AngularShop</title>
  <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
  <header class="header">
    <div class="logo">AngularShop</div>
    <nav class="nav-menu">
      <ul>
        <li><a href="#">Каталог</a></li>
        <li><a href="#">Контакти</a></li>
      </ul>
    </nav>
  </header>
  <main class="main-content">
    <section class="product-cards">
```

```

<!-- Кожна карточка товару може мати таку структуру -->
<div class="product-card">
  
  <h3 class="product-name">Назва товару</h3>
  <p class="product-description">Опис товару</p>
  <p class="product-price">Ціна товару</p>
  <button class="buy-button">Купити</button>
</div>
<!-- Тут можуть бути інші карточки товарів -->
</section>
</main>
<footer class="footer">
  <p>© 2023, AngularShop</p>
</footer>
</body>
</html>

```

За допомогою такої структури головна сторінка сайту має наступний вигляд (рис. 1.7)

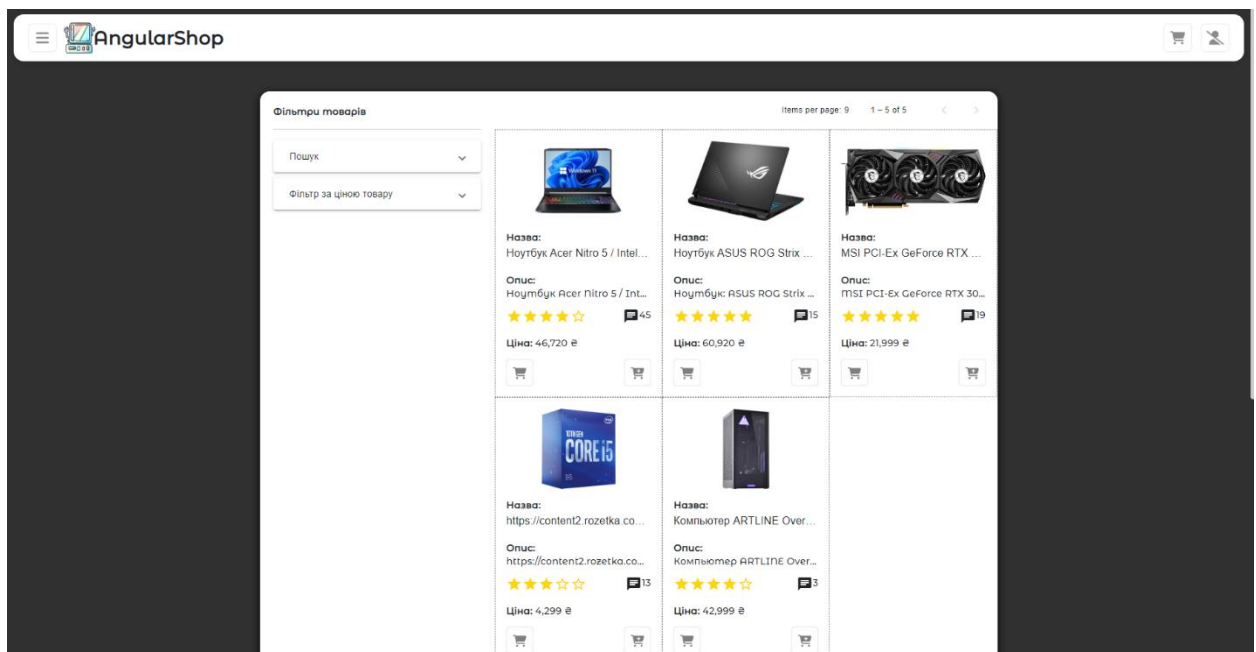


Рисунок 1.7 – Головна сторінка сайту

Каталог товарів є ключовим елементом для будь-якого інтернет-магазину. Він повинен бути зручним, інформативним та легким у навігації. Розробка макету каталогу товарів включає створення структури HTML та стилізацію за допомогою CSS.

					РП 06.18.001 ДП ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

HTML структура каталогу товарів може містити список товарів, де кожен товар представлений у вигляді карточки. Кожна карточка товару може включати зображення, назву, короткий опис та ціну товару, а також кнопку "Купити".

CSS використовується для стилізації карточок товарів, зокрема для встановлення їх розмірів, відступів, шрифтів тощо. Особливу увагу слід приділити створенню адаптивного дизайну, який забезпечує коректне відображення каталогу товарів на різних пристроях.

Нижче наведено приклади HTML і CSS коду для каталогу товарів.

```
<section class="product-catalog">
  <!-- Кожна карточка товару може мати таку структуру -->
  <div class="product-card">
    
    <h3 class="product-name">Назва товару</h3>
    <p class="product-description">Опис товару</p>
    <p class="product-price">Ціна товару</p>
    <button class="buy-button">Купити</button>
  </div>
  <!-- Тут можуть бути інші карточки товарів -->
</section>
* Стили для каталогу товарів */
.product-catalog {
  display: flex;
  flex-wrap: wrap;
  justify-content: space-between;
}

.product-card {
  width: calc((100% / 3) - 20px);
  margin-bottom: 20px;
  padding: 20px;
  border: 1px solid #ddd;
  border-radius: 5px;
}

.product-image {
  width: 100%;
  height: auto;
}
```

```
.buy-button {
  display: inline-block;
  padding: 10px 20px;
  color: white;
  background-color: #007bff;
  border-radius: 5px;
  text-decoration: none;
  text-align: center;
}
```

Після розробки макету каталогу товарів, також важливо провести тестування. Перевірити на коректність відображення в різних браузерах, на різних пристроях з різними розмірами екранів. Слід звернути увагу на коректність відображення зображень, текстових описів, цін та кнопки "Купити" на карточках товарів.

Далі необхідно провести тестування на користувацькому досвіді. Залучити реальних користувачів, щоб зрозуміти, чи зручно їм користуватися каталогом, чи зрозумілі навігація та описи товарів.

Важливо також перевірити швидкість завантаження сторінки каталогу, бо це може вплинути на загальний користувацький досвід і SEO.

Отже, розробка макету каталогу товарів - це не тільки створення естетично привабливого та інформативного дизайну, а й забезпечення зручності користування, швидкості завантаження та коректного відображення на всіх типах пристроїв.

Приклад вигляду сторінки каталогу наведено на рисунку 2.2.

					<i>РП 06.18.001 ДП ПЗ</i>	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

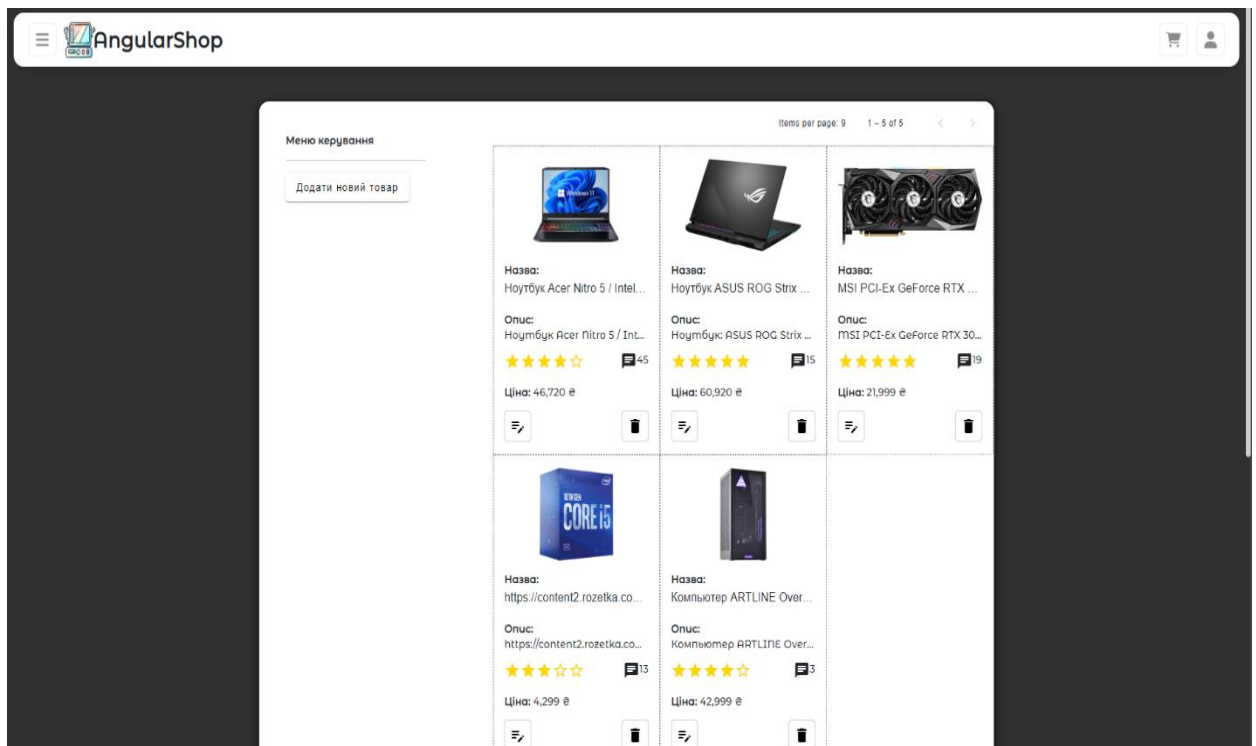


Рисунок 1.8 – Каталог товару користувача

Для того щоб здійснити заказ товарів, відкривається наступна форма (рис. 2.3)

Рисунок 1.9 – Замовлення товару

					РП 06.18.001 ДП ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

Створення сторінки окремого товару є одним з найважливіших аспектів розробки інтернет-магазину. Ця сторінка надає детальну інформацію про товар, включаючи зображення, повний опис, ціну, відгуки клієнтів та можливість додавання товару до кошика.

HTML структура сторінки товару може включати блоки для великого зображення товару, заголовку, детального опису, ціни, кнопки "Додати до кошика" та відгуків клієнтів.

CSS використовується для стилізації цих елементів та забезпечення зручності користування та читабельності інформації.

Важливою є також перевірка швидкості завантаження сторінки товару. Вона повинна завантажуватися достатньо швидко, щоб користувачі не втратили інтерес до товару і не покинули сайт.

Отже, розробка макету сторінки товару - це не тільки створення естетично привабливого та інформативного дизайну, а й забезпечення зручності користування, швидкості завантаження та коректного відображення на всіх типах пристроїв. Вигляд сторінки товару зображений на рис.2.4

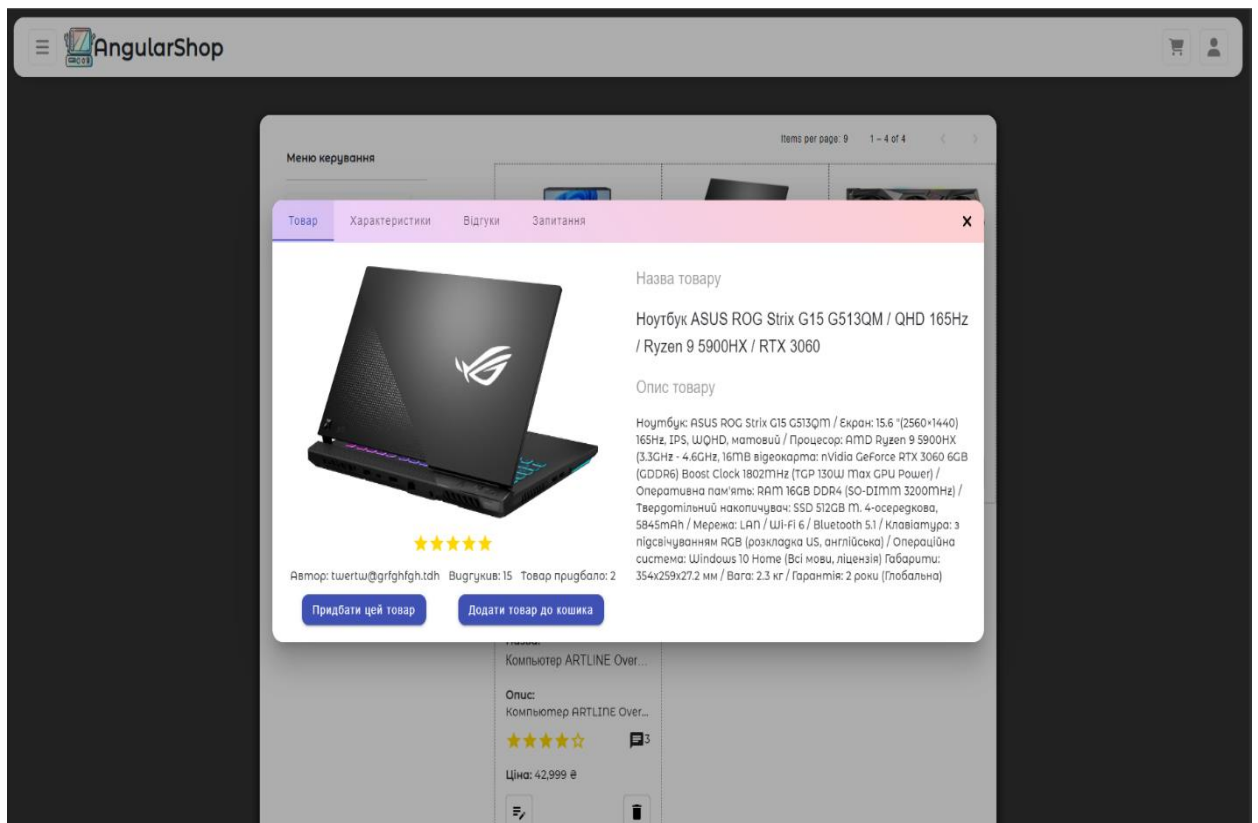


Рисунок 1.10 – Сторінка товару

						Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

Корзина для покупок є одним з основних елементів інтернет-магазину. Вона дозволяє користувачам зберігати обрані товари, переглядати загальну вартість покупок та переходити до оформлення замовлення.

HTML структура корзини може включати список товарів, які користувач додав до корзини, з інформацією про кількість кожного товару та його ціну. Також в корзині може бути інформація про загальну вартість покупок та кнопка "Оформити замовлення".

CSS використовується для стилізації цих елементів, забезпечуючи чіткість, читабельність та зручність використання.

1.2.2 Розробка серверної частини сайту.

Розробка серверної частини сайту є важливим етапом створення інтернет-магазину. Серверна частина відповідає за обробку запитів від клієнта, управління даними (товари, замовлення, інформація про користувачів), а також забезпечує безпеку та продуктивність сайту.

Ми використовуємо ASP.NET для розробки серверної частини, який є потужним і гнучким фреймворком для розробки веб-додатків.

На початку роботи над серверною частиною потрібно спроектувати базу даних, яка буде зберігати всю інформацію про товари, замовлення та користувачів. Для цього можна використовувати Entity Framework, який дозволяє легко інтегрувати базу даних з ASP.NET додатком.

Далі потрібно розробити API, який буде обробляти запити від клієнтської частини. ASP.NET MVC може бути використаний для створення контролерів, які обробляють ці запити, витягують необхідні дані з бази даних і відправляють їх назад клієнту у форматі JSON.

Також важливо реалізувати механізми безпеки, такі як аутентифікація та авторизація. ASP.NET Identity може бути використаний для цього.

Після розробки серверної частини, важливо провести її тестування. Можна використовувати інструменти, такі як Postman, для відправки запитів до API і перевірки відповідей. Також слід перевірити продуктивність та безпеку серверної частини.

					<i>РП 06.18.001 ДП ПЗ</i>	Арк.
						25
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Нижче наведено приклад коду контролера в ASP.NET, який обробляє запити до товарів:

```
public class ProductsController : ControllerBase
{
    private readonly StoreContext _context;

    public ProductsController(StoreContext context)
    {
        _context = context;
    }

    // GET: api/Products
    [HttpGet]
    public ActionResult<List<Product>> GetProducts()
    {
        return _context.Products.ToList();
    }

    // GET: api/Products/5
    [HttpGet("{id}")]
    public ActionResult<Product> GetProduct(int id)
    {
        var product = _context.Products.Find(id);

        if (product == null)
        {
            return NotFound();
        }

        return product;
    }

    // Other methods for updating, deleting, creating a product...
}
}
```

У цьому прикладі ProductsController обробляє HTTP-запити /Products на сервері. Атрибути HttpGet визначають, що методи повинні реагувати на HTTP GET-запити, а {id} в URL-шляху вказує на параметр методу.

StoreContext – це клас, який успадковується від DbContext і використовується для взаємодії з базою даних.

					<i>РП 06.18.001 ДП ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

Можна використовувати інструменти, такі як Postman, для відправлення запитів до API і перевірки відповідей.

Не менш важливо також перевірити продуктивність та безпеку серверної частини. Слід впевнитися, що сервер відповідає на запити швидко, і що всі дані користувачів захищені.

Для забезпечення безпеки даних користувачів, варто реалізувати механізми аутентифікації та авторизації. В ASP.NET для цього можна використовувати Identity, що дозволяє легко додавати функції реєстрації, входу в систему та управління профілями користувачів.

Таким чином, розробка серверної частини сайту включає проектування бази даних, розробку API для обробки запитів від клієнта, реалізацію механізмів безпеки, а також тестування продуктивності та безпеки сервера.

1.2.3 Розробка клієнтської частини сайту.

Клієнтська частина інтернет-магазину - це те, що бачить користувач при відвідуванні сайту. Вона включає в себе інтерфейс користувача, взаємодію з сервером через API, обробку даних від сервера та реагування на дії користувача.

Для розробки клієнтської частини ми використовуємо Angular - потужний JavaScript фреймворк, що дозволяє створювати гнучкі та високопродуктивні веб-додатки.

Angular пропонує компонентний підхід до розробки веб-додатків, де кожна частина інтерфейсу описується окремим компонентом. Компоненти можуть включати в себе шаблон (HTML), стилі (CSS) та логіку (TypeScript).

Для прикладу, ми можемо мати компонент "ProductCard", який відображає інформацію про один товар. Цей компонент може мати таку структуру:

```
product-card.component.html:  
<div class="product-card">  
  <img [src]="product.imageUrl" alt="{{ product.name }}">  
  <h2>{{ product.name }}</h2>  
  <p>{{ product.description }}</p>
```

					РП 06.18.001 ДП ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

```
<p>{{ product.price }} грн</p>
<button (click)="addToCart(product)">Додати до кошика</button>
</div>
```

product-card.component.css:

```
.product-card {
  /* стилі для карточки товару */
}
```

/* інші стилі... */

product-card.component.ts:

```
import { Component, Input } from '@angular/core';
import { Product } from './product.model';
```

```
@Component({
  selector: 'app-product-card',
  templateUrl: './product-card.component.html',
  styleUrls: ['./product-card.component.css']
})
```

```
export class ProductCardComponent {
  @Input() product: Product;
```

```
  addToCart(product: Product) {
    // код для додавання товару до кошика
  }
}
```

Компонент "ProductCard" приймає в якості вхідного параметра об'єкт "product", який містить інформацію про товар. Він використовує цю інформацію для відображення даних про товар в шаблоні. Кнопка "Додати до кошика" викликає метод "addToCart", коли на неї натискають.

Функція "addToCart" може виконувати різні дії, залежно від вимог до магазину. Наприклад, вона може відправити запит на сервер, щоб додати товар до корзини користувача, або зберегти товар в локальному сховищі браузера, якщо корзина реалізована на клієнтській стороні.

Розробка клієнтської частини також включає налаштування маршрутизації для навігації між різними сторінками сайту. Angular надає потужний механізм маршрутизації, який дозволяє визначати маршрути у вигляді URL-шляхів та відповідних компонентів.

					РП 06.18.001 ДП ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2.4 Розробка системи авторизації/реєстрації користувачів.

Авторизація та реєстрація користувачів - це важлива частина будь-якого інтернет-магазину. Це дозволяє користувачам створювати особисті облікові записи, зберігати інформацію про їх замовлення, корзину та особисті дані.

В рамках нашого проекту ми можемо розробити просту систему авторизації на основі ASP.NET Identity на серверній стороні та використовувати JWT (JSON Web Tokens) для авторизації на клієнтській стороні.

ASP.NET Identity - це гнучка система, що дозволяє управляти користувачами та ролями в додатку. Вона надає API для реєстрації, авторизації, зміни паролів, відновлення забутих паролів та ін.

JWT - це стандарт для безпечної передачі даних між двома сторонами через JSON об'єкт. В нашому випадку, після авторизації на сервері, користувач отримує JWT, який він повинен відправляти з кожним запитом до API для підтвердження своєї ідентичності.

На клієнтській стороні, Angular додаток повинен включати форми реєстрації та входу, які відправляють відповідні запити до сервера, а також сервіс аутентифікації, який зберігає JWT та використовує його для авторизації запитів до API.

Після розробки системи авторизації/реєстрації, її необхідно ретельно протестувати, переконавшись, що всі сценарії роботи коректно обробляються. Це включає реєстрацію нових користувачів, вхід вже існуючих користувачів, обробку помилок, таких як неправильний пароль або неіснуючий обліковий запис, та забезпечення безпечності даних користувачів.

Особливу увагу слід приділити безпеці. Необхідно переконатися, що паролі користувачів не зберігаються в явному вигляді, а хешуються та соляться перед зберіганням у базі даних. Також важливо забезпечити, що JWT не може бути перехоплений або змінений під час передачі між клієнтом та сервером, використовуючи для цього HTTPS-з'єднання.

					<i>РП 06.18.001 ДП ПЗ</i>	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

Крім того, система авторизації може включати додаткові функції, такі як двофакторна автентифікація, зміна паролю, відновлення забутого паролю через електронну пошту, блокування облікового запису після декількох невдалих спроб входу тощо.

Завершивши розробку та тестування системи авторизації/реєстрації, можна перейти до наступного етапу - розробки функціоналу для обробки замовлень.

Рис. 1.11– Форма реєстрації та авторизації

1.2.5 Документація для користувача

При першому відвідуванні нашого сайту ви зможете побачити головну сторінку додатка, яка містить всі товари користувачів сайту. Сайт складається з трьох основних частин: хедеру, боді та футеру.

На хедері розміщуються різноманітні кнопки: бургер-меню з навігацією по сайту та логотипом з лівого боку, а з правого боку - кнопки корзини та авторизації/управління аккаунтом.

					РП 06.18.001 ДП ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

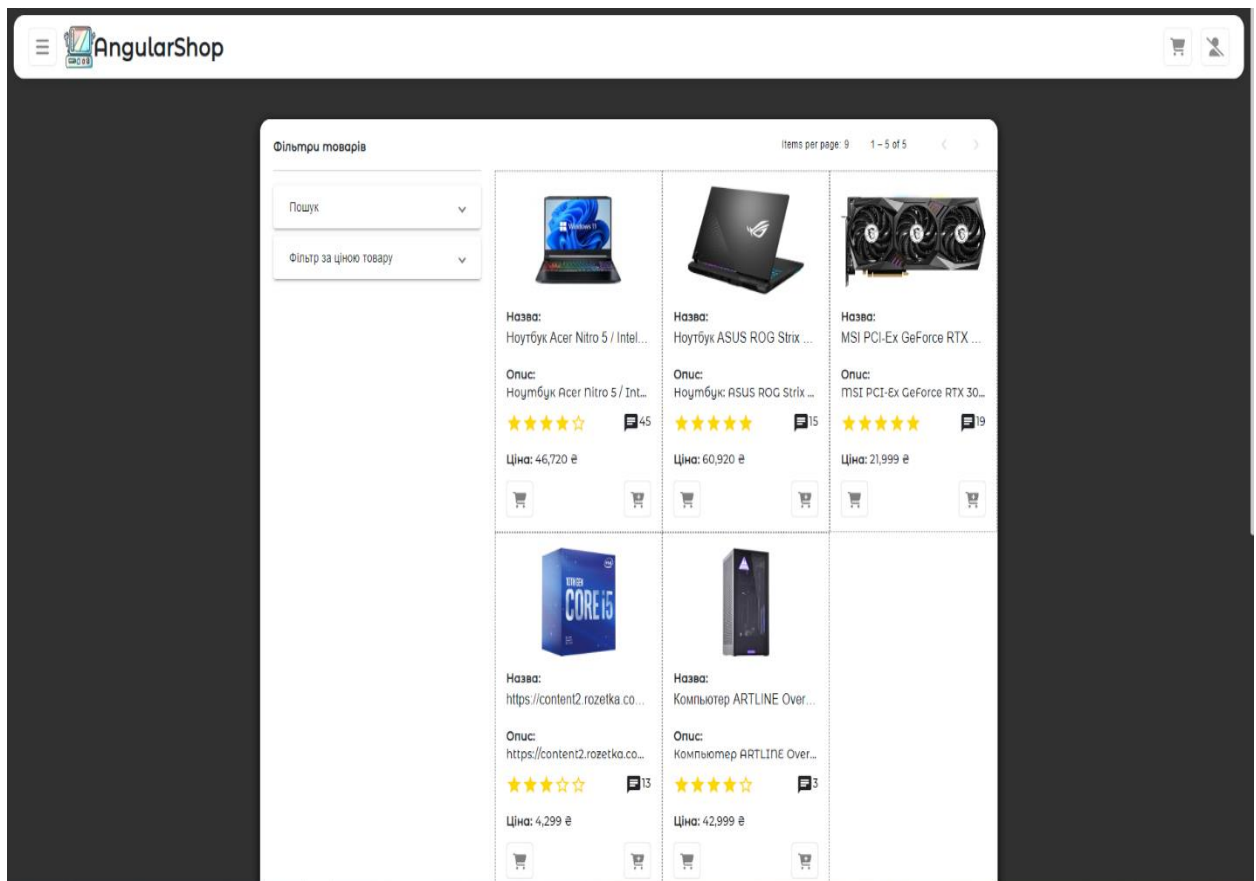


Рисунок 1.12– Головна сторінка інтернет-магазину

У тілі боді буде контент, який відповідає вибраному роуту через навігацію. Наприклад, користувач може побачити всі товари на сайті, а також є окремий роут, за допомогою якого користувач може побачити лише свої товари. Також є роут, за допомогою якого користувач зможе переглянути/редагувати свій профіль.

На футері розміщується важлива інформація про сайт, включаючи кнопки-іконки соціальних мереж та слова вдячності для тих, хто обрав наш сайт.

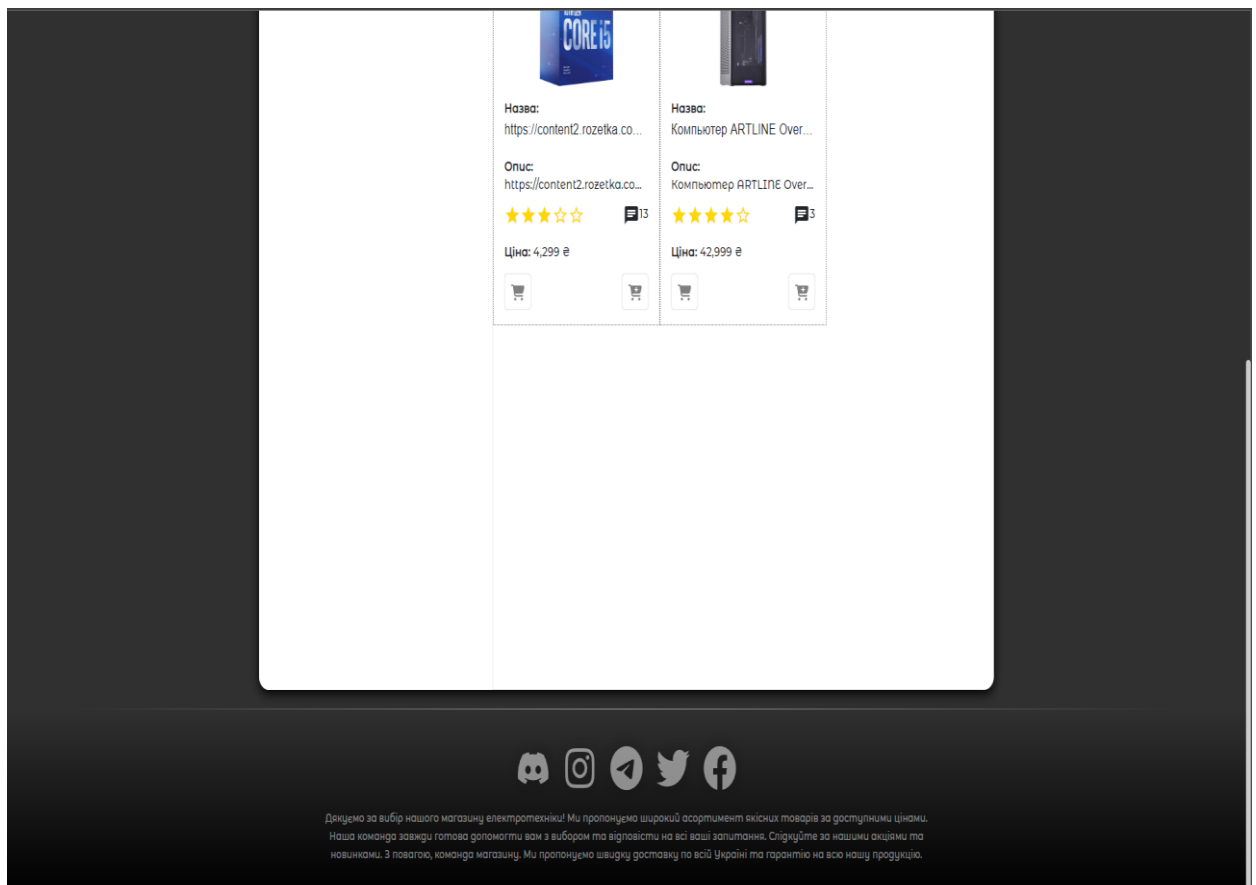


Рисунок 1.13 – Футер сайту

Функціонал сайту (конкретніше: взаємодія з сайтом та товарами) відкривається після того, як користувач зареєструється або увійде в свій аккаунт.

Для того, щоб зареєструватися або увійти в аккаунт, користувач може натиснути на відповідні кнопки в навігаційному меню з правого боку. Після натискання на профіль відкриється меню, де буде 2 кнопки на вибір: зареєструватися або увійти. Після натискання на одну з них користувач побачить діалогове вікно з формою реєстрації/авторизації.

					<i>РП 06.18.001 ДП ПЗ</i>	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

Рис. 1.14 – Форма авторизації

Після того, як користувач успішно зареєструвався або увійшов у свій аккаунт, він отримує доступ до всіх функцій сайту.

Корзина товарів: користувач може додавати товари до своєї корзини, переглядати та редагувати вміст корзини, а також оформлювати замовлення. При натисканні на кнопку корзина у верхньому правому куті екрана користувач перенаправляється на сторінку з корзиною.

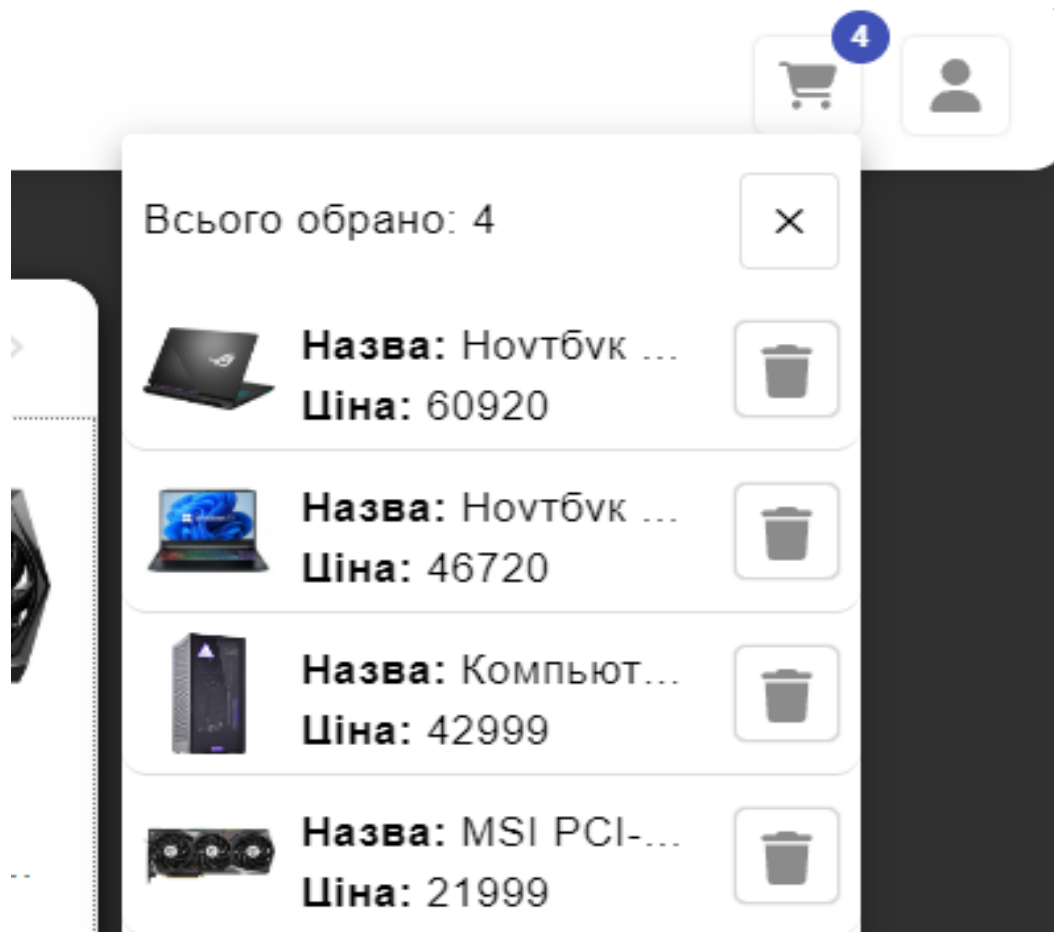


Рис. 1.15 – Кошик покупця

Профіль користувача: на сторінці профілю користувач може редагувати свої особисті дані, переглядати своє замовлення та інформацію про них, а також змінювати пароль.

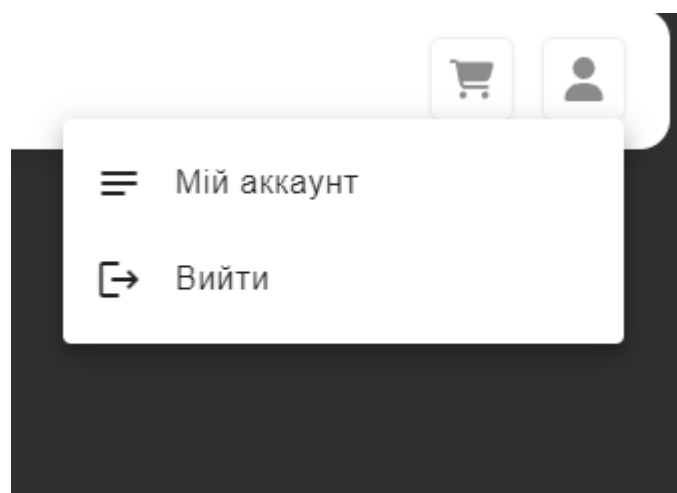


Рис. 1.16 – Панель управління аккаунтом

Оформлення замовлення: при переході до оформлення замовлення користувач може вказати адресу доставки, вибрати спосіб оплати, а також додати коментар до замовлення. Після оформлення замовлення користувач отримує підтвердження про успішне замовлення, а також інформацію про дату та час доставки.

Пошук товарів: на сторінці з товаром користувач може використовувати фільтри для пошуку товарів за текстом, ціною.

Рейтинг товарів: на сторінці з товаром користувач може залишати відгуки та оцінки товару.

Якщо користувач авторизований, він може додавати чужі товари в корзину та купувати їх. Крім того, він може ставити рейтинг у зірках та писати відгуки, якщо купив даний товар. Користувач також може задавати запитання під товаром та залишати відповіді для тих, хто задав питання.

Для своїх товарів користувач може робити наступне: редагувати створений товар, видаляти створений товар та створювати свої товари.

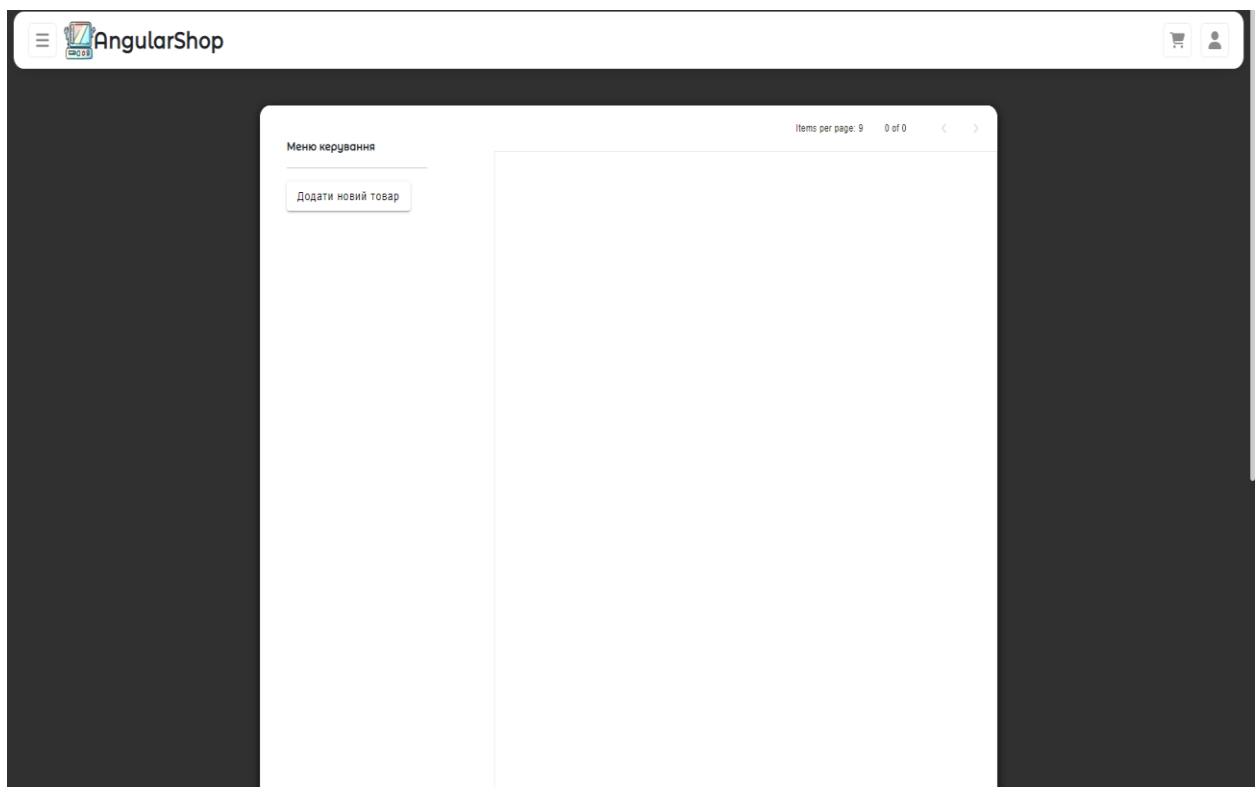


Рис. 1.17 – Панель управління товарами

					<i>РП 06.18.001 ДП ПЗ</i>	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

Створення товару відбувається наступним чином: користувач натискає на відповідну кнопку на роуті своїх товарів, після чого відкривається діалогове вікно, в якому він може створити товар. Товар створюється за його назвою, описом, зображенням, ціною та додаванням характеристик до нього (назва характеристики та тип). Щоб додати характеристику до товару, потрібно перейти на другий етап додавання товару, натиснути на відповідну кнопку, після чого створюється поле, у якому буде індикатор на скільки воно заповнене. Щоб воно стало валідним, потрібно заповнити назву та тип, після чого треба натиснути на 3 крапки та зберегти, щоб поле стало валідним. Форма має валідацію на поля - щоб кнопка створення стала доступною, всі поля повинні бути заповнені та має бути хоча б 1 збережена характеристика.

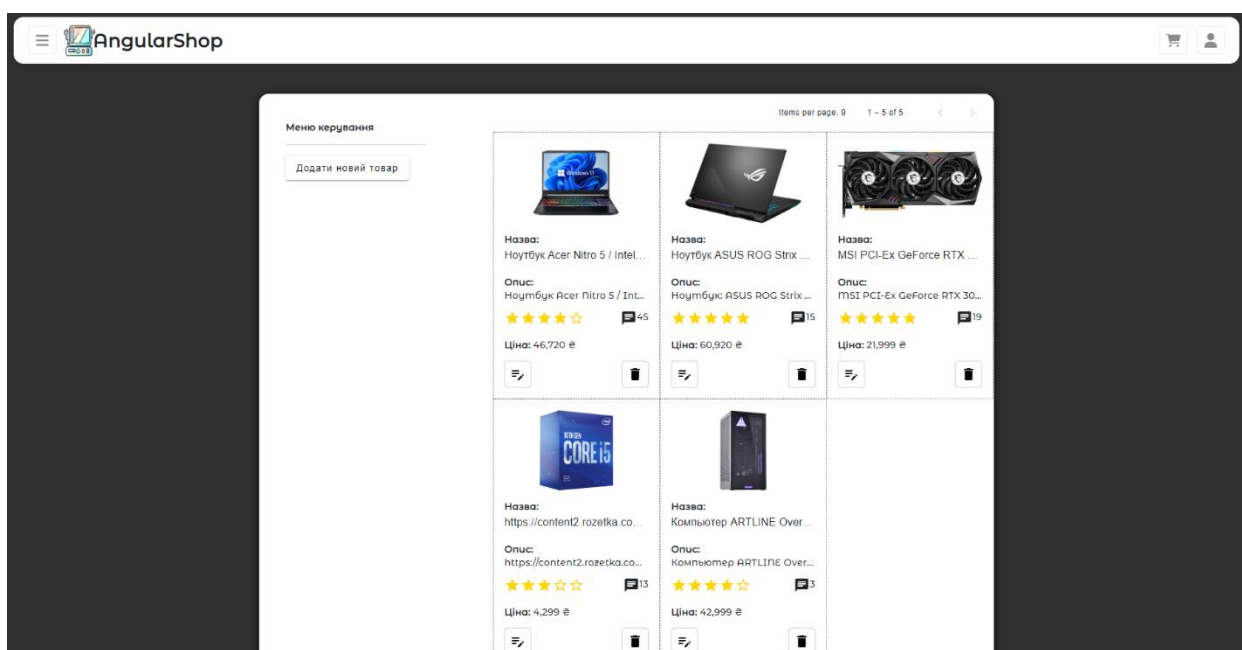


Рис. 1.18 – Збережені товари

Також на сайті присутня можливість пошуку товарів за назвою або за фільтрами, такими як категорії, ціновий діапазон, рейтинг тощо. Кнопки для пошуку та фільтрації розташовані на головній сторінці сайту, а також на сторінці з каталогом товарів.

					<i>РП 06.18.001 ДП ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

Крім того, сайт має систему оплати товарів. Після додавання товару до кошика та переходу до оформлення замовлення. Для покупки товару є діалогове вікно, в якому користувач може ввести своє ім'я прізвище, адресу доставки та вказати дані картки (номер, дату, CVV2) та вказати кількість товару, після чого стане доступна кнопка оформлення покупки.

					<i>РП 06.18.001 ДП ПЗ</i>	Арк.
						37
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

2 ЕКОНОМІЧНА ЧАСТИНА

В дипломному проекті розроблено функціональний інтернет-магазин комп'ютерної техніки. Розробка включала створення різних компонентів сайту, включаючи головну сторінку, каталог товарів, сторінку товару, кошик та систему авторизації та реєстрації користувачів. Кожен з цих компонентів було розроблено з урахуванням особливостей роботи інтернет-магазину та вимог до зручності користувачів.

Завдяки використанню сучасних технологій, таких як ASP.NET для серверної частини та Angular для клієнтської частини, було створено ефективний і безпечний сервіс для продажу комп'ютерної техніки.

Створений сайт призначений для отримання прибутку його власником за рахунок залучення потенційних клієнтів з конкретної цільової аудиторії при мінімальних витратах.

Навряд чи хто буде заперечувати твердження, що реклама – двигун торгівлі. Але реклама буває різною: привабливою і неприємною, розумною і дратівливою, дорогою та не дуже.

Якщо великі фірми можуть дозволити собі проводити рекламні акції вартістю в багато сотень тисяч гривень (телебачення, радіо, газети, спеціалізовані журнали, презентації тощо), то у фірм малого і середнього бізнесу можливості скромніше. Великі фірми можуть дозволити собі проводити рекламні акції з віддачею результатів реклами в перспективі, а фірми малого і середнього бізнесу потребують негайних результатів.

Саме тому доцільне створення сайтів для підприємств малого та середнього бізнесу, тому що позитивний результат виявляється практично відразу після розміщення сайту в Інтернеті. Саме малі підприємства (фірми) щонайкраще орієнтуються на ринку, гостро відчувають, що в даний момент цікавить споживача і здатні оперативно задовольнити його вимоги.

При оцінці ефективності створюваного сайту слід виходити з того, що залежно від характеру ефекту, що досягається, можуть бути визначені наступні види ефективності сайту: економічна, функціональна та соціальна

					РП 06.18.002 ДП ПЗ	Арк.
						38
Змн.	рк.	докум.	дпис	Дата		

ефективність. Ці види ефективності в свою чергу залежать від технологічних, технічних, маркетингових та економічних показників веб-сайту. Між доходом від сайту і витратами на його створення існує певна залежність. Слід також враховувати, що на величину доходу від сайту, окрім інших чинників ефективності, можуть вплинути витрати на його оптимізацію та просування.

В цьому розділі пояснювальної записки до дипломного проекту обчислимо економічну ефективність розробки створеного web-сайту.

Загальні витрати (B_3) на створення сайту складаються з декількох параметрів:

$$B_3 = B_p + B_v + B_e, (2.1)$$

де B_p – витрати на розробку сайту;

B_v – витрати на впровадження сайту;

B_e – витрати на експлуатацію сайту;

Витрати на розробку сайту (B_p) є одноразовими та складаються з вартості наступних видів робіт зі створення сайту: розробка дизайну сайту; реалізація на сайті меню; підготовка сторінок-шаблонів; наповнення сайту інформацією; обробка зображень для публікації їх на веб-сторінках; програмна розробка сайту; реалізація пошукових можливостей сайту та ін.

Витрати на впровадження сайту (B_v) складаються з витрат на реєстрацію доменного імені терміном мінімум на один рік та з витрат на реєстрацію в пошукових системах.

Витрати на експлуатацію сайту (B_e) складаються з витрат на підтримку сайту в робочому стані та витрат на хостинг, заробітну платню обслуговуючому персоналу – адміністратору, який буде підтримувати сайт у робочому стані (як правило не щомісячну, а в разі необхідності – згідно акту виконаних робіт), пошук і просування сайту та оплату послуг провайдера доступу до мережі Інтернет.

В зв'язку з тим, що орендна плата за канали зв'язку, приміщення, електрику та ін. входить до оплати послуг хостингу (так само, як і амортизаційні відрахування), при розрахунку економічної ефективності веб-

					<i>РП 06.18.002 ДП ПЗ</i>	Арк.
						39
Змн.	рк.	докум.	дпис	Дата		

сайту, який не розміщений на технічному обладнанні підприємства та у його приміщеннях (орендованих площах), ці витрати окремою статтею враховувати не потрібно.

Роботи по підтримці сайту в робочому стані включають в себе: оновлення даних на сайті; створення нових розділів на сайті; видалення застарілої інформації з сайту; додавання потрібної інформації на сайт; налаштування параметрів сервера хостингу; моніторинг роботи сервера хостингу; забезпечення щомісячного захисту сайту; створення резервних копій сайту тощо.

Для визначення витрат на розробку сайту (B_p) необхідно розрахувати оплату праці виконавців, безпосередньо задіяних для її виконання. Для реалізації проекту web-системи використовуються наступні професійні ресурси: дизайнер, програміст, веб-технолог і тестувальник.

Для визначення трудомісткості розробки сайту (B_p) складено план-графік по розробці web-сайту і тривалості виконання робіт. Розподіл робіт по етапах і видах виконавців наведено в таблиці 2.1.

Таблиця 2.1. План-графік по розробці Web-сайту

<i>Назва етапу</i>	<i>Час виконання (днів)</i>	<i>Посада виконавця</i>
Аналіз завдання на розробку	1	Веб-технолог
Аналіз технологій побудови Web-системи	2	
Проектування структури сайту	2	
Проектування схеми роботи сайту	2	
Створення дизайну Web-сторінок	4	Дизайнер
Розробка Web-сторінок	4	Веб-технолог Дизайнер

	<i>Назва етапу</i>	<i>Час виконання (днів)</i>	<i>Посада виконавця</i>
	Виконання верстки Web-сторінок	3	Веб-технолог Дизайнер Програміст
	Підключення CSS-фреймворку Bootstrap	2	Програміст
	Написання коду сценаріїв на JavaScript та PHP та робота з БД	5	Програміст
0	Відладка і тестування роботи сайту	2	Програміст Тестувальник
1	Завантаження створеного сайту на хостинг	1	Веб-технолог Тестувальник
ВСЬОГО:		28	

Розрахунок трудомісткості здійснений в наступній послідовності:

1. Складений перелік всіх етапів і видів робіт, які необхідно виконати в ході даної розробки. Після узгодження з керівником проекту допущено виключення, доповнення, об'єднання окремих етапів і видів робіт.

2. По кожному виду робіт визначений кваліфікаційний рівень виконавців. В разі виконання однієї роботи виконавцями різної кваліфікації, робота розподілена на ряд паралельних конкретних робіт для кожної категорії виконавця.

В умовах відсутності нормативної бази тривалість виконання окремих робіт розраховується на основі вірогідних оцінок робіт, що задаються виконавцями.

Розмір заробітної плати розраховується виходячи з чисельності різних категорій виконавців, трудомісткості, що витрачається ними на виконання різних видів робіт, а також їх середньої заробітної плати (ставки)

					<i>РП 06.18.002 ДП ПЗ</i>	Арк.
						41
<i>Змн.</i>	<i>рк.</i>	<i>докум.</i>	<i>дпис</i>	<i>Дата</i>		

за один робочий день.

Витрати на заробітну плату приведені в таблиці 3.2.

Відповідно до статті 8 «Закону про Державний бюджет України на 2023» встановлено мінімальну заробітну плату у місячному розмірі з 1 січня 2023 року – 6700 гривень; мінімальну погодинну тарифну ставку – 40,46 грн.

Середня заробітна плата за один робочий день для кожного виконавця визначається по формулі:

$$Z_{\text{ден}} = \text{п.т.с.} \times 8 \text{ (2.2),}$$

де:

п.т.с. – погодинна тарифна ставка, грн.

8 – тривалість робочого дня, год.

$$Z_{\text{ден}} \text{ дизайнера} = 100,00 \times 8 = 400,00 \text{ грн.}$$

$$Z_{\text{ден}} \text{ програміста} = 100,00 \times 8 = 400,00 \text{ грн.}$$

$$Z_{\text{ден}} \text{ веб-технолога} = 80,00 \times 8 = 640,00 \text{ грн.}$$

$$Z_{\text{ден}} \text{ тестувальника} = 80,00 \times 8 = 640,00 \text{ грн.}$$

Таблиця 2.2. Витрати на заробітну плату

№	Персонал	Етапи розробки	Кількість днів	Денна ставка, грн.	Витрати, грн.
1	Дизайнер	5, 6, 7	10	800,00	8000,00
2	Програміст	7, 8, 9, 10	11	800,00	8800,00
3	Веб-технолог	1–4, 6, 7, 11	13	640,00	8320,00
4	Тестувальник	10, 11	3	640,00	1920,00
ВСЬОГО:					27040,00

До складу витрат на оплату праці також включаються податки, збори та інші обов'язкові платежі, встановлені діючою системою оподаткування. Розмір єдиного соціального внеску наразі складає 22% від заробітної плати. Отже обчислюємо його за наступною формулою:

$$V_{\text{есв}} = V_{\text{зп}} \times 0,22 = 27040,00 \times 0,22 = 5948,8 \text{ грн. (2.3)}$$

Визначимо загальні витрати (V_p) на розробку рекламного веб-сайту для фірми з продажу та оренди звукового обладнання, як суму витрат на заробітну плату праці персоналу ($V_{\text{зп}}$) та єдиного соціального внеску ($V_{\text{есв}}$):

$$V_p = V_{\text{зп}} + V_{\text{есв}} \text{ (2.4)}$$

$$V_p = 27040,00 + 5948,80 = 32988,80 \text{ грн.}$$

Витрати на впровадження та експлуатацію сайту варто розглядати як постійні витрати ($V_{\text{пост}}$), які будуть враховуватися протягом життя проекту:

$$V_{\text{пост}} = V_v + V_e \text{ (2.5)}$$

У таблиці 3.3 визначимо постійні витрати як суму витрат на впровадження та експлуатацію сайту протягом року (щомісячні витрати помножимо на 12 для отримання суми витрат протягом року, а заробітну платню адміністратору веб-сайту врахуємо як щорічну послугу, в зв'язку з тим, що роботи по оновленню контенту на сайті проводяться раз на декілька місяців), включимо до цих витрат: реєстрацію доменного імені в домені in.ua; послуги хостингу (щомісячна витрата); послуги з пошукової оптимізації сайту; послуги з пошукового просування сайту; послуги провайдера, що надає широкопasmовий доступ до мережі Інтернет (щомісячна витрата); заробітна платня обслуговуючому персоналу (адміністратору веб-сайту).

					<i>РП 06.18.002 ДП ПЗ</i>	Арк.
						43
Змн.	рк.	докум.	дпис	Дата		

Таблиця 2.3. Постійні витрати

№	Стаття витрат	Вартість за рік, грн.
1	Доменне ім'я (in.ua)	312,00
2	Хостинг	12*119,36 = 1432,32
3	Пошукова оптимізація сайту	810,00
4	Пошукове просування сайту	1620,00
5	Доступ до мережі Інтернет	12*140,00 = 1680,00
6	Заробітна платня адміністратору (на рік)	6700,00
ВСЬОГО:		12554,32

Враховуючи отримані показники, розрахуємо загальні витрати (V_3) на створення, впровадження та експлуатацію веб-сайту:

$$V_3 = V_p + (V_v + V_e) = V_p + V_{\text{пост}} \quad (2.6)$$

$$V_3 = 32988,80 + 12554,32 = 45543,12 \text{ грн.} \quad (2.7)$$

Для розрахунків ефективності впровадження розробки було вирішено використати метод визначення періоду окупності витрат на створення рекламного web-сайту. Цей метод враховує загальні витрати на розробку, впровадження та експлуатацію web-сайту, вірогідну кількість здійснених оптових замовлень та можливий прибуток із замовлення.

Прибуток від кожної угоди приблизно оцінимо як середнє від середньої мінімальної вартості кожної з послуг, яку можливо отримати в компанії, приблизне значення прибутку визначимо 1500 грн.

Згідно відкритих Інтернет-джерел, орієнтовне число відвідувачів сайту компанії в день коливається від 10 до 20. За статистикою 2-3 з загальної кількості відвідувачів сайту купують товари або замовляють послуги, пропоновані на сайті, тоді кількість угод складатиме приблизно

0,2-0,3 в день. Для розрахунків візьмемо середнє значення, яке дорівнює 0,25. Термін окупності створення веб-сайту розрахуємо по формулі:

$$T_{\text{окуп}} = B_z / K_y \times \Pi, \text{ (2.8) де}$$

$T_{\text{окуп}}$ – термін окупності, дні;

K_y – кількість угод в день;

Π_i – середнє значення прибутку, грн.

$$T_{\text{окуп}} = 45543,12 / (0,25 \times 1500) = 121 \text{ день або } 4 \text{ місяці}$$

Далі розрахуємо приріст прибутку за рік:

$$\Delta\Pi = \Pi_i \times K_y \times Д, \text{ (2.9) де}$$

$Д$ – кількість робочих днів за рік;

$$\Delta\Pi = 1500 \times 0,25 \times 251 = 94125 \text{ грн. (2.10)}$$

На основі проведених розрахунків визначаємо, що розробка і впровадження веб-сайту буде вигідним для компанії з продажу комп'ютерної техніки оскільки окупиться приблизно за 4 місяці та може забезпечити приріст прибутку за рік 94125 грн.

					<i>РП 06.18.002 ДП ПЗ</i>	Арк.
						45
Змн.	рк.	докум.	дпис	Дата		

3 ОХОРОНА ПРАЦІ

Безпека праці відіграє ключову роль у практичній сфері діяльності, спрямованій на створення безпечних і нешкідливих умов праці. Забезпечення безпеки та запобігання небезпечним умовам праці вимагає значних фінансових вкладень і впровадження наукових розробок в галузі охорони праці.

Умови праці для програмістів повинні відповідати нормам безпеки праці, враховуючи законодавчі акти, такі як "Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями" (НПАОП 0.00-7.15-18) та "Правила охорони праці під час експлуатації електронно-обчислювальних машин" (НПАОП 0.00-1.28-10), що були затверджені Державним комітетом України з промислової безпеки, охорони праці та гірничого нагляду 26 березня 2010 року наказом № 65.

Тому знання про безпеку праці мають вирішальне значення для технічних спеціалістів, зокрема програмістів.

3.1 Гігієнічні вимоги до виробничого середовища

3.1.1 Вимоги до приміщення

– Співіставлення з нормативно-правовими актами вимагає, щоб площа одного робочого місця була не менша за 6,0 квадратних метри, а об'єм - 20,0 кубічних метрів.

– Стосовно кліматичних умов, приміщення має бути обладнане системою опалення для забезпечення комфортної температури, кондиціонуванням повітря для контролю вологості та системою припливно-втяжної вентиляції для забезпечення свіжого повітря.

– Щодо поверхні підлоги, вона повинна бути рівною, неслизькою та мати антистатичні властивості для запобігання статичному електрику, який може пошкодити обладнання.

– Забезпечення вимог безпеки праці включає не тільки належну організацію приміщення, але також враховує розгляд конструкції робочого

					РП 06.18.003 ДП ПЗ	Арк.А
						46
Змн.	Арк.А	№ докум.№	ПідписПі	Дата		

місця та його обладнання з метою забезпечення оптимальної робочої позиції та комфорту працівника. З цією метою важливо дотримуватися таких вимог:

– **3.1.2 Вимоги до організації робочого місця користувача ПК**

– При організації робочого місця важливо враховувати не лише вимоги до освітлення, а й інші аспекти ергономіки, які забезпечують оптимальну робочу позицію користувача та сприяють збереженню здоров'я, зниженню втоми й підвищенню продуктивності під час роботи з комп'ютером.

– Робочі місця з візуальним екраном (ВДТ) слід розташовувати таким чином, щоб природне світло падало з боку, переважно зліва, що допомагає уникнути надмірного навантаження на очі працівника.

– Відстань між робочими місцями: Важливо дотримуватися відстані не менше 1,2 метра між бічними поверхнями ВДТ. Також відстань від тильної поверхні одного ВДТ до екрана іншого ВДТ повинна становити 2,5 метра, щоб уникнути зайвих перешкод і забезпечити комфортну робочу зону.

– Конструкція робочого столу: Робочий стіл має відповідати сучасним ергономічним вимогам. Він повинен забезпечувати оптимальне розташування обладнання (дисплея, клавіатури, принтера) й документів на робочій поверхні. Крім того, робочий стіл повинен мати достатньо простору для ніг, регульовану висоту, кут і нахил сидіння й спинки, а також зручну форму переднього краю сидіння. Регулювання цих параметрів має бути легким, незалежним і надійним.

– Робочий стілець має бути гнучким у налаштуванні, з можливістю підйому й повороту, а також регулювання висоти, кута нахилу сидіння й спинки. Поверхня сидіння повинна бути рівною, а передній край слід заокруглити для комфорту працівника. Регулювання кожного параметра має бути простим й надійним. Крім того, шаг регулювання елементів стільця повинен бути в межах 15-20 мм для лінійних розмірів й 2-5 градусів для кутових розмірів. Сила, необхідна для здійснення регулювання, не повинна бути великою.

					<i>РП 06.18.003 ДП ПЗ</i>	Арк.А
						47
Змн.	Арк.А	№ докум.№	ПідписПі	Дата		

– Для забезпечення оптимальної робочої позиції під час користування комп'ютерами та моніторами, необхідно враховувати наступні організаційні вимоги до робочого місця:

– Розташування робочих столів з моніторами: Важливо розташовувати робочі місця так, щоб природне світло падало збоку, переважно зліва. Це забезпечить належне освітлення і запобіжить зайвому напруженню очей.

– Відстань між моніторами: Робочі столи з моніторами повинні бути розміщені на відстані не менше 1,2 метра один від одного по боках. Крім того, відстань від задньої поверхні одного монітора до екрану іншого монітора повинна бути 2,5 метра, щоб уникнути взаємного впливу екранів.

– Ергономічна конструкція робочого столу: Робочі столи повинні відповідати сучасним вимогам ергономіки і забезпечувати належне розташування обладнання (монітора, клавіатури, принтера) і документів на робочій поверхні. Важливо, щоб робочий стіл мав достатньо місця для ніг, з висотою не менше 600 мм, шириною не менше 500 мм, глибиною на рівні колін не менше 450 мм і на рівні простягнутої ноги не менше 650 мм.

– Регульований стілець: Стілець, на якому працює користувач, повинен мати можливість підніматися та повороту, а також регулюватися за висотою, кутом сидіння та спинки. Важливо також забезпечити можливість регулювання відстані між спинкою та переднім краєм сидіння. Поверхня сидіння повинна бути плоскою, а передній край стільця - заокругленим, щоб забезпечити комфорт під час сидіння. Кожен параметр стільця повинен бути легко регульованим і надійно фіксуватися в потрібному положенні. Крок регулювання елементів стільця має складати приблизно 15-20 мм для лінійних розмірів і 2-5 градусів для кутових регулювань. Зусилля, необхідні для регулювання, не повинні перевищувати 20 Н (ньютонів).

Дотримання цих організаційних вимог щодо робочого місця для користувачів комп'ютерів та моніторів забезпечить їм комфортні умови праці.

					<i>РП 06.18.003 ДП ПЗ</i>	Арк.А
						48
Змн.	Арк.А	№ докум.№	ПідписПі	Дата		

Це сприятиме уникненню непотрібного фізичного напруження та покращенню їх продуктивності й здоров'я.

3.1.3 Освітлення робочого місця

Відповідно до ДСанПІН 3.3.2.007-98, рекомендується використовувати рівномірне загальне освітлення в приміщеннях з робочими місцями, де використовуються комп'ютери та монітори. Це означає, що світло має розподілятися однаково по всій робочій зоні.

У виробничих та адміністративно-громадських приміщеннях, де проводиться робота з документами, може бути застосована комбінована система освітлення. Вона передбачає використання як загального освітлення, так і світильників місцевого освітлення для додаткового підсвічування окремих зон з документами.

Рекомендована освітленість на робочому столі у зоні розміщення документів складає 300-500 люксів. Якщо системою загального освітлення не можна досягти такого рівня освітленості, можна використовувати світильники місцевого освітлення. Але важливо уникати блисків на екрані комп'ютера та не допускати освітленість екрана більше 300 люксів.

При виборі джерел світла рекомендується використовувати переважно люмінесцентні лампи типу ЛБ. У деяких приміщеннях можуть бути використані металогалогенні лампи потужністю 250 Вт для відбитого освітлення. Для світильників місцевого освітлення можна використовувати лампи розжарювання.

При використанні штучного освітлення важливо уникати прямого сонячного світла на робочому місці, оскільки це може призводити до блисків на екрані й завдавати незручності користувачеві.

Додатково, рекомендується застосовувати світлодіодні лампи (ЛД) разом з розсіювачами та екрануючими сітками. Це сприяє рівномірному розподілу світла і попереджає появу різких тіней або блисків на робочій поверхні.

					<i>РП 06.18.003 ДП ПЗ</i>	Арк.А
						49
Змн.	Арк.А	№ докум.№	ПідписПі	Дата		

3.1.4 Мікроклімат

Згідно зі стандартами ДСН 3.3.6.042.-99, у виробничих приміщеннях, де розташовані робочі місця з використанням візуально-дисплейних технологій (ВДТ), важливо забезпечити оптимальні параметри мікроклімату, такі як температура, відносна вологість та рух повітря, відповідно до значень, наведених у таблиці 1.

Таблиця 3.1

Пора року	Категорія робіт	Температура повітря, град.С	Відносна вологість повітря	Швидкість руху повітря м/с
		оптимальна	оптимальна	оптимальна
Холодна	Легка-1 а	22-24	40-60	0.1
	Легка-1 б	21-23	40-60	0.1
Тепла	Легка-1 а	23-25	40-60	0.1
	Легка-1 б	22-24	40-60	0.2

З метою створення комфортних умов на робочих місцях з використанням візуально-дисплейних технологій (ВДТ) і забезпечення здоров'я та продуктивності працівників, особливу увагу слід звернути на забезпечення оптимальних параметрів мікроклімату.

Одним з аспектів цього є рівень іонів у повітрі

У відповідному приміщенні використовується іонізатор повітря, який сприяє підтримці оптимального рівня іонів у повітрі, створюючи комфортні умови для працівників.

Згідно з нормативними вимогами, встановленими санітарно-гігієнічними нормами N 2152-80, рівні позитивних і негативних іонів у повітрі приміщень з ВДТ мають відповідати певним стандартам. Для отримання конкретних вказівок щодо цих норм можна звернутися до таблиці 3.2.

– При необхідності, зупинити доступ повітря до вогнища використовуючи доступні засоби.

На робочому місці використовуються вуглекислотні та порошкові вогнегасники для придушення пожежі. Кількість, види вогнегасників та їх розташування відповідають нормам ГОСТ 12.4.009-75 та ISO3941-77.

У приміщенні дотримуються всі визначені вимоги щодо пожежної безпеки відповідно до національних стандартів України, зокрема, вимог НАПБ А.0.001-95 "Правила пожежної безпеки в Україні". Також встановлений план евакуації на випадок пожежі, відповідний час для проведення евакуації відповідає вимогам стандарту СНиП 2.01.02-85, а максимальна відстань між робочими місцями та виходами для евакуації відповідає вимогам СНиП 2.09.02-85.

Дослідження та вирішення проблем, пов'язаних із забезпеченням здорових і безпечних умов праці, є надзвичайно важливими в системах виробництва. Це відноситься до одного з найзначущих завдань, спрямованих на забезпечення безпеки та добробуту працівників. Проведення досліджень і виявлення можливих причин виробничих нещасних випадків, професійних захворювань, аварій і пожеж, а також розробка ефективних заходів для їх усунення є способом створення безпечних умов для праці людей. Це сприяє збереженню здоров'я працівників, підвищенню продуктивності праці і поліпшенню загального функціонування організації.

Також, забезпечення комфортних та безпечних умов праці відіграє суттєву роль у забезпеченні благополуччя працівників та ефективності їхньої роботи. Це охоплює належні параметри мікроклімату, адекватне освітлення, застосування безпечного обладнання та вжиття відповідних заходів безпеки. Створення таких умов сприяє підтримці фізичного та психологічного комфорту працівників, зменшенню ризику нещасних випадків та покращенню загального робочого середовища в організації. Це суттєвий фактор, що сприяє здоров'ю працівників та сприятливому розвитку підприємства.

					<i>РП 06.18.003 ДП ПЗ</i>	Арк.А
						52
Змн.	Арк.А	№ докум.№	ПідписПі	Дата		

3. Розробка серверної частини на ASP.NET:

- Реалізовано API для обробки запитів від клієнтської частини, що включає отримання списку товарів, додавання товару в корзину, реєстрацію та авторизацію користувачів.
- Впроваджено збереження даних в базі даних за допомогою Entity Framework.

4. Тестування та оптимізація:

- Проведено ряд тестів для перевірки функціональності, продуктивності та безпеки інтернет-магазину.
- Здійснено оптимізацію завантаження сторінок та роботи з даними для забезпечення високої швидкості роботи магазину.

Отже, завдяки виконанню цієї дипломної роботи, було отримано практичні навички в розробці інтернет-магазинів з використанням сучасних технологій. Даний досвід буде корисним для подальшої професійної діяльності в області веб-розробки.

					<i>РП 06.18.000 ДП ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Angular University [Електронний ресурс] <https://angular-university.io/> (15.05.2023)
2. ASP.NET Core Documentation [Електронний ресурс] <https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-3.1> (16.05.2023)
3. Duckett, J. (2014). JavaScript and JQuery: Interactive Front-End Web Development. Wiley.
4. Fowler, M. (2012). Patterns of Enterprise Application Architecture. Addison-Wesley Professional.
5. Mardan, A. (2018). Full-Stack React, TypeScript, and Node. Apress.
6. Negrino, T., & Smith, D. (2018). JavaScript and jQuery: Visual QuickStart Guide. Peachpit Press.
7. Richardson, C., & Smith, R. (2019). Microservices Patterns: With examples in Java. Manning Publications.
8. ASP.NET Core документація [Електронний ресурс] <https://docs.microsoft.com/uk-ua/aspnet/core/?view=aspnetcore-5.0> (14.05.2023)
9. Веллі, К. Angular: Повне керівництво для розробників / К. Веллі, А. Балес. - Київ: Видавництво "БХВ-Петербург", 2020. - 564 с.
10. Крокфорд, Д. JavaScript: Хороші частини / Д. Крокфорд. - Львів: Видавництво Старого Лева, 2013. - 176 с.
11. Макдональд, М. ASP.NET Core 2.0 MVC и Razor Pages для начинающих: как самостоятельно изучить основы MVC для работы над реальными проектами / М. Макдональд. - Київ: Комп'ютерна література, 2018. - 610 с.
12. Основи Angular [Електронний ресурс] <https://angular.io/start> (12.05.2023)
13. Фрімен, Е. HTML і CSS. Навчаємося з головою / Е. Фрімен, Р. Робсон. - Київ: Комп'ютерна література, 2015. - 624 с.

					<i>РП 06.18.000 ДП ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		55

ДОДАТОК А. Лістинг візуалізації даних та обробки запитів(мова Angular)

```
<div
  class="d-flex justify-content-between position-relative w-100"
  style="min-width: 1000px; min-height: 500px; position: relative;"
  [@dialogExpand]="dialogProperty"
>
  <mat-tab-group
    mat-stretch-tabs="false"
    mat-align-tabs="start"
    class="menu-position w-100"
    [selectedIndex]="selectedTabIndex"
    style="max-height: 550px"
  >
    <mat-tab label="Товар">
      <div class="d-flex justify-content-between mt-2 mat-tab-p">
        <div class="me-4 d-flex flex-column justify-content-around">
          <img [src]="item.imgSrc" class="item-img">

          <div matTooltip="Рейтинг" class="d-flex justify-content-center my-3" style="color: gold">
            <ng-container *ngFor="let star of [1, 2, 3, 4, 5]; let i = index">
              <mat-icon [ngClass]="{ 'filled': i < (item.rating.number || 0) }">
                {{ i < (item.rating.number || 0) ? 'star' : 'star_outline' }}
              </mat-icon>
            </ng-container>
          </div>

          <div class="d-flex flex-row justify-content-around">
            <p> Автор: {{ item.author.login }} </p>
            <p> Видгуків: {{ item.comments.length }} </p>
            <p> Товар придбало: {{ item.numberOfPurchases }} </p>
          </div>

          <div class="d-flex justify-content-around">
            <button
              mat-raised-button
              style="border-radius: 10px"
```

```
    color="primary"
    (click)="onBuyItem()"
    [disabled]="isAuthorItem || !authorId"
  > Придбати цей товар </button>
```

```
<button
  mat-raised-button
  style="border-radius: 10px"
  color="primary"
  (click)="onAddToShoppingCard()"
  [disabled]="getQualityId() || !onUserAuth"
> Додати товар до кошика </button>
</div>
</div>
```

```
<div class="text-hyphens" style="max-height: 500px;">
  <h2 class="opacity-50"> Назва товару </h2>
  <h2 style="overflow: auto;"> {{ item.productName }} </h2>

  <h2 class="opacity-50"> Опис товару </h2>
  <p style="overflow: auto; max-height: 200px;"> {{ item.description }} </p>
</div>
```

```
</div>
</mat-tab>
```

```
<mat-tab label="Характеристики">
  <div class="mat-tab-p d-flex flex-column">
    <div class="d-flex flex-row justify-content-center">
      <h3> Характеристики до товару {{ item.productName }} </h3>
    </div>
  </div>
```

```
<div
  *ngFor="let property of item.properties; let i = index"
  class="d-flex flex-row justify-content-between pt-2 px-4"
  style="border-bottom: 1px solid #d5d5d5;"
  [ngStyle]="{ background: i % 2 == 0 ? '#f6f6f6': 'transparent' }"
```

```

>
  <p> {{ property.name }} </p>
  <p> {{ property.type }} </p>
</div>
</div>
</mat-tab>

<mat-tab label="Відгуки">
  <div class="mat-tab-p" style="max-height: 500px">
    <div class="d-flex justify-content-between align-items-center flex-row review-block w-100 mt-2">
      <h3 class="m-0"> Залишити відгук про товар </h3>
      <button
        mat-raised-button
        [matTooltip]="matToolTipText"
        style="border-radius: 10px"
        [disabled]="!checkIsProductPurchased || userAlreadyReviewed"
        (click)="isCreateReviewMode = true"
      > Залишити відгук </button>
    </div>

    <div
      class="d-flex justify-content-center mt-3 flex-column align-items-center"
      style="min-height: 200px"
    >
      <h2
        *ngIf="!item.comments.length && !isCreateReviewMode"
        class="opacity-75 font-weight-700"
      > На цей товар ще не було відгуків. </h2>

      <div *ngIf="isCreateReviewMode" class="d-flex flex-column w-100 review-block">
        <div matTooltip="Рейтинг" style="color: gold">
          <ng-container *ngFor="let star of [1, 2, 3, 4, 5]; let i = index">
            <mat-icon
              (click)="onSetRating(i + 1);"
              (mouseover)="onChangeRating(i + 1)"
              [ngClass]="{ 'filled': i < currentRating }"
            >

```

```

    {{ i < currentRating ? 'star' : 'star_outline' }} </mat-icon>
  </ng-container>
</div>

<mat-form-field appearance="outline">
  <mat-label> Ваш відгук </mat-label>
  <textarea matInput [(ngModel)]="comment"></textarea>
</mat-form-field>

<div class="d-flex flex-row justify-content-between p-0">
  <button
    mat-raised-button
    color="primary"
    style="border-radius: 10px;"
    [disabled]="!comment.length"
    (click)="onSendComment()"
  > Відправити </button>

  <button
    mat-raised-button
    color="warn"
    (click)="isCreateReviewMode = false"
    style="border-radius: 10px;"
  > Скасувати </button>
</div>
</div>

<div *ngIf="item.comments.length" class="w-100">
  <div *ngFor="let comment of item.comments" class="review-block my-3 w-100">
    <div class="d-flex justify-content-between"> <!-- USER INFO -->
      <div class="d-flex flex-row">
        <span> {{ comment?.author?.login }} </span>
      </div>

      <div matTooltip="Рейтинг" style="color: gold">
        <ng-container *ngFor="let star of [1, 2, 3, 4, 5]; let i = index">
          <mat-icon [ngClass]="{ 'filled': i < comment.rating }">

```

```

        {{ i < comment.rating ? 'star' : 'star_outline' }}
    </mat-icon>
</ng-container>
</div>

    <span> {{ comment.dateCreate | date: 'M/d/y, h:mm' }} </span>
</div>

<hr>

<div>
    {{ comment.commentText }}
</div>
</div>
</div>
</div>
</div>

<mat-form-field *ngIf="false" appearance="outline" class="w-100 my-2">
    <mat-label> Textarea </mat-label>
    <input matInput>
    <button class="action-btn position-absolute" style="right: 0; top: 50%">
        <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 384 512">
            <path
                d="M73 39c-14.8-9.1-33.4-9.4-48.5-.9S0 62.6 0 80V432c0 17.4 9.4 33.4 24.5 41.9s33.7 8.1 48.5-.9L361
                297c14.3-8.7 23-24.2 23-41s-8.7-32.2-23-41L73 39z" />
            </svg>
        </button>
    </mat-form-field>
</div>
</mat-tab>

<mat-tab label="Запитання">
    <div class="d-flex flex-column justify-content-center mat-tab-p">
        <div class="d-flex flex-row align-items-center justify-content-center review-block w-100 mt-2">
            <button
                mat-raised-button
                style="border-radius: 10px; padding: 10px;"

```

```
(click)="toggleCreateQuestionMode()"
[color]="!isCreateQuestionMode ? 'default' : 'warn'"
[disabled]="!authorId"
> {{ !isCreateQuestionMode ? 'Поставити своє питання щодо товару' : 'Скасувати' }} </button>
</div>
```

```
<div
  *ngIf="!item.questions.length && !isCreateQuestionMode"
  class="d-flex justify-content-center align-items-center"
> <h3 class="py-3 "> До цього товару не було питань </h3> </div>
```

```
<div *ngIf="isCreateQuestionMode" class="d-flex flex-column w-100 review-block w-100 mt-4">
  <mat-form-field appearance="outline">
    <mat-label> Ваше питання </mat-label>
    <textarea matInput [(ngModel)]="questionText"></textarea>
  </mat-form-field>
```

```
<div class="d-flex flex-row-reverse">
  <button
    mat-raised-button
    style="border-radius: 10px;"
    (click)="onCreateQuestion()"
    [disabled]="!questionText.length"
    color="primary"
  > Відправити </button>
</div>
```

```
</div>
```

```
<div *ngFor="let question of item.questions; let questionIndex = index" class="d-flex flex-column"
style="max-height: 500px">
```

```
<div class="d-flex flex-column review-block w-100 mt-4 answer-box">
  <div class="d-flex justify-content-between flex-row pb-2" style="border-bottom: 1px solid #8080806b;">
    <span> {{ question.author.login }} </span>
    <span> {{ question.dateCreate | date: 'M/d/y, h:mm' }} </span>
  </div>
```

```

<div class="d-flex flex-column w-100 mt-2">
  <div> {{ question.messageText }} </div>

  <div class="d-flex justify-content-between align-items-center">
    <div class="d-flex justify-content-center flex-grow-1">
      <button
        mat-icon-button
        [disabled]="!question.answers.length"
        (click)="question.isAnswerReviewMode = !question.isAnswerReviewMode"
      >
      <mat-icon
        *ngIf="!question.isAnswerReviewMode"
        [matBadge]="question.answers.length"
        matBadgeSize="small"
      > keyboard_arrow_down </mat-icon>

      <mat-icon
        *ngIf="question.isAnswerReviewMode"
        [matBadge]="question.answers.length"
        matBadgeSize="small"
      > keyboard_arrow_up </mat-icon>
    </button>
  </div>

  <div class="d-flex flex-row-reverse">
    <button
      mat-raised-button
      style="border-radius: 10px;"
      (click)="toggleCreateAnswerMode(questionIndex)"
      [color]="!question.isCreateAnswerMode ? 'primary' : 'warn'"
      [disabled]="!authorId"
    > {{ !question.isCreateAnswerMode ? 'Дати відповідь' : 'Скасувати' }} </button>
  </div>
</div>

<div *ngIf="question.isCreateAnswerMode" class="review-block w-100 mt-2">

```



```

<div
  class="header-gradient header-menu"
  cdkDrag
  cdkDragRootElement=".cdk-overlay-pane"
  cdkDragHandle
> </div>

<button mat-dialog-close class="btn-item position-absolute" style="right: 0; z-index: 100">
  <svg class="modal-icon" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 384 512">
    <path
      d="M342.6 150.6c12.5-12.5 12.5-32.8 0-45.3s-32.8-12.5-45.3 0L192 210.7 86.6 105.4c-12.5-12.5-32.8-12.5-45.3 0s-12.5 32.8 0 45.3L146.7 256 41.4 361.4c-12.5 12.5-12.5 32.8 0 45.3s32.8 12.5 45.3 0L192 301.3 297.4 406.6c12.5 12.5 32.8 12.5 45.3 0s12.5-32.8 0-45.3L237.3 256 342.6 150.6z"/>
    </svg>
  </button>
</div>

```

```

import { ChangeDetectorRef, Component, Inject, OnInit, ViewChild } from '@angular/core';
import { CommonModule } from '@angular/common';

import { MatPaginator, MatPaginatorModule, PageEvent } from '@angular/material/paginator';
import { MatButtonModule } from '@angular/material/button';
import { MatIconModule } from '@angular/material/icon';
import { MatRippleModule } from '@angular/material/core';
import { MatTooltipModule } from '@angular/material/tooltip';
import { animate, state, style, transition, trigger } from '@angular/animations';
import { MatInputModule } from '@angular/material/input';
import { DragDropModule } from '@angular/cdk/drag-drop';

import { MAT_DIALOG_DATA, MatDialog, MatDialogModule } from '@angular/material/dialog';
import { CreateItemDialogComponent } from '../create-item/create-item-dialog.component';
import { ProductItem } from '../models/product-item';
import { AuthenticationService } from '../services/authentication.service';
import { MatTabsModule } from '@angular/material/tabs';
import { IUser } from '../interfaces/user';
import { PurchaseService } from 'src/services/purchase.service';

```

```
import { PurchaseOfGoodsComponent } from '../purchase-of-goods/purchase-of-goods.component';
import { ShoppingCartService } from 'src/services/shopping-cart.service';
import { FormsModule } from '@angular/forms';
import { Subscription } from 'rxjs';
import { MatBadgeModule } from '@angular/material/badge';
```

```
@Component({
  selector: 'app-goods',
  templateUrl: './item-info.component.html',
  styleUrls: ['./item-info.component.css'],
  standalone: true,
  imports: [
    CommonModule,
    MatPaginatorModule,
    MatButtonModule,
    MatIconModule,
    MatRippleModule,
    MatTooltipModule,
    MatInputModule,
    MatTabsModule,
    MatDialogModule,
    DragDropModule,
    FormsModule,
    MatBadgeModule
  ],
  animations: [
    trigger('fadeInOut', [
      state(
        'void',
        style({
          opacity: 0,
        })
      ),
      transition('void <=> *', animate(300)),
    ]),
    trigger('dialogExpand', [
      state('expanded', style({
```

```

    height: 'fit-content',
    width: 'fit-content'
  )),
  transition('void => expanded', [
    style({ height: '0', width: '0' }),
    animate('300ms ease-in')
  ]),
  transition('expanded => void', [
    animate('300ms ease-out', style({ height: '0', width: '0' })))
  ])
],
})

```

```

export class ItemInfoComponent implements OnInit {

```

```

  readonly subscription = new Subscription();
  @ViewChild(MatPaginator) paginator!: MatPaginator;
  items: ProductItem[] = [];
  pagedItems = this.items.slice(0, 9);
  isExpanded!: string;
  rating: boolean[] = [true, true, false, false, false];
  authorId!: string | number;
  item!: ProductItem;
  user!: IUser;
  checkUser!: IUser;

  currentRating = 0;
  selectedTabIndex = 0;

  dialogProperty = 'expanded';

  comment = "";
  questionText = "";
  answerText = "";

  isCreateReviewMode = false;
  isCreateQuestionMode = false;

```

```
isCreateAnswerMode = false;
```

```
isAnswerReviewMode = false;
```

```
answersCreate: { answerId: number }[] = [];
```

```
constructor(
```

```
  private dialog: MatDialog,
```

```
  private authenticationService: AuthenticationService,
```

```
  @Inject(MAT_DIALOG_DATA)
```

```
  private data: { product: ProductItem; selectedTabIndex: number },
```

```
  private purchaseService: PurchaseService,
```

```
  private shoppingCartService: ShoppingCartService,
```

```
  private changeDetector: ChangeDetectorRef
```

```
) {}
```

```
ngOnInit() {
```

```
  this.checkUser = this.purchaseService._currentUser;
```

```
  this.item = this.data.product;
```

```
  this.subscription.add(this.purchaseService.product.subscribe(
```

```
    (item) => {
```

```
      if (item) {
```

```
        this.changeDetector.detectChanges();
```

```
      }
```

```
      this.item = item;
```

```
    }
```

```
  ));
```

```
  this.selectedTabIndex = this.data.selectedTabIndex ?? 0;
```

```
  this.authorId = this.authenticationService.getAccountId();
```

```
  this.initMyGoods();
```

```
  this.subscription.add(this.purchaseService.user.subscribe(user => this.checkUser = user));
```

```
  this.item.questions.map(question => question.isCreateAnswerMode = false);
```

```
}
```

```
onAddItem() {
```

```
  this.onPageChange({
```

```
    pageIndex: this.paginator.pageIndex,
```

```
    pageSize: this.paginator.pageSize,
```

```

    length: this.items.length,
  });
}

onPageChange(event: PageEvent) {
  const startIndex = event.pageIndex * event.pageSize;
  this.pagedItems = this.items.slice(startIndex, startIndex + event.pageSize);
}

onCreateItem() {
  const ref = this.dialog.open(CreateItemDialogComponent, {
    width: '30%',
  });
}

private initMyGoods() {
  this.items = JSON.parse(localStorage.getItem('items') ?? []).filter(
    (item: ProductItem) => item.author?.id === this.authorId
  );
  this.pagedItems = this.items.slice(0, 9);
}

getRating(item: ProductItem): number {
  return item?.rating?.number ?? 0;
}

get checkIsProductPurchased(): boolean {
  let purchasedItem;
  if (this.checkUser) {
    purchasedItem = this.checkUser.purchasedGoods.find(item => item.productId === this.item.productId);
  }
  return !!purchasedItem;
}

get matToolTipText(): string {
  let purchasedItem;
  if (this.checkUser) {

```

```
    purchasedItem = this.checkUser.purchasedGoods.find(item => item.productId == this.item.productId);
  }
  if (purchasedItem?.isReviewed) {
    return 'Ви вже ставили оцінку';
  } else if (!purchasedItem) {
    return 'Щоб залишити відгук потрібно купити товар';
  } else {
    return '';
  }
}
```

```
onBuyItem() {
  const ref = this.dialog.open(PurchaseOfGoodsComponent);
  ref.componentInstance.item = this.item;
  ref.afterClosed().subscribe(() => this.purchaseService._currentUser);
}
```

```
getQualityId(): boolean {
  const users: IUser[] = JSON.parse(localStorage.getItem('users') ?? '');
  const accountId = JSON.parse(localStorage.getItem('accountId') ?? '0');
  const user = users.find(user => user.id == accountId);

  const isAlreadyExist = user?.shoppingCart.find((item) => item.productId == this.item.productId);
  return this.item.author?.id == accountId || !!isAlreadyExist;
}
```

```
get isAuthorItem(): boolean {
  return this.item.author.id == JSON.parse(localStorage.getItem('accountId') ?? '0');
}
```

```
get onUserAuth(): boolean {
  return !!this.authenticationService.getAccountId();
}
```

```
onAddToShoppingCard() {
  this.shoppingCartService.addToCart(this.item);
}
```

```
}
```

```
onSendComment() {  
  this.purchaseService.reviewItem(this.item, this.comment, this.currentRating);  
  this.comment = "";  
  this.isCreateReviewMode = false;  
}
```

```
onChangeRating(rating: number) {  
  this.currentRating = rating;  
}
```

```
onSetRating(rating: number) {  
  this.currentRating = rating;  
}
```

```
get userAlreadyReviewed(): boolean {  
  const flag = this.item.comments.find((user) => user.author.id === this.checkUser.id);  
  return !!flag;  
}
```

```
onCreateQuestion() {  
  this.isCreateQuestionMode = false;  
  this.purchaseService.askQuestion(this.item, this.questionText);  
  this.questionText = "";  
}
```

```
onCreateAnswerForQuestion(questionId: number, questionIndex: number, text: string) {  
  this.isCreateAnswerMode = false;  
  this.purchaseService.askAnswer(this.item, questionId, text);  
  this.item.questions[questionIndex].tempQuestionText = "";  
  this.item.questions[questionIndex].isCreateAnswerMode = false;  
}
```

```
toggleCreateAnswerMode(questionIndex: number) {  
  this.item.questions[questionIndex].isCreateAnswerMode =  
  !this.item.questions[questionIndex].isCreateAnswerMode;  
}
```

```
// this.isCreateAnswerMode = !this.isCreateAnswerMode;
this.answerText = "";
}

toggleCreateQuestionMode() {
  this.isCreateQuestionMode = !this.isCreateQuestionMode;
  this.questionText = "";
}
}
```

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ
КОЛЕДЖ ОДЕСЬКОГО НАЦІОНАЛЬНОГО
ТЕХНОЛОГІЧНОГО УНІВЕРСИТЕТУ»

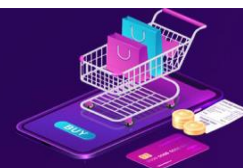
Розробка інтернет-магазину з продажу комп'ютерної техніки

Студента групи 4РП-06
Прокопчука Андрія Михайловичка

Мета проекту

Мета цього дипломного проекту полягає в розробці функціонального інтернет-магазину з продажу комп'ютерної техніки за допомогою сучасних технологій. Для досягнення цієї мети необхідно:

1. Проаналізувати вимоги до функціональності та дизайну інтернет-магазину.
2. Розробити серверну частину за допомогою ASP.NET, що буде обробляти запити від клієнтів, керувати даними користувача та обробляти замовлення.
3. Розробити клієнтську частину за допомогою Angular, що буде відповідальною за інтерфейс користувача та взаємодію з сервером.
4. Створити інтерфейс користувача за допомогою HTML та CSS, включаючи головну сторінку, каталог товарів, сторінку товару, кошик та систему авторизації та реєстрації користувачів.
5. Реалізувати систему авторизації та реєстрації користувачів для забезпечення безпечного доступу до особистих даних та управління замовленнями.
6. Провести тестування всієї системи, щоб переконатися в її коректній роботі та відсутності помилок.



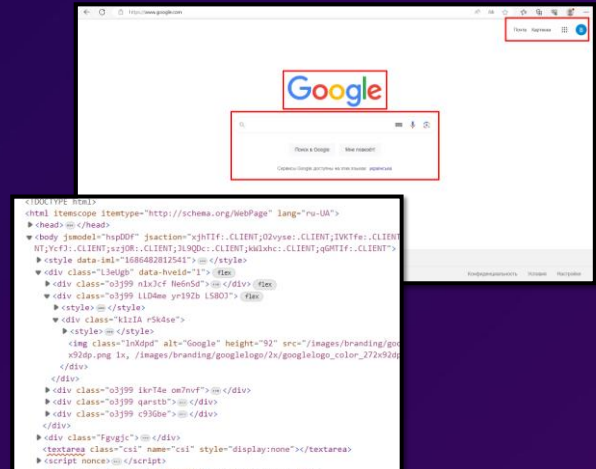
Що таке веб-сторінка та її складові?



Веб-сайт складається з багатьох елементів, кожен зі своєю унікальною функцією і призначений для поліпшення користувацького досвіду.

Сторінки сайтів - це файли з текстом, розмічених мовою HTML. Ці файли, будучи завантаженими відвідувачем на його комп'ютер, обробляються браузером і виводяться на його засіб відображення (монітор, екран КПК, принтер).

Сторінки сайтів можуть бути простим статичним набором файлів або створюватися спеціальною комп'ютерною програмою на сервері - так званим движком сайту. Движок може бути або зроблений на замовлення для окремого сайту, або бути готовим продуктом, розрахованим на певний клас сайтів. Деякі з движків можуть забезпечити власнику сайту можливість гнучкої настройки структуризації і виведення інформації на веб-сайті. Такі движки називаються системами управління змістом (CMS).

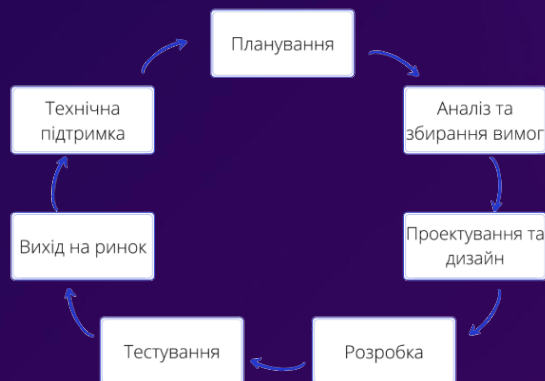


Планування та побудова проекту



Сучасні засоби проектування (СП) можна розділити на великі категорії. Першу складають CASE-системи (як незалежні (upper CASE), так і інтегровані з СУБД), що забезпечують проектування БД та додатків у комплексі з інтегрованими засобами розробки додатків "клієнт-сервер". Інструменти цієї категорії, як правило, мають суттєву складність, широкую сферу застосування та високу гнучкість

Другу категорію складають власне кошти проектування БД, що реалізують ту чи іншу методологію, як правило, "сутність-зв'язок" ("entity relationship") і розглядаються в комплексі із засобами розробки додатків. До засобів цієї категорії можна віднести такі, SILVERRUN+JAM, ERwin/ERX+PowerBuilder та ін.



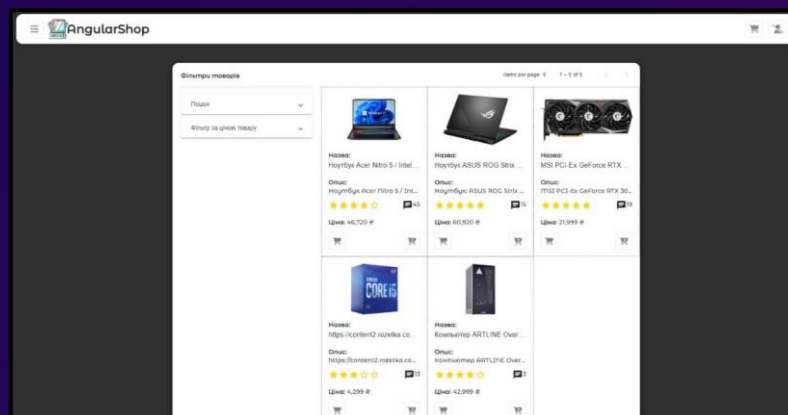
Структура веб-додатку



Структура інтернет-магазину складається з таких функціональних елементів:

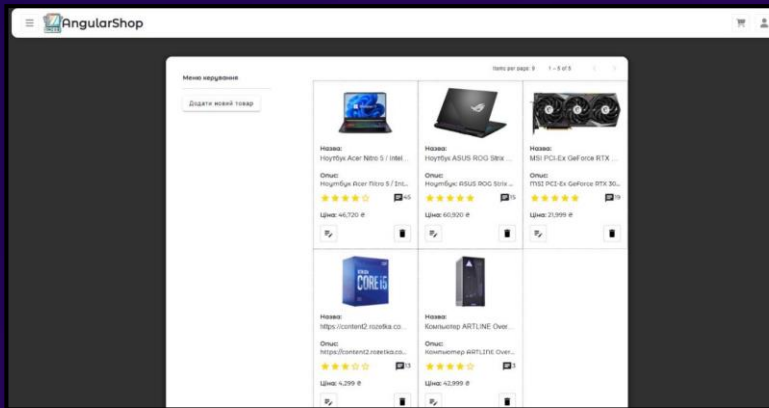
- Каталог товарів;
- Користувальницький кошик;
- Реєстраційна форма;
- Форма відправлення замовлення.

Структура веб-додатку

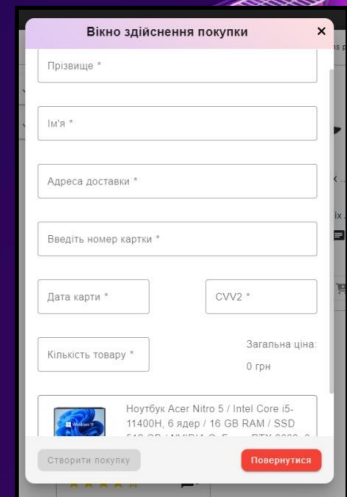


Головна сторінка сайту

Структура веб-додатку

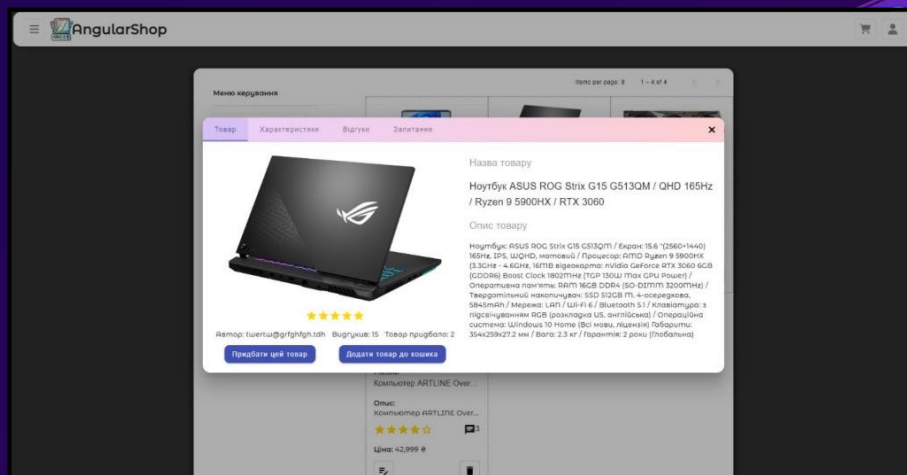


Каталог товару користувача



Вікно здійснення покупки

Структура веб-додатку



Сторінка товару

Структура веб-додатку



- ASP.NET Identity - це гнучка система, що дозволяє управляти користувачам та ролями в додатку. Вона надає API для реєстрації, авторизації, змін паролів, відновлення забутих паролів та ін.
- JWT - це стандарт для безпечної передачі даних між двома сторонами через JSON об'єкт. В нашому випадку, після авторизації на сервері, користувач отримує JWT, який він повинен відправляти з кожним запитом до API для підтвердження своєї ідентичності.

Структура веб-додатку

Two side-by-side screenshots of web forms. The left form is titled 'Авторизація' (Authorization) and contains three input fields: 'Логін *' (Login), 'Електронна адреса *' (Email), and 'Пароль *' (Password), each with a '0 / 50' character count. At the bottom are buttons 'Авторизуватися' (Login) and 'Повернутись' (Back). The right form is titled 'Реєстрація' (Registration) and contains four input fields: 'Логін *', 'Електронна адреса *', 'Пароль *', and 'Повторіть пароль *', each with a '0 / 50' character count. At the bottom are buttons 'Зареєструватися' (Register) and 'Повернутись' (Back).

Форми авторизації та реєстрації

Висновки



У процесі виконання дипломної роботи було розроблено функціональний інтернет-магазин комп'ютерної техніки. Завдяки використанню сучасних технологій, таких як ASP.NET для серверної частини та Angular для клієнтської частини, було створено ефективний і безпечний сервіс для продажу комп'ютерної техніки.

Розробка включала створення різних компонентів сайту, включаючи головну сторінку, каталог товарів, сторінку товару, кошик та систему авторизації та реєстрації користувачів. Кожен з цих компонентів було розроблено з урахуванням особливостей роботи інтернет-магазину та вимог до зручності користувачів.

Проведене тестування дозволило переконатися в стабільності роботи розробленого інтернет-магазину та відсутності помилок. Система авторизації та реєстрації користувачів забезпечує безпечний доступ до особистих даних та управління замовленнями.

Отже, завдяки виконанню цієї дипломної роботи, було отримано практичні навички в розробці інтернет-магазинів з використанням сучасних технологій. Даний досвід буде корисним для подальшої професійної діяльності в області веб-розробки

Ім'я користувача:
Наталія Вікторівна Копусь

ID перевірки:
1015553693

Дата перевірки:
12.06.2023 09:07:29 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
12.06.2023 09:09:58 EEST

ID користувача:
100011688

Назва документа: 4РП-06 Прокопчук А.М

Кількість сторінок: 81 Кількість слів: 12116 Кількість символів: 96803 Розмір файлу: 8.80 MB ID файлу: 1015205877

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

18.8% Схожість

Найбільша схожість: 2.21% з Інтернет-джерелом (<https://studfile.net/preview/5535698/page:16>)

18.8% Джерела з Інтернету

1000

Сторінка 83

Не знайдено джерел з Бібліотеки

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

20

Підозріле форматування

16
сторінок

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Прокопчук Андрій Михайлович,
здобувач освіти гр. 4РП-06, та

Шувалова Ірина Олегівна,
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до випускної кваліфікаційної роботи молодшого спеціаліста на тему:

«Розробка інтернет-магазину з продажу комп'ютерної техніки» (автор роботи – Прокопчук А.М., керівник роботи – Шувалова І.О.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2023 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.


Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи, і даємо згоду на обробку персональних даних.

Виконавець



/ Прокопчук А.М. /

Керівник



/ Шувалова І.О. /

« 09 » 06 20 23 р.

ВІДГУК

керівника на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Прокопчук Андрій Михайлович

(прізвище, ім'я та по батькові)

Спеціальність: 121 - Інженерія програмного забезпечення

Освітня програма: Розробка програмного забезпечення

Тема дипломного проекту: Розробка інтернет-магазину з продажу комп'ютерної
техніки

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки) _____

Пояснювальна записка складена відповідно до стандартів ДСТУ. Обсяг та зміст роботи відповідають поставленим цілям дипломного проектування. Розрахунково-пояснювальний розділ проекту характеризується чіткою структурою та високою якістю виконання.

б) самостійність роботи над проектом: _____

Прокопчук А.М. ефективно зайнявся опрацюванням наукового матеріалу, провів глибокий аналіз методик та стратегій розробки веб-орієнтованих систем, продемонстрував уміння критично аналізувати літературні джерела та вдало використовував новітні інформаційні технології під час досліджень.

в) теоретична підготовка випускника (випускниці): _____

Студент Прокопчук А.М. має солідну теоретичну та практичну базу на високому рівні. Він проявляє високу концентрацію, пильність та відповідальність у своїй роботі.

г) вміння розв'язувати виробничі та конструкторські питання _____

Студент Прокопчук А.М. за період роботи показав Високий рівень теоретичної та

практичної підготовки

Оцінка розрахункової частини «Добре»

Оцінка графічної частини «Добре»

Загальна оцінка «Добре»

Прізвище, ім'я, по батькові керівника дипломного проекту _____

Шувалова Ірина Олегівна

Місце роботи і посада керівника дипломного проекту _____

Підпис _____

« 09 » 06 2023 р.

РЕЦЕНЗІЯ

на дипломний проект (роботу) здобувача (здобувачки) освіти
відділення комп'ютерних систем

Прокопчук Андрій Михайлович

(прізвище, ім'я та по батькові)

Спеціальність 121 “Інженерія програмного забезпечення”

Освітня програма «Розробка програмного забезпечення»

Керівник дипломного проекту (роботи) Шувалова Ірина Олегівна

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) Розробка інтернет-магазину з продажу
комп'ютерної техніки

Обсяг розрахунково-пояснювальної записки 77 сторінок

Обсяг графічної (презентаційної) частини 10 аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту (роботи) завданню
Дипломний проект відповідає поставленим завданням і включає розробку сучасного інтернет-магазину комп'ютерної техніки з використанням передових технологій.

б) характеристика виконання кожного розділу дипломного проекту (роботи) _____
У кожному розділі дипломного проекту виконано глибокий аналіз, розробка інтерфейсу, вибір технологій та розробка функціоналу інтернет-магазину. Зокрема, було розроблено структуру бази даних, дизайн сайту, систему авторизації та алгоритми обробки даних.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту (роботи) _____
Пояснювальна записка має чітку структуру, логічно викладені матеріали та високу якість графічних матеріалів. Графічна частина включає схеми, діаграми та ілюстрації, які допомагають краще зрозуміти суть проекту.

г) перелік позитивних якостей дипломного проекту (роботи) _____

До позитивних якостей дипломного проекту належать: високий рівень розробки, актуальність теми, застосування сучасних технологій, наявність детального аналізу ринку, який допомагає зрозуміти потенціал проекту.

д) основні недоліки дипломного проекту (роботи) _____

Проект може мати обмежену масштабованість і потребуватиме подальшого тестування для впевненості в його стабільності при великому навантаженні.

Оцінка розрахункової частини _____ добре

Оцінка графічної частини _____ добре

Загальна оцінка _____ добре

Прізвище, ім'я, по батькові рецензента _____ Кривченко Юрій Вікторович

Місце роботи і посада рецензента _____

ВСП "Одеський технічний фаховий коледж ОНТУ", голова циклової комісії комп'ютерних технологій та програмної інженерії

Підпис: _____

« 16 » _____ червня _____ 2023 р.