

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

Інститут електроенергетики  
(інститут)

Факультет інформаційних технологій  
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем  
(повна назва)

**ПОЯСНЮВАЛЬНА ЗАПИСКА**  
**кваліфікаційної роботи ступеня**  
**бакалавра**

(назва освітньо-кваліфікаційного рівня)

студента *Соломатіна Владислава Максимовича*  
(ПІБ)

академічної групи *122-19ск-2*  
(шифр)

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

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

на тему: *Розробка web-сайту інтернет - магазину*  
*електротоварів «Лайтон» з використанням технології Note JS*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	<i>доц. Кабак Л.В.</i>			
<b>розділів:</b>				
спеціальний	<i>доц. Кабак Л.В.</i>			
економічний	<i>доц. Касьяненко Л.В.</i>			
<b>Рецензент</b>	<i>доц. Шедловський І.А.</i>			
<b>Нормоконтролер</b>	<i>доц. Гуліна І.Г.</i>			

Дніпро  
2022

Міністерство освіти і науки України  
НТУ «Дніпровська політехніка»

**ЗАТВЕРДЖЕНО:**

завідувач кафедри  
програмного забезпечення комп'ютерних систем

(повна назва)

І.М. Удовик

(підпис)

(прізвище, ініціали)

«    »                      2022 року

**ЗАВДАННЯ**

на кваліфікаційну роботу  
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента 122-19ск-2 Соломатіна Владислава Максимовича  
(група) (прізвище та ініціали)

тема кваліфікаційної роботи Розробка web-сайту інтернет - магазину  
електротоварів «Лайтон» з використанням технології Note JS

затверджена наказом ректора НТУ «ДП» від «18» травня 2022 р. № 268-с

Розділ	Зміст виконання	Термін виконання
Спеціальний	<i>На основі матеріалів практик та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми</i>	13.05.2022 р.
Економічний	<i>Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки</i>	27.05.2022 р.

Завдання видав

(підпис)

доц. Кабак Л.В.

(посада, прізвище, ініціали)

Завдання прийняв до виконання

(підпис)

Соломатін В.М.

(прізвище, ініціали)

Дата видачі завдання: 14.01.2022 р.

Термін подання кваліфікаційної роботи до ЕК: 13.06.2022 р.

## РЕФЕРАТ

Пояснювальна записка: 62 с., 20 рис., 1 табл., 3 дод., 20 джерел.

Об'єкт розробки: інтернет-магазин з продажу електротоварів.

Мета кваліфікаційної роботи: підвищення ефективності продажу електротоварів за рахунок розробки та використанню відповідного веб-додатку.

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

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

У другому розділі виконано аналіз існуючих рішень, обрано вибір платформи для розробки, виконано проектування і розробка програми, наведено опис алгоритму і структури функціонування підсистеми, визначені вхідні і вихідні дані, наведені характеристики складу параметрів технічних засобів, описаний виклик та завантаження застосунку, описана робота програми.

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

Практичне значення полягає в автоматизації процесу продажу електротоварів.

Актуальність даного типу додатків визначається великим попитом на подібні розробки.

Список ключових слів: ВЕБ-ДОДАТОК, ІНТЕРНЕТ-МАГАЗИН, CSS, JAVA SCRIPT, HTML.

## **ABSTRACT**

Explanatory note: 62 pp., 20 fig., 1 tab., 3 apps, 20 sources.

Object of development: online store for the sale of electrical goods.

The purpose of the qualification work: to increase the efficiency of sales of electrical goods through the development and use of a relevant web application.

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

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

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

The economic section determines the complexity of the developed information system, calculates the cost of work to create an application and calculates the time to create it.

The practical significance lies in the automation of the process of selling electrical goods.

The relevance of this type of application is determined by the high demand for such developments.

Keywords: WEB APPLICATION, ONLINE STORE, CSS, JAVA SCRIPT, HTML.

## ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT.....	4
СПИСОК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ.....	10
1.1 Загальні відомості з предметної галузі.....	10
1.2 Призначення розробки та область застосування.....	14
1.3 Підстава для розробки.....	14
1.4 Постановка завдання.....	15
1.5 Вимоги до програми або програмного виробу.....	16
1.5.1 Вимоги до функціональних характеристик .....	16
1.5.2 Вимоги до інформаційної безпеки.....	16
1.5.3 Вимоги до складу та параметрів технічних засобів.....	17
1.5.4 Вимоги до інформаційної та програмної сумісності.....	18
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	19
2.1 Функціональне призначення системи.....	19
2.2 Опис застосованих математичних методів.....	19
2.3 Опис використаних технологій та мов програмування.....	20
2.4 Опис структури системи та алгоритмів її функціонування.....	24
2.5 Обґрунтування та організація вхідних та вихідних даних програми.....	31
2.6 Опис роботи розробленої системи.....	31
2.6.1 Використані технічні засоби.....	31
2.6.2 Використані програмні засоби.....	32
2.6.3 Виклик та завантаження програми.....	37

2.6.4	Опис інтерфейсу користувача.....	37
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ.....		44
3.1	Розрахунок трудомісткості та вартості розробки програмного продукту .....	44
3.2	Розрахунок витрат на створення програми.....	48
ВИСНОВКИ.....		50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....		51
Додаток А. Код програми.....		53
Додаток Б. Відгук керівника економічного розділу.....		61
Додаток В. Перелік файлів на диску.....		62

## СПИСОК УМОВНИХ ПОЗНАЧЕНЬ

- БД - база даних;
- ЕОМ - електронно-обчислювальна машина;
- ІС - інформаційна система;
- ПЗ - програмне забезпечення;
- СУБД - система управління базами даних;
- CMS - система управління контентом;
- ІТ - інформаційні технології;
- SEO - пошукова оптимізація;
- CSS - Cascading Style Sheets (каскадні таблиці стилів).

## ВСТУП

Обрана тема вважається актуальною на сьогоднішній день, оскільки сьогодні, майже, всі люди щодня, не виходячи з дому, купують різні товари в інтернет магазинах. У світі, а зокрема в Україні величезними темпами зростає кількість користувачів всесвітної павутини, і як наслідок, кількість «електронних» покупців.

Інтернет-магазини істотно зменшують витрати виробника, заощадивши на утриманні звичайного магазину, розширюють ринки збуту, так само як і розширює можливість покупця - купувати будь-який товар в будь-якій країні, в будь-якому місті, в будь-який час доби, в будь-який час року. Це дає інтернет магазинам переваги перед звичайними «оффлайн» магазинами. Цей момент є істотним при переході виробників з «оффлайн» торгівлі на «Інтернет».

Висока якість продукції, вміння донести інформацію про продукт до споживача і ефективна система збуту робить підприємство успішним на ринку. Припустимо, що ці умови виконані, залишається - налагодити успішний процес продажів. Його складові також відомі - структурний підрозділ, вирішальне завдання збуту, кваліфікований працівники у цьому підрозділі, діюча система матеріальних і моральних стимулів і технологія збуту.

Уміння продавати вважається сьогодні мало не мистецтвом, даром згори. Тому щасливих продавців цінують, в їх роботу краще не втручатися, а основна частина співробітників відділу продажів змінюється з регулярністю, гідною подиву. Діяльність відділу продажів виглядає як "чорний ящик", де на вході - безліч контактів з потенційними клієнтами, а на виході - той чи інший фінансовий результат. У багатьох компаніях зустрічаються проблеми збуту, які заважають ефективно працювати відділу продажів, і не зникають навіть з підбором хороших продавців. Вирішити їх можна лише шляхом автоматизації процесу продажів. У вузькому та технологічному сенсі, під електронним бізнесом раніше розумілося використання інформаційних технологій (в першу чергу пов'язаних з Інтернетом) для організації взаємодії підприємства із



зовнішнім середовищем, включаючи постачальників, споживачів, партнерів і т.д. При такому підході електронний бізнес виступає, передусім, як досить складна прикладна інформаційна система. Більш широкий, або концептуальний, підхід розглядає електронний бізнес як спосіб підприємництва, що сприяє досягненню стратегічного успіху в нову інформаційну епоху. При такому розумінні електронний бізнес аж ніяк не зводиться до інформаційних технологій або активності в Інтернеті. Він зачіпає всі аспекти бізнесу, включаючи стратегію, процеси, організацію і технологію, і виводить його далеко за сформовані кордони.

# РОЗДІЛ 1

## АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

### 1.1. Загальні відомості з предметної галузі

Сучасні інформаційні технології все більше проникають у життя кожної людини, отже, і в сферу соціально-економічних відносин. Найбільш яскравим прикладом може служити Інтернет, застосовується у всіх сферах людської діяльності. Сучасний стан мережі дозволяє розглядати її як нову інформаційну інфраструктуру бізнесу, що дозволяє підприємствам розвивати традиційні методи маркетингової активності на базі сучасних інформаційно-комунікаційних технологій, тим самим наближаючи свою діяльність до споживача. Можливості Інтернету дозволяють розширити торговий простір, формують нові способи комунікації зі споживачами, створюють інноваційні способи просування продукції. Іншими словами, Інтернет може розглядатися як місце і інструмент для створення не тільки окремих додаткових послуг (фінансових, інформаційних, консалтингових та ін). але і відносно нових видів бізнесу, наприклад в галузі електронної комерції.

Таким чином, інтернет-маркетинг можна визначити як сукупність методів і засобів організації і здійснення ринкової діяльності підприємства за всіма складовими комплексу маркетингу в інформаційній мережі в цілях отримання бажаного відгуку від цільової аудиторії і задоволення споживачів. Інтернет-маркетинг охоплює такі функції, як маркетингові дослідження, розробка і просування продукції, зв'язку з громадськістю та ін.

Інтернет відкриває можливості доступу до інформації та думки споживача в набагато більшому обсязі, ніж традиційні способи маркетингової діяльності. Простий аналіз сайтів конкурентів може дати принципово важливе розуміння їх стратегій, плюсів і мінусів позиціонування і методів просування. Моніторинг ринку, ознайомлення з великими обсягами маркетингової інформації в зручний

час і практично в будь-якому місці зробив Інтернет унікальним інструментом для побудови або розвитку вже існуючого бізнесу.

Інтернет надає користувачам великі можливості по розповсюдженню, отримання та аналізу інформації, а також вироблення та прийняття рішень. Наприклад, для маркетологів - це можливість комунікації з споживачами, налагодження зворотного зв'язку, просування товарів і вивчення цільової аудиторії.

Сайт - сукупність електронних документів (файлів) приватної особи або організації в комп'ютерній мережі, об'єднаних під одним адресою (доменним ім'ям або IP-адресою).

Офіційної класифікації сторінок в Інтернеті не існує, але є основні характеристики, за якими можна розділити існуючі сайти, які слугують діловому світові.

Сайт-візитка – набір інформації, поширення якої в Інтернеті, безсумнівно, корисно як починаючим, так і акулам бізнесу. Це своєрідний довідник про фірму, який містить всі необхідні контактні дані та інформацію про діяльність компанії. Це повинна бути коротка, чітка і найважливіша інформація про Вас, Вашу сферу діяльності, інтереси. Не варто перевантажувати Ваш сайт-візитку інформацією, яка можливо не знадобиться потенційному клієнтові або партнерові.

Сайт-візитка – це:

- зручний засіб для сканування ґрунту ринку;
- недорого;
- швидко;
- вигідний для організацій, у яких бізнес здійснюється не в Інтернеті.

Корпоративний сайт – є більш серйозним виглядом представництва компанії в Інтернеті. Це обов'язковий інструмент успішного бізнесу будь-якої компанії. У першу чергу, сайти подібного роду допомагають формувати позитивний імідж компанії, залучають нових замовників, які можуть не знати про існування Вашої фірми на ринку, і відповідно, розширюють ринок збуту компанії. За допомогою корпоративного сайту клієнти та замовники отримують

інформацію про ціни на товари і послуги в зручному вигляді. Значно збільшити прибуток Ви зможете шляхом надання можливості робити замовлення Вашим клієнтам прямо в Інтернеті, на Вашій сторінці.

Отже, корпоративний сайт – це:

- доказ серйозності і перспективності Вашого бізнесу;
- гідне представництво Вашої компанії на ринку;
- найдешевший і наймасовіший вид реклами Вашого бізнесу;
- зручно і комерційно виправдано.

Промо-сайт – є прекрасним рекламним інструментом. Особливо, якщо Ви плануєте провести рекламну компанію окремого товару або послуги. Структура промо-сайту сама по собі повинна виконувати функцію рекламного проспекту або буклету, і привертати увагу клієнтів. Також на такому сайті можна розмістити ту інформацію, яка не поміститься на сторінці журналу, в радіоповідомленні, телевізійному ролику. Обсяг промо-сайту - до 10 сторінок, на яких можна розмістити дані про компанію, напрямок діяльності, просування послуги, контактні дані. Окремо повідомляється інформація про акцію, із зазначенням термінів і місця проведення, умов участі, новин та ін.

Промо-сайт – це:

- використання можливостей інтернету для впливу на цільову аудиторію;
- донесення до потенційних клієнтів максимум інформації про товар і послугу;
- потужний й ефективний інструмент реклами компанії та її продуктів.

Сайт-вітрина – є бюджетним варіантом реклами Вашої компанії. Якщо Ви хочете продавати в Інтернеті один конкретний товар або групу товарів, то цей тип сайтів ідеально підходить для Вас. На таких сторінках знаходиться вся інформація, необхідна для того, щоб клієнт побачив, зацікавився і купив даний товар, не відволікаючись на новини та посилання. Це структурований каталог продукції Вашої компанії. Але цей каталог організований таким чином (фото,

опис, ціна), щоб клієнт зробив як можна менше кліків, для того, щоб знайти потрібний товар. Головне призначення сайту-вітрини - продавати.

Сайт-вітрина – це:

- максимальний прибуток при мінімальному вкладенні сил та засобів;
- стабільний дохід;
- позитивний імідж компанії.

Інтернет – магазин – це сайт, де клієнт може не тільки побачити наявність всіх товарів, а може ще й зробити замовлення, вибрати варіант розрахунку, спосіб отримання замовлення та одержати рахунок на оплату. Такий сайт має бути зручним і функціональним, щоб відвідувач міг легко знайти те, що йому потрібно, відправити товар у кошик та оформити покупку в кілька кліків. Інтернет-магазин істотно зменшить Ваші витрати, так, як не треба орендувати приміщення для магазину, не треба платити заробітну плату співробітникам. Серед інших переваг: величезна аудиторія, можливість розміщувати необмежений асортимент товарів, своєчасно реагувати на зміни на ринку, враховувати потреби клієнтів та інші.

Інтернет – магазин – це:

- нові можливості для бізнесу;
- розширення ринку збуту;
- залучення широкої аудиторії клієнтів;
- скорочення витрат.

Крім класифікації сайтів за характеристиками можна розрізняти веб-сторінки і за складністю. Всього існує два види: статичний або динамічний сайт.

Так, статичним називають сайт з набором постійних, незмінних сторінок. Наприклад, сайт-візитка або сайт-вітрина. Інформація на таких сторінках жорстко закріплена, і зміни можуть здійснюватися тільки фахівцем у веб-технологіях. Такі сайти швидко завантажуються, можуть переглядатися в браузері, мають мінімальні вимоги до веб- сервера. Для компаній такі сайти - бюджетне рішення. Але для підтримки та актуалізації інформації на сторінці все одно будуть потрібні фахівці.

У цьому плані, динамічний сайт має безліч переваг. Для функціонування подібних веб-сторінок використовується система CMS (система управління змістом). Це забезпечення, яке дозволяє оновлювати інформацію, публікувати і змінювати її без залучення фахівців. Досить провести навчання персоналу лише в області порядку роботи з системою.

Загалом, вирішуйте самі, який сайт підходить Вашій компанії, які цілі і завдання він повинен буде виконувати. У будь-якому випадку, сторінка в Інтернеті незалежно від своїх характеристик буде приносити додатковий або основний прибуток у бізнесі.

## **1.2. Призначення розробки та галузь застосування**

Сайт призначено для просування на ринку продукції та послуг, що представлені у веб-додатку.

Функціонал сайту досить простий, але в той же час задовольняє всім вимогам що висуваються до веб-ресурсів подібного типу. Як правило, такі сайти не потребують значного оновлення структури, тому призначенням є можливість оновлення лише каталогу товарів та проведення заходів компанії.

Інтернет-магазин призначено для торгової компанії, яка хоче реально управляти процесом інтернет-торгівлі, маркетинговими акціями, торгувати і на замовлення, і зі складу, зменшити число менеджерів з продажу та підвищити об'єми продажів та інформування клієнтів про характеристики товарів.

Отже область застосування розробки кваліфікаційної роботи є глобальна мережа Інтернет.

## **1.3. Підстава для розробки**

Підставами для розробки (виконання кваліфікаційної роботи) є:

- освітня програма спеціальності 122 «Комп'ютерні науки»;
- навчальний план та графік навчального процесу;

- наказ ректора Національного технічного університету «Дніпровська політехніка» № 268-с від 18.05.2022 р;
- завдання на кваліфікаційну роботу на тему: «Розробка web-сайту інтернет - магазину електротоварів «Лайтон» з використанням технології Note JS».

#### **1.4. Постановка завдання**

Темою кваліфікаційної роботи є «Розробка web-сайту інтернет - магазину електротоварів «Лайтон» з використанням технології Note JS».

Розробка програмного забезпечення призначеного для надання універсального інструменту для відображення контенту (текстового та графічного) даного інтернет-магазину та формування замовлень.

На сайті необхідно реалізувати наступні пункти:

- відображення товарів для клієнта;
- формування накладної на продаж;
- видалення товару з накладної на продаж;
- додавання нових клієнтів у базу даних;
- пошук товарів на сайті за назвою;
- обробка get та post запитів;
- маршрутизація по сайту;
- отримання даних від серверу;
- підключення та робота з MySQL;
- обробка запитів MySQL.

Сервер бази даних повинен виконувати наступні пункти:

- коректно обробляти запити від web-сайту;
- обробка та повернення результатів виконаних запитів.

Створений сайт повинен знайти практичне використання та допомагати при створенні покупок в інтернеті.

## **1.5. Вимоги до програми або програмного виробу**

### **1.5.1. Вимоги до функціональних характеристик**

Для досягнення поставлених цілей програмне забезпечення, що розробляється, повинно підтримувати виконання наступних дій:

Програма повинна реалізувати наступні функції:

- формування web-сторінок на основі шаблонів;
- завантаження і видалення інформації про товари та акції;
- формування замовлень;
- забезпечення зворотнього зв'язку між клієнтами та представниками компанії.

Також система повинна мати зручний та інтуїтивно зрозумілий інтерфейс.

### **1.5.2. Вимоги до інформаційної безпеки**

Для уникнення некоректної роботи програми необхідно реалізувати:

- семантичний та синтаксичний контроль вхідних даних;
- обробку виняткових ситуацій;
- виведення повідомлень про помилки;
- можливість повторного введення даних;
- можливість безперервної роботи протягом не менше 120 годин (5 діб);
- платформну незалежність.
- час відновлення після збою - 5 хв;
- час відновлення після відмови одного з елементів підсистеми не повинно перевищувати половини операційного банківського дня;
- вірогідність виникнення не більше 2 логічних помилок на 1000 операторів за 1 рік експлуатації;



– забезпечення неушкодженого стану даних, що зберігаються в базі даних, у випадку відмови підсистеми.

### **1.5.3. Вимоги до складу та параметрів технічних засобів**

Для нормального функціонування програми необхідно, щоб обчислювальна машина, на якій буде функціонувати веб-орієнтована підсистема, відповідала наступним вимогам:

- процесор класу Intel Xeon з тактовою частотою не менш 2.4 ГГц;
- не менше 2 GB оперативної пам'яті;
- рідкокристалічний монітор з діагоналлю не менше 17";
- 20 Гб вільного місця на жорсткому диску;
- доступ до мережі Internet;
- клавіатура;
- маніпулятор "миша".

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

Також слід зазначити що більша частина подібних розробок розміщено на серверах сторонніх організацій, таким чином саме вимоги до до складу та параметрів технічних засобів не є актуальними.

З точки зору користувачів програма не є вимогливою до складу та параметрів технічних засобів та може завантажуватись на ПК, ноутбуках, планшетах чи смартфонах різних типів та конфігурацій під управлінням різних ОС. Для завантаження сторінки необхідно лише веб-браузер.

#### **1.5.4. Вимоги до інформаційної та програмної сумісності**

Програма не є вимогливою до інформаційної та програмної сумісності та може Програма не є вимогливою до складу та параметрів технічних засобів та може завантажуватись на ПК, ноутбуках, планшетах чи смартфонах різних типів та конфігурацій під управлінням різних ОС. Для завантаження сторінки необхідно лише веб-браузер.

## РОЗДІЛ 2

### ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

#### 2.1. Функціональне призначення системи

Основним призначенням системи, що розробляється в кваліфікаційній роботі є організація інтернет торгівлі електротоварами за рахунок розробки та використання розробленого у роботі веб-додатку (інтернет-магазину).

Основним призначенням системи, що розробляється є підвищення ефективності продажів електротоварів, та просування на ринку продукції компанії й підвищення об'ємів її продажів.

Основні функції веб-додатку:

- інформування користувачів про товари та послуги що надає компанія;
- інформування користувачів про акції;
- надання можливості клієнту зробити замовлення та обробка цих замовлень;
- забезпечення взаємозв'язку між клієнтами та співробітниками компанії;
- проведення рекламних та маркетингових компаній.

#### 2.2. Опис застосованих математичних методів

Оскільки особливості предметної галузі розв'язуваної задачі не передбачають застосування математичних методів, при розробці даного додатку математичні методи не використовувалися.

Математичні методи та функції, що використовуються системою для адаптації сторінки до різних типів пристроїв, є стандартними та знаходяться у відповідних бібліотеках.

## 2.3. Опис використаних технологій та мов програмування

Мова HTML. Всі відомі нам сайти написані на HTML, але багато людей вважають, що HTML це мова програмування, нажаль вони всі помиляються.

HTML - скорочення від "HyperText Mark-up Language" - перекладається як "Мова розмітка гіпертексту" (Гіпертекст - це текст, що не послідовно зв'язаний з іншими документами, тобто у вас є змога з першої сторінки документу перейти на останню). Іншими словами HTML - це мова розмітки, або ще один спосіб зберігання інформації. За допомогою HTML позначається текст, вказуючи своєму веб-переглядачу, як він має розуміти позначений текст, так само як і на жорсткому диску інформація зберігається в блоках, кластерах, секторах, доріжках і тільки за допомогою, такої, визначеної структури твій комп'ютер розуміє, що треба, а що не треба зчитувати.

У HTML текст позначається за допомогою тегів. Кожен HTML документ буде складатися з деякої групи елементів, де кожен елемент буде визначатися (починатися та закінчуватися) певним тегом (Для деяких елементів кінцевий тег не є обов'язковим). Тег — це назва елемента, записана у кутових дужках (<>).

Версії мови гіпертекстових посилань оновлюються – останньою є п'ята версія. HTML5 підтримує роздільні інструкції форматування, що звуться каскадними таблицями стилів і інші засоби оформлення. Перемістивши всі налаштування форматування в таблиці стилів, легко оформити в єдиному стилі не лише окремі абзаци, а і кожну сторінку сайту. Таблиці стилів відділяють деталі дизайну сторінки (опис зовнішнього вигляду) від її структури і змісту. Це значно облегшує написання коду і допомагає сконцентруватися на інформації, а не оформленні.

Основні теги мови HTML, представлені у таблиці 2.1.

### Основні теги HTML

Тег	Опис
<html>	Означає початок документу HTML
<head>	Ідентифікує область заголовку сторінки
<title>	Задає назву сторінки
<style>	Задає стиль
<body>	Ідентифікує область основного змісту веб-сторінки
<header>	Задає «шапку» сайту, в якій зазвичай розташовується заголовок
<main>	Призначений для основного вмісту документа
<p>	Означає абзац
<strong>	Виділяє текст жирним шрифтом
 	Перехід на нову строку
<a>	Задає «якір» чи посилання на зовнішню сторінку
<img>	Додає зображення

Кожному елементу, також, можна назначити свої унікальні атрибути, багатьом з яких треба додати значення. Це може бути число або деякий вираз, наприклад вирівнювання абзацу на сторінці задається з допомогою атрибуту align, який може приймати значення вирівнювання по лівому краю (left), правому (right) або по центру (center). Значення атрибутів завжди повинні бути замкнуті в лапки і міститися лише в відкриваючому тегі.

Іноді на сторінці необхідно ввести якийсь спеціальний символ, наприклад знак авторського права. Такий символ відсутній на клавіатурі користувача. Але рішення є – спеціальні символи вводяться за допомогою комбінації звичайних символів, що починаються з & і закінчуються крапкою з комою.

Щоб уникнути помилок при написанні коду рекомендується постійно перечитувати написане. Треба постійно переконуватися, що всі теги замкнуті в кутові дужки, а всі закриваючі теги містять косу риску. Всі значення атрибутів

повинні бути замкнуті в лапки. Порядок, в якому проставлені закриваючі теги, повинен бути зворотнім порядку тегів, що відкриваються.

Щоб документ було зручно читати, довгі рядки рекомендується розбивати на більш короткі. Іноді зручно робити відступи пробілами, розмір файла при цьому майже не зміниться, а код стане більш зрозумілим.

Коли редагування сторінки закінчене можна перейти до її збереження. Оскільки нам треба зберегти не просто сторінку з кодом, а саме HTML сторінку, то після введення її назви необхідно задати розширення - .html.

CSS (Cascading Style Sheets - каскадні таблиці стилів) - одна з базових технологій у сучасному Інтернеті. Нечасто можна зустріти сайт, зверстаний без використання CSS. CSS-код - це список інструкцій для браузера, - як і де відображати елементи веб-сторінки, написаний особливим чином. Під «елементами» зазвичай маються на увазі теги HTML і їх вміст[8].

Інструкції CSS зручно зберігати у вигляді окремого текстового файлу з розширенням «.css» , або у вигляді окремого текстового фрагмента на початку HTML документа. Основна ідея CSS в тому, щоб відокремити дизайн документа від його вмісту. CSS відповідає за оформлення і зовнішній вигляд, а HTML - за зміст і логічну структуру документа.

Завдяки CSS творці веб-сторінок легко можуть задавати для сторінки кольори, шрифти, розташування та багато інших аспектів візуального представлення HTML-документу, написаного на CSS. Таке розділення підвищує доступність документу, його гнучкість та легкість редагування візуальних елементів сторінки, а також спростити документ та зменшити повторюваність в його структурі. Завдяки CSS можна представити одну і ту ж сторінку в багатьох стилях або методах висновку, таких як вивід на екран, друк, голосове відтворення завдяки спеціальним програмам, або передати код для розпізнавання програмам, що використовують шрифт Брайля.

Стандарт CSS визначає пріоритети, у порядку яких застосовуються правила стилів, якщо для якогось елемента підходять деякі правила одночасно.

Це називається "каскадом", в якому для правил розраховуються пріоритети або "ваги", що робить результати передбаченими.

Таблиця стилів складається з набору правил. Кожне правило, у свою чергу, складається з одного або декількох селекторів, розділених комами і блоку визначень.

До появи CSS оформлення веб-сторінок здійснювалося безпосередньо усередині вмісту документа. Проте з появою CSS стало можливим принципове розділення змісту і представлення документа. За рахунок цього нововведення стало можливим легке застосування єдиного стилю оформлення для маси схожих документів, а також швидка зміна цього оформлення.

Переваги CSS розмітки:

- декілька дизайнів сторінки для різних пристроїв перегляду. Наприклад, на екрані дизайн буде розрахований на велику ширину, під час друку меню не виводитиметься, а на КПК і стільниковому телефоні меню буде слід за вмістом;
- зменшення часу завантаження сторінок сайту за рахунок перенесення правил представлення даних в окремий CSS-файл. В цьому випадку браузер завантажує тільки структуру документа і дані, що зберігаються на сторінці, а представлення цих даних завантажується браузером тільки один раз і кеширується;
- простота подальшої зміни дизайну. Не потрібно правити кожен сторінку, а лише змінити CSS-файл;
- додаткові можливості оформлення.

JavaScript (JS) – це мова програмування, що дозволяє зробити Web -сторінку інтерактивною, тобто такою що реагує на дії користувача[3].

Послідовність інструкцій (що називається програмою, скриптом або сценарієм) виконується інтерпретатором, вбудованим в звичайний Web -браузер. Іншими словами, код програми вбудовується в HTML - документ і виконується на боці клієнта. Для виконання програми не потрібно навіть перезавантажувати Web -сторінку, всі програми виконуються в відповідь на будь-яку подію. Наприклад, перед відправленням даних форми можна перевірити їх на допустимі

значення і, якщо значення не відповідають очікуваним, заборонити відправлення даних.

JavaScript - об'єктно-орієнтована скриптова мова програмування і є діалектом мови ECMAScript.

JavaScript зазвичай використовується як вбудована мова для програмного доступу до об'єктів додатків. Найбільш широке застосування знаходить у браузерях як мова сценаріїв для надання інтерактивності веб-сторінкам.

Основні архітектурні риси:

- динамічна типізація;
- автоматичне керування пам'яттю;
- прототипне програмування;
- функції як об'єкти першого класу.

На JavaScript вплинули багато мов, при розробці була мета зробити мову схожою на Java, але при цьому легкою для використання непрограмістами. JavaScript має низку властивостей об'єктно-орієнтованої мови, але реалізоване в мові прототипування обумовлює відмінності в роботі з об'єктами в порівнянні з традиційними об'єктно-орієнтованими мовами. Крім того, JavaScript має ряд властивостей, властивих функціональним мовам, - функції як об'єкти першого класу, об'єкти як списки, каррінг, анонімні функції, замикання - що додає мові додаткову гнучкість.

## **2.4. Опис структури системи та алгоритмів її функціонування**

Bootstrap. Це безкоштовний набір інструментів з відкритим кодом, призначений для створення сайтів та веб-застосунків, який містить шаблони CSS та HTML для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення JavaScript. Він спрощує розробку динамічних сайтів і веб-застосунків.

Bootstrap – це клієнтський фреймворк, тобто інтерфейс для користувача, на відміну від коду серверної сторони, який знаходиться на сервері.



Bootstrap сумісний з останніми версіями браузерів Google Chrome, Firefox, Internet Explorer, Opera і Safari (деякі з цих браузерів підтримуються не на всіх платформах).

Основа Bootstrap — модульна сітка з 12 колонок, яка вміє перелаштовуватися в залежності від розміру екрану пристрою відвідувача вашого сайту.

Крім того, Bootstrap включає в себе стилі оформлення безлічі елементів сучасних сайтів: кнопок, форм, спливаючих вікон, навігаційних панелей і т.п.

Bootstrap — не єдиний спосіб створення адаптивної верстки. Є альтернативи — такі, як Foundation. Також можна зробити адаптивність повністю вручну, використовуючи Media Queries. Це значно довше, але цілком можливо.

Важлива перевага Bootstrap — в його масовості. Він використовується так давно і на такій великій кількості сайтів, що можна бути впевненим, що більшість проблем сумісності з браузерами та операційними системами в ньому вирішені.

Node.js. Платформа з відкритим кодом для виконання високопродуктивних мережевих застосунків, написаних мовою JavaScript[6].

Якщо раніше Javascript застосовувався для обробки даних в браузері користувача, то node.js надав можливість виконувати JavaScript-скрипти на сервері та відправляти користувачеві результат їх виконання.

Платформа Node.js перетворила JavaScript на мову загального використання з великою спільнотою розробників.

Node.js має наступні властивості:

- асинхронна одно-ниткова\_модель виконання запитів;
- неблокуючий ввід/вивід;
- система модулів CommonJS;
- рушій JavaScript Google V8.

Для керування модулями використовується пакетний менеджер npm (node package manager).

NPM модулі які використовуються в проекті:

- "body-parser": 1.19.0;
- "ejs": v.3.1.2;
- "express": v.4.17.1;
- "hbs": v.4.1.0;
- "mysql2": v.2.1.0;
- "nodemon": v.2.0.3.

Express. Це мінімалістичний та гнучкий фреймворк для веб-застосунків, побудованих на Node.js, що надає широкий набір функціональності.

Маючи в своєму розпорядженні безліч допоміжних HTTP-методів та проміжних обробників, створювати надійні API можна легко і швидко.

Express забезпечує тонкий прошарок базової функціональності для веб-застосунків, що не спотворює звичну та зручну функціональність Node.js[7].

Хост. Це термін, що використовується в комп'ютерних мережах та програмуванні і має декілька визначень.

Будь-який комп'ютерний пристрій, що має доступ до IP мережі тобто синонім терміна вузол мережі. У сучасних мережах цей термін поширюється не тільки на традиційні комп'ютери, а також і на смартфони, планшети, телевізори з доступом до Інтернету та інші подібні пристрої.

Комп'ютерний сервер, тобто пристрій, що містить ресурс і надає до нього доступ у форматі клієнт-сервер.

Веб-сервер, пристрій який надає послуги веб-хостингу.

Комп'ютерна програма, що надає послуги іншим програмам та застосункам, наприклад забезпечує їх певними даними.

Вживання слова «хост» має сенс тільки з поясненням, хостом якого сервісу є цей пристрій. А втім, часто назва сервісу опускається і припускається, що це очевидно з контексту.

Для прикладу, в реалізації протоколу USB керуванням і координацією кожної групи клієнтських пристроїв займається окремий USB Host Controller. У такому разі він виступає як хост протоколу USB. З другого боку, при виконанні,

наприклад, команди ping певної IP-адреси хостом можна назвати комп'ютер, який володіє цією адресою і надає сервіс відповіді на ICMP-пакети. У такому разі такий комп'ютер виступає як хост сервісів ICMP.

Хостом спрощеною мовою називається просто унікальний відвідувач сайту. Скільки б сторінок відвідувач не проглянув на сайті, скільки б часу він на ньому не залишався, все це буде вважатися одним хостом. Зазвичай, статистика обмежує «час унікальності» добою. Якщо на сайт з однієї IP-адреси зайшов відвідувач, тобто хост, його черговий візит через добу і більше знову буде вважатися унікальним.

Домен – це ім'я сайту. Ви вводите його в адресному рядку веб-переглядача, коли хочете перейти на сайт.

Доменне ім'я – це еквівалент вашого фактичної адреси, вашого місця розташування. Фактично, це адреса вашого сайту, місце його розміщення в інтернеті, щоб ваші користувачі потрапляли саме туди, куди хочуть.

Доменне ім'я складається з двох основних елементів. Наприклад, доменне ім'я Facebook.com складається з імені сайту (Facebook) і розширення доменного імені (.com). Коли компанія чи окрема особа оплачують домен, вони можуть вказати, на який сервер спрямований домен.

Доменні імена вигадали, щоб було зручніше знаходити сайти в інтернеті. Раніше потрібно було запам'ятати цифрову адресу сайту, щоб відкрити його у веб-переглядачі. Цифрова чи IP-адреса — це комбінація з чотирьох чисел, розділених крапками. IP-адреса вказує на сервер, де розташований сайт, і має такий вигляд: 7.135.10.13.

Доменні імена працюють як ярлики для сервера, на якому розміщений сайт. Без імені домена будь-хто, хто захоче відвідати ваш сайт, повинен буде ввести повний IP-адреса. Але проблема в тому, що IP-адреса не так просто запам'ятати і він буде дивно виглядати в рекламних матеріалах.

Середовище обробки заказів. Для обробки заказів використовується база MySQL. Після оформлення заказів на сайті, інформація записується в базу

даних. Менеджер компанії може побачити оформлений заказ, скорегувати його, зв'язатись с замовником та остаточно з'ясувати склад замовлення.

Приклад використання новоствореного замовлення можна побачити на рис. 2.1 - 2.2.

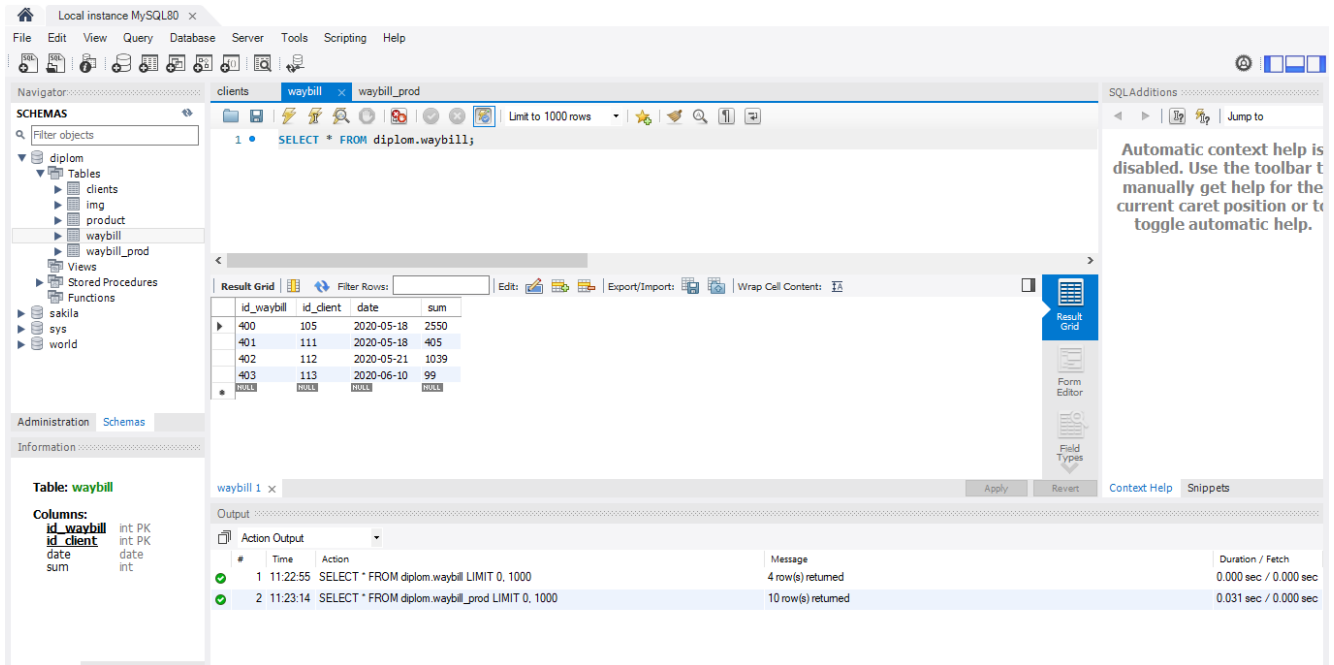


Рис. 2.1. Накладна на продаж з ID клієнта

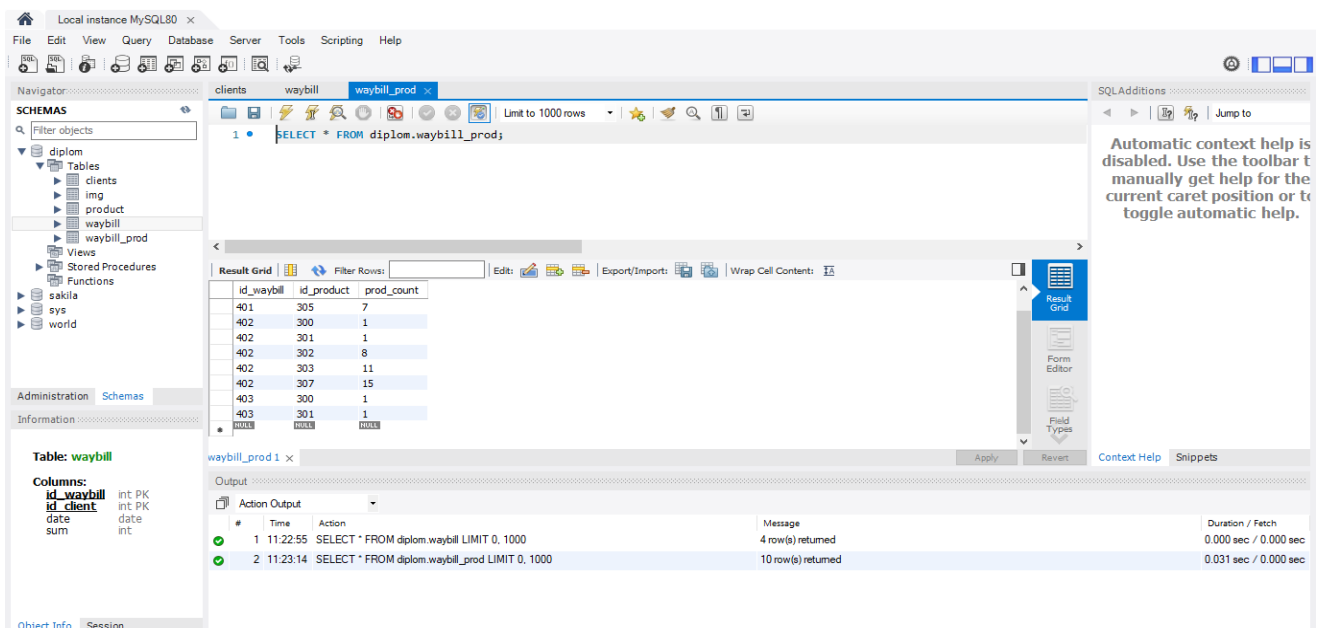


Рис. 2.2. Накладна на продаж з ID товара

Структурна схема інтернет-магазину «LightOn» подана на рис. 2.3.

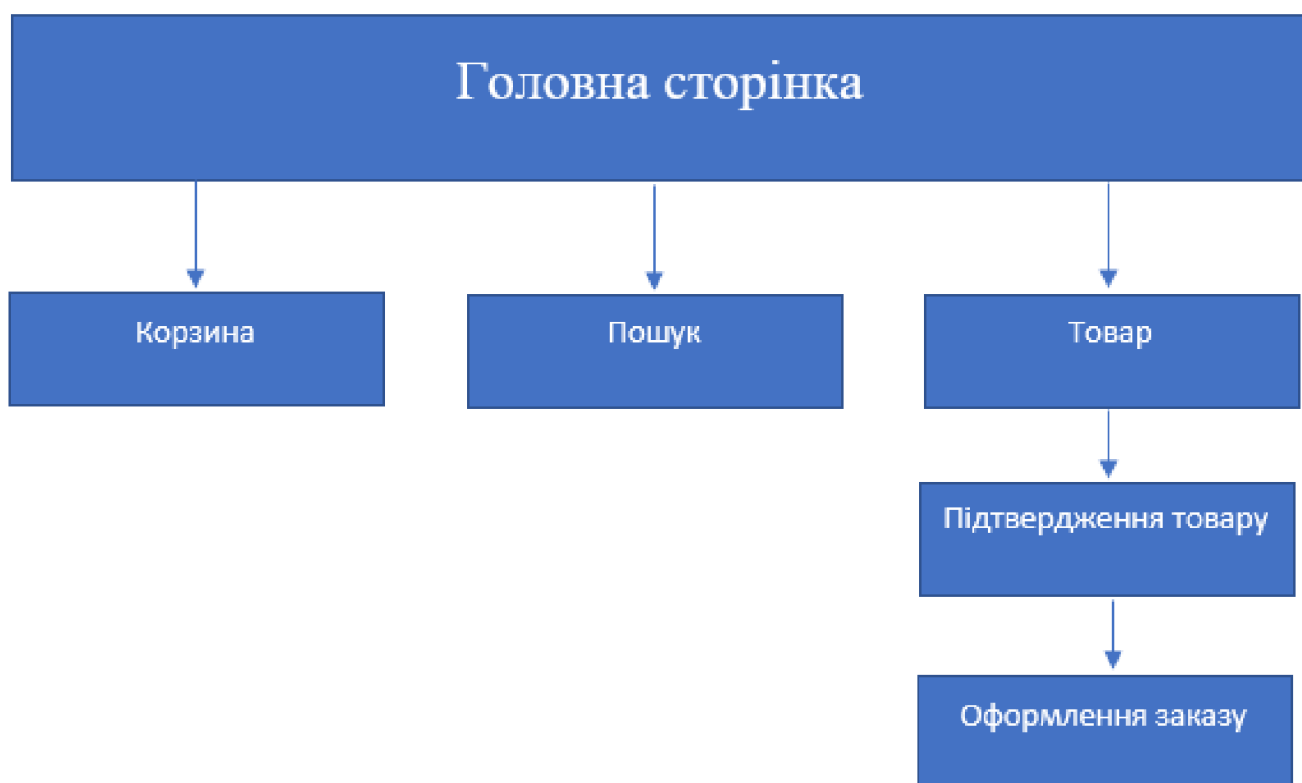


Рис. 2.3. Структурна схема інтернет-магазину

MySQL - це система керування базами даних з відкритим вихідним кодом (СУБД) з моделлю клієнт-сервер[4]. СУБД - це програмне забезпечення або служба, яка використовується для створення та управління базами даних на основі реляційної моделі.

Бази даних мають наступні види:

1. Нереляційних база даних - це база даних, в якій на відміну від більшості традиційних систем баз даних не використовується таблична схема рядків і стовпців. У цих базах даних застосовується модель зберігання, оптимізована під конкретні вимоги типу збережених даних. Наприклад, дані можуть зберігатися як прості пари "ключ - значення", документи JSON або граф, що складається з ребер і вершин.

Переваги нереляційної моделі:

- багато нереляційної рішення мають обмежену функціональність, оскільки вирішують певні завдання. Тому для роботи з такими базами даних не

потрібно глибоких знань SQL-запитів. Це сильно знижує вхідний поріг для початку роботи з нереляційної сховищами;

- використання хмарних обчислень і сховищ. Використання для тестування і розробки локального устаткування, а потім перенесення системи в хмару;

- швидка розробка. NoSQL бази даних не вимагають великого обсягу підготовчих дій, який потрібен для реляційних баз.

Недоліки нереляційної моделі:

- додаток сильно прив'язується до конкретної СКБД. Мова SQL універсальний для всіх реляційних сховищ і тому в разі зміни СКБД не доведеться переписувати весь код;
- труднощі швидкого переходу з однієї нереляційних бази даних на іншу;
- доводиться розробляти власні інструменти для роботи з БД;
- обмежена ємність вбудованої мови запитів.

2. Реляційні бази даних - найстаріший тип досі широко використовуваних БД загального призначення. Дані та зв'язку між даними організовані за допомогою таблиць. Кожен стовпець у таблиці має ім'я і тип. Кожен рядок представляє окремий запис або елемент даних в таблиці, що містить значення для кожного з стовпців [5].

Переваги реляційної моделі:

- простота і доступність для розуміння користувачем. Єдиною використовуваною інформаційною конструкцією є «таблиця»;
- строгі правила проектування, які базуються на математичному апараті;
- повна незалежність даних. Зміни в прикладній програмі при зміні реляційної БД мінімальні;
- для організації запитів і написання прикладного ПЗ немає необхідності знати конкретну організацію БД у зовнішній пам'яті.

Недоліки реляційної моделі:

- далеко не завжди предметна область може бути представлена у вигляді «таблиць»;

- в результаті логічного проєтування з'являється безліч «таблиць». Це призводить до труднощів розуміння структури даних;
- БД займає відносно багато зовнішньої пам'яті;
- відносно низька швидкість доступу до даних.

Підсумовуючи все вище сказане можна зробити висновок, що для проєкту найбільше підходить реляційна модель бази даних тому, що вона може вирішити описані вище проблеми та аномалії.

## **2.5. Обґрунтування та організація вхідних та вихідних даних програми**

В якості вхідних даних система використовує інформацію запиту від користувача про замовлення. Вхідними даними також можна вважати інформацію про розмір вікна браузера для адаптації відображення сторінки на екрані пристрою, оскільки веб-додаток є адаптивним.

Також в якості вхідних та вихідних даних використовуються заповнена форма зв'язку користувача зі співробітниками компанії.

## **2.6. Опис роботи розробленої системи**

### **2.6.1. Використані технічні засоби**

Програма не є вимогливою до складу та параметрів технічних засобів та може завантажуватись на ПК, ноутбуках, планшетах чи смартфонах різних типів та конфігурацій під управлінням різних ОС.

Для завантаження сторінки необхідно лише веб-браузер.

Сайт розроблено на ноутбучі Ноутбук Dell Inspiron G5 5500:

Екран 15.6" WVA (1920x1080) Full HD, глясовий з антивідблисковим покриттям / Intel Core i5-10300H (2.5 — 4.5 ГГц) / RAM 8 ГБ / SSD 1 ТБ / nVidia GeForce GTX 1650 Ti, 4 ГБ / без ОД / LAN / Wi-Fi / Bluetooth / вебкамера

Розміщення сайту організовано на зовнішніх серверах за рахунок технічних засобів провайдеру.

## 2.6.2. Використані програмні засоби

Для написання програмного забезпечення використовують прикладні програми, що сприяють більш комфортній розробці, допомагають підказками тощо.

Програмою для написання програмного забезпечення може стати будь-який текстовий редактор, починаючи від текстового блокноту Windows до Microsoft Word, але всі вони програють в порівнянні з професіональними редакторами, такими як:

- Sublime Text3;
- Visual Studio Code;
- Atom;
- WebStorm;

Для розробки web-сайту був обрано текстовий редактор Sublime Text3.

Sublime Text - багатофункціональний текстовий редактор з широким набором зручних інструментів для виділення, маркування та обробки текстових фрагментів коду[1].

Інтерфейс цього редактора дуже лаконічний. Зате швидкість роботи і відгуку на всі Ваші дії на досить високому рівні. Підтримує величезну кількість мов (C ++, Dylan, Erlang, HTML, Haskell, Java, JavaScript, Lua, Markdown, MATLAB, Perl, PHP, Python, Ruby, SQL, XML та ін.) І пропонує на вибір близько 20 колірних схем. Вельми зручно, що реалізований повноекранний режим - дуже корисно, якщо не хочете, щоб Вас щось відволікало від повноцінної роботи. Крім того, для Вас приготовлені так звані Мультипанель: можна паралельно працювати з декількома файлами в одному вікні, що набагато зручніше, ніж використовувати кілька окремих вікон. Вигляд робочого вікна програми можна побачити на рисунку 2.4.



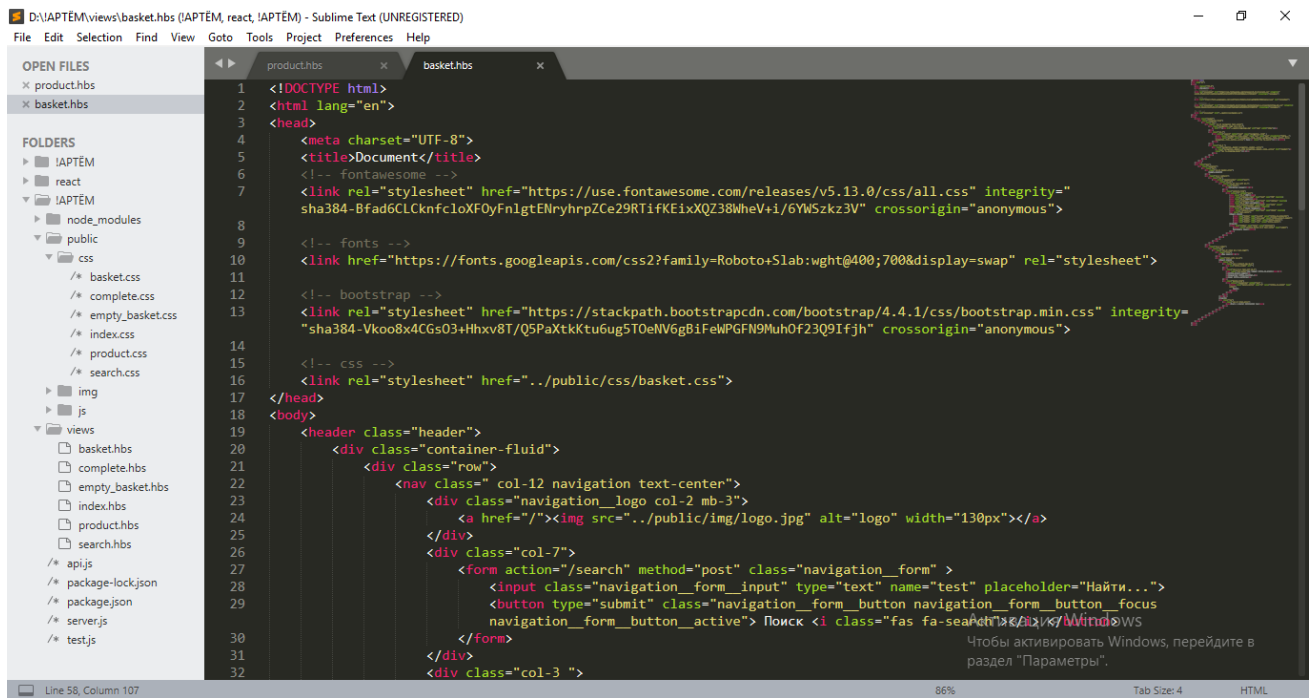


Рис. 2.4. Вигляд робочого вікна програми Sublime Text3

Для комфортної роботи за базою даних найбільш підходить MySQL Workbench 8.0[2].

MySQL Workbench - це уніфікований візуальний інструмент для архітекторів баз даних і розробників БД. MySQL Workbench надає можливість моделювання даних, розробку SQL і комплексні інструменти адміністрування для конфігурації сервера, адміністрування користувачів, резервного копіювання та багато іншого. MySQL Workbench доступний на Windows, Linux і Mac OS X. Вигляд робочого вікна програми можна побачити на рис. 2.5.

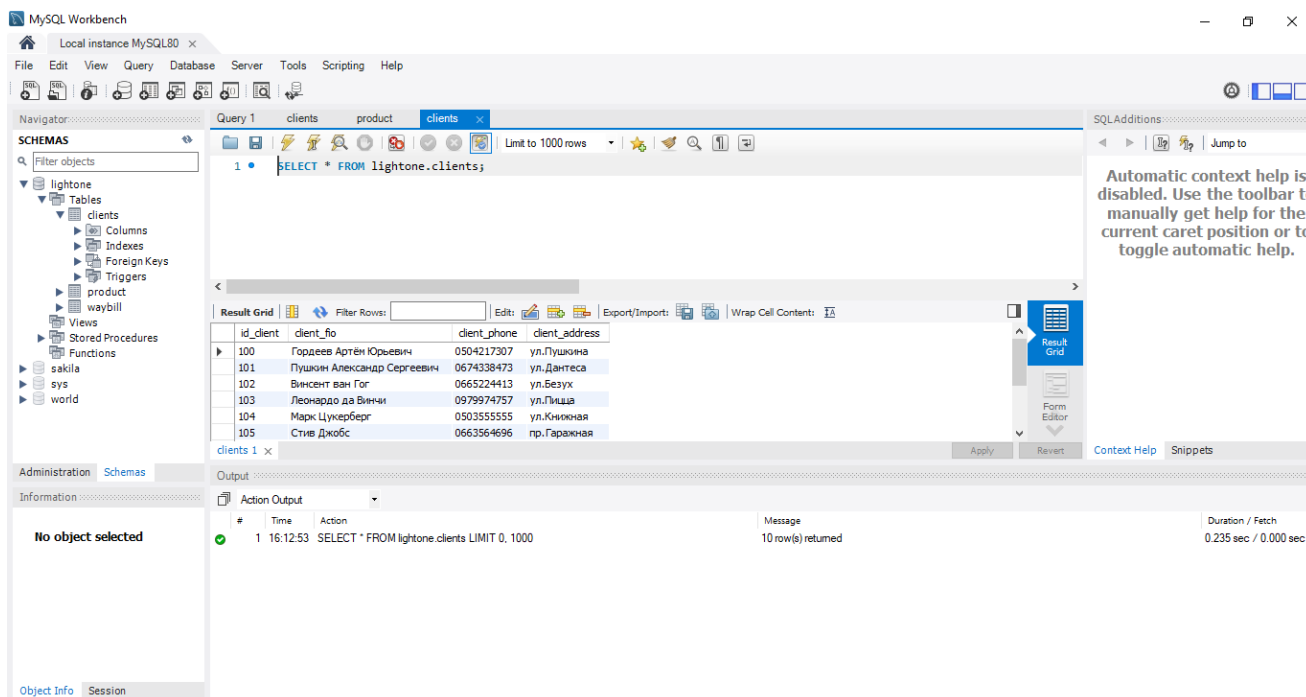


Рис. 2.5. Вигляд робочого вікна програми MySQL Workbench

Окрім технічної частини сайту так дуже важливою є його графічна частина. Для цього був обран Adobe Photoshop CC (рис. 2.6).

Photoshop – це багатофункціональний графічний редактор, розробкою якого (як і попередньої програми - Dreamweaver) займається американська корпорація Adobe. Переважно працює з растровим типом зображень, але має і деякі векторні функції й інструменти. Ця програма є світовим лідером в області графічного редагування зображень, працювати в якій можна не лише на операційній системі Windows, а й на інших популярних операційних системах. Вигляд робочого вікна програми можна побачити на рис. 2.6.

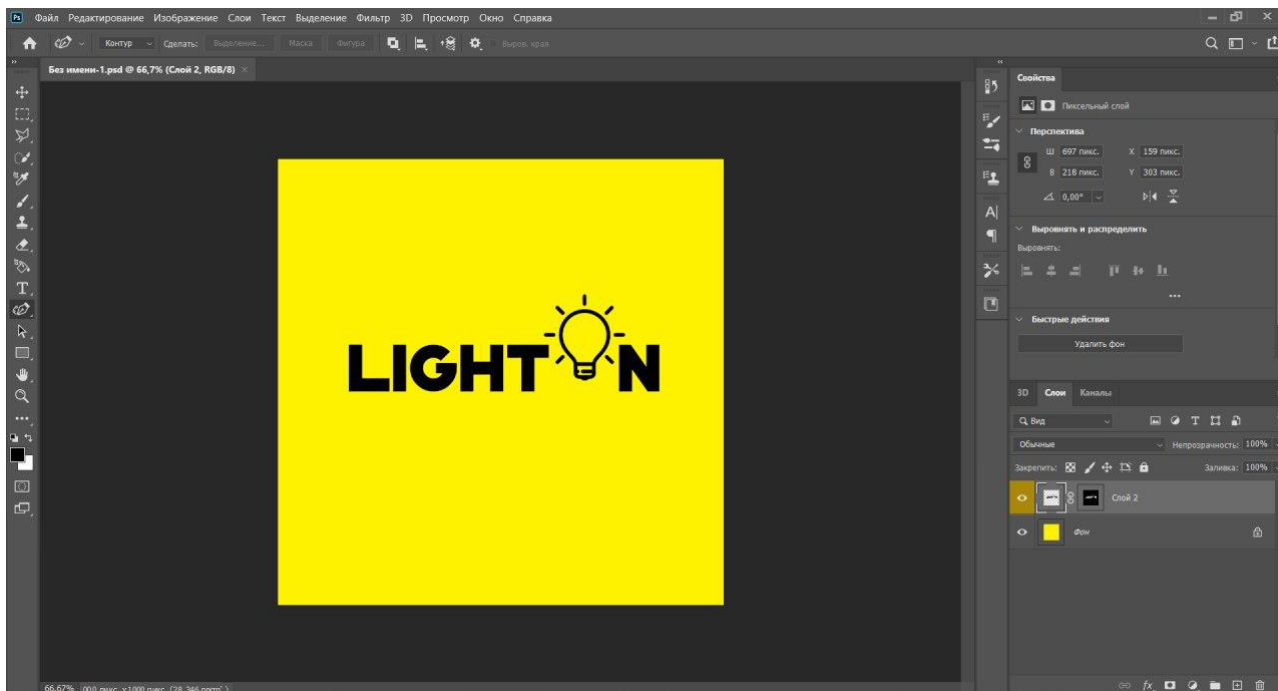


Рис. 2.6. Вигляд робочого вікна програми Adobe Photoshop

Як видно, у програмі три основних панелі з різноманітними функціями – панель інструментів ліворуч, панель шарів та стилів праворуч та панель додаткових можливостей зверху.

Ліва панель інструментів поміж багатьох своїх принад має такі основні: «Ласо», «Чарівна паличка», кадрування інструментом «Рамка», «Пензель» та «Олівець», «Штамп», «Ластик», «Розмиття» та «Різкість», «Освітлення» та «Затемнення», «Текст» та багато інших. Тут же обираються кольори, з якими в даний момент йде робота. Так, мінімальний набір інструментів, необхідний для створення кнопки у веб-документі повинен складатися з «Пензля», «Рамки» і, звичайно ж, «Тексту».

Праворуч розташовані одразу декілька панелей. Це «Навігатор», який допомагає визначити в якій області зображення ми працюємо при збільшуванні масштабу, панель «Історія», яка показує весь хід зміни документа, що дуже корисно, якщо є потреба відмінити попередні дії, або порівняти їх результат, панель «Операції», «Пензлі», що містить всі наявні у бібліотеках на вашому комп'ютері набори пензлів, які, до речі, можна додатково завантажувати в Інтернеті, «Джерело клонів» для інструменту «Штамп», «Символ» для

управління шрифтом, розміром, інтерліньяжем, кернінгом, трекінгом, кольором, жирністю та нахилом тексту.

Ще нижче розташовані останні три панелі, розташовані праворуч: «Шари», де можна керувати шарами зображень, з яких складається наш графічний документ, «Канали», де можна працювати окремо з кожним з чотирьох каналів (всі кольори, червоний, жовтий та синій) і «Контури», відповідно, для роботи з контурами.

Останніми розглянемо елементи верхнього меню. У вкладці «Редагування» найбільш значимі елементи – «Крок назад» і «Крок вперед», «Скопіювати», «Вирізати», «Вставити», «Вільне трансформування». У наступній вкладці («Зображення») виділимо вкладки «Корекція» в якій знаходяться інструменти кольорової, світової, контрастної корекції і ще декілька корисних елементів, «Вертання холсту», і «Тримінг». Далі йде вкладка «Шар», де містяться всі інструменти для роботи з шарами документу – «Новий шар», «Дублікат шару», «Видалити шар», «Параметри шару», «3D шари», «Згрупувати шари» та «Об'єднати шари», наступна вкладка верхньої панелі – «Виділення», а за нею слідом йде вкладка «Фільтр». Вона є однією з найважливіших вкладок, бо пропонує дуже великий вибір різноманітних операцій-фільтрів над зображеннями.

Тут представлено важливі групи фільтрів як: «Імітація», «Оформлення», «Розмиття», «Рендеринг», «Стилізація», «Шум» та багато інших, які дозволяють отримувати дуже цікаві ефекти. Більшість з наведених фільтрів потребує вказування значень сили ефектів, таким чином їх можна застосовувати від ледь помітного результату до досить суттєвого перетворення.

Таким чином бачимо, що програма Adobe Photoshop має величезний арсенал допомоги тим, хто бажає працювати з графікою в цілому і з веб-дизайном конкретно.

### 2.6.3. Виклик та завантаження програми

Спосіб виклику програми з відповідного носія даних та умови його завантаження є стандартними для запуску веб-додатків та може завантажуватись на ПК, ноутбуках, планшетах чи смартфонах різних типів та конфігурацій під управлінням різних ОС.

Веб-сайт не потребує інсталяції. Необхідно лише знати доменне ім'я та зробити до нього запит за допомогою веб-браузера.

Для завантаження сторінки необхідно лише веб-браузер.

### 2.6.4. Опис інтерфейсу користувача

На приведеному рис. 2.7. відображено головну сторінку сайту, з якої користувачі можуть перейти до будь-якого цікавого їм товару та переглянути інформацію про нього.

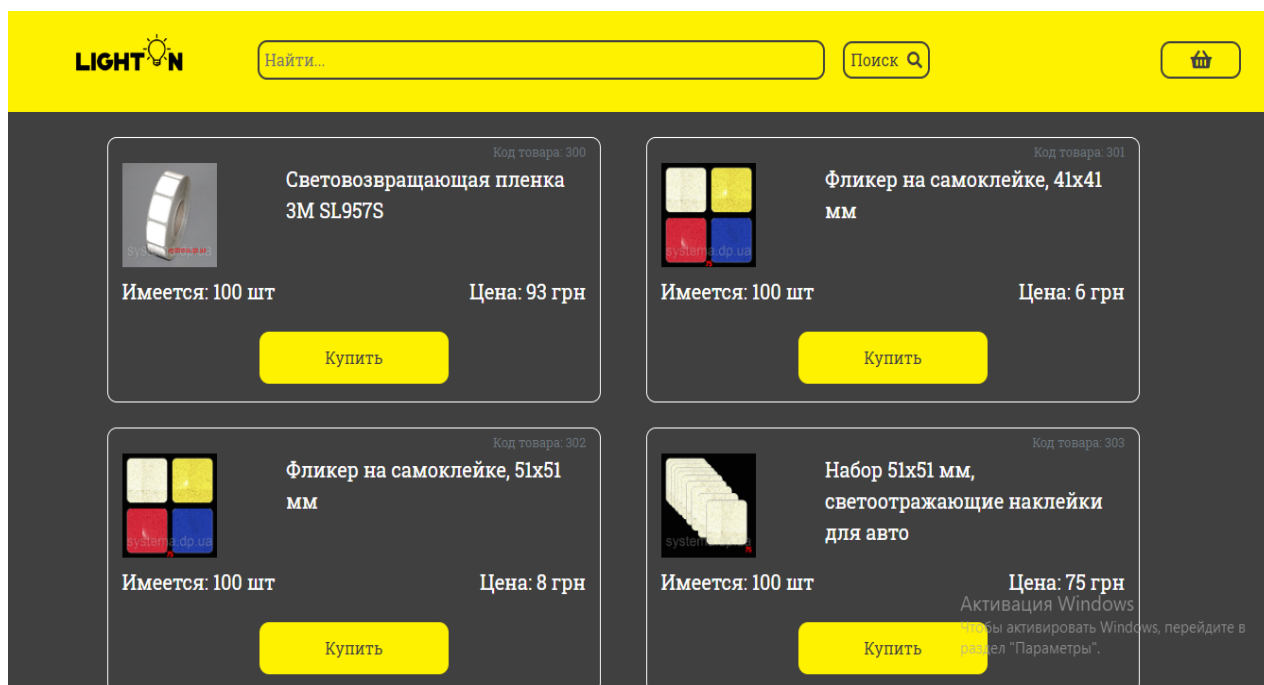


Рис. 2.7. Головна сторінка сайту

Навігація по сайту працює за допомогою HTML тегу <a>, який дозволяє пересуватися по сайту.

В шапці сайту розташовані наступні пункти меню (рисунок 2.8):

- лого;
- пошук;
- кошик.



Рис. 2.8. Шапка сайту

Якщо користувачу необхідно повернутися на головну сторінку сайту він може натиснути на лого, яке перемістить його на головну сторінку (рисунок 2.9).



Рис. 2.9. Лого сайту

Справа від лого розташована строка пошуку товарів (рисунок 2.10)



Рис. 2.10. Строка для пошуку товарів

Справа від строки пошуку знаходиться кнопка для переходу до кошику (рис. 2.11).

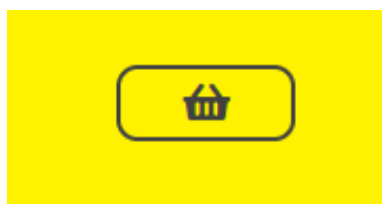


Рис. 2.11. Кнопка перехода до кошика

Коли користувачу потрібно переглянути інформацію про необхідний йому товар, він мусить натиснути на кнопку «Купити» - тоді він перейде на сторінку обраного товару (рис. 2.12 - 2.13).

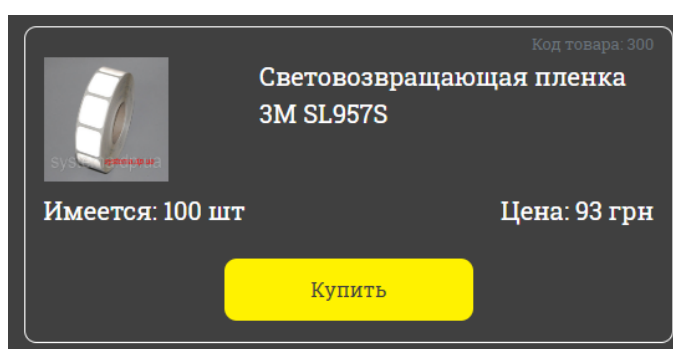


Рис. 2.11. Картка товару



Рис. 2.13. Сторінка товару

На сторінці товару користувач може побачити всю необхідну інформацію про товар:

- кількість товару на складі;
- ціну за одиницю;
- назву товару;
- код товару;
- описання товару.

Також користувач може обрати кількість товару, яку він хоче придбати (рис. 2.14).



Рис. 2.14. Лічильник кількості товарів

Щоб перейти до оформлення товарів користувачу необхідно натиснути кнопку «Купити» після чого він побачить форму для підтвердження обраного товару (рис.2.15).

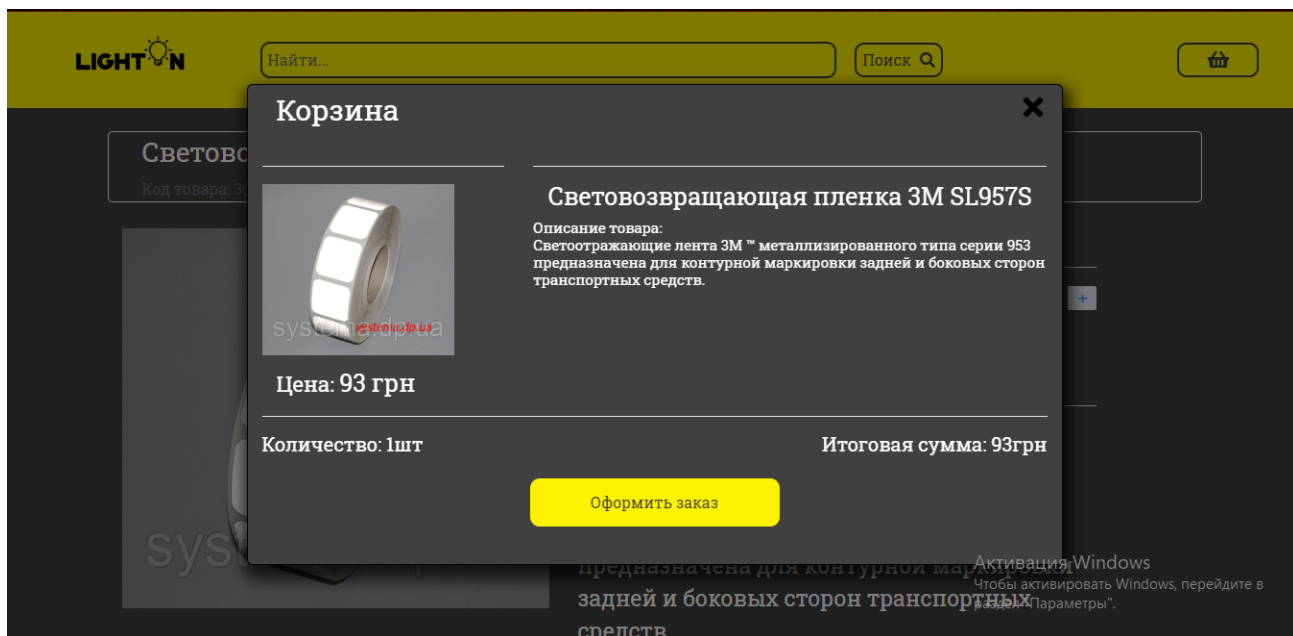


Рисунок 2.15. Вікно для підтвердження товару



Після підтвердження товару користувач потрапить на сторінку «Кошик» де будуть відображені всі товари які він додав до нього (рис. 2.16).

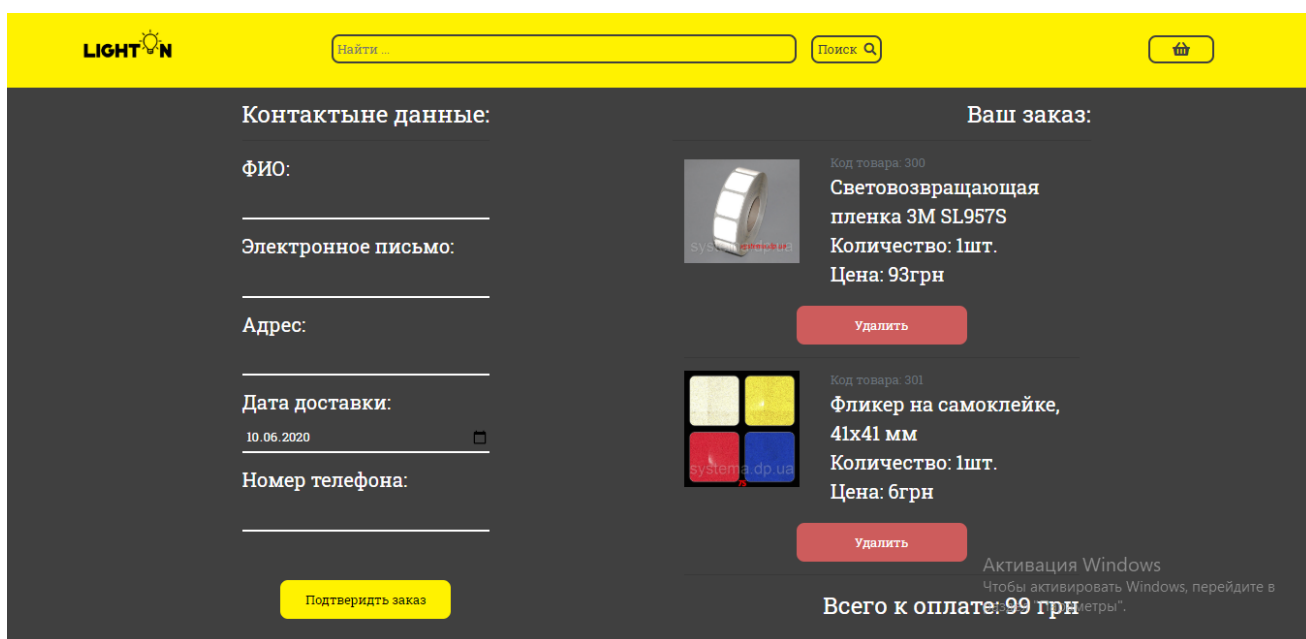
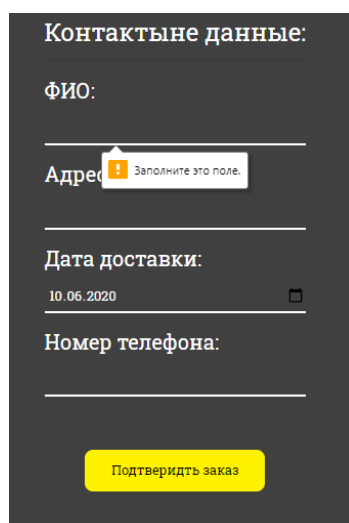


Рис. 2.16. Сторінка кошику

На цій сторінці користувач може оформити замовлення. Для цього він повинен ввести свої данні для оформлення заказу (рис. 2.17).

Рис. 2.17. Форма для заповнення особистої інформації

Якщо користувач не заповнить жодного поля, чи заповнить всі крім одного – він побачить наступне повідомлення (рис. 2.18 - 2.19).



Контактыне данные:

ФИО:

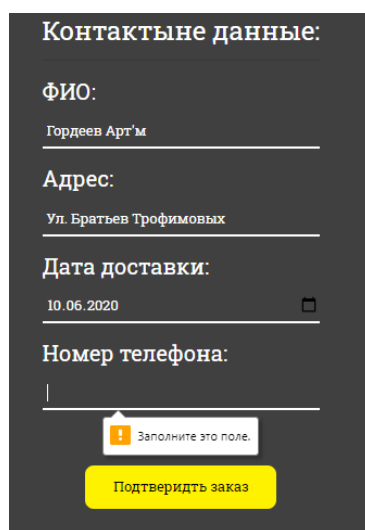
Адрес: ❗ Заполните это поле.

Дата доставки:  
10.06.2020

Номер телефона:

Подтвердить заказ

Рис. 2.18. Користувач не заповнив жодного поля



Контактыне данные:

ФИО:  
Гордеев Арт'м

Адрес:  
Ул. Братьев Трофимовых

Дата доставки:  
10.06.2020

Номер телефона:

❗ Заполните это поле.

Подтвердить заказ

Рис. 2.19. Користувач заповнив всі поля крім одного

Після оформлення замовлення, користувач потрапляє на сторінку яка повідомляє, що замовлення було прийнято в обробку (рис. 2.20).

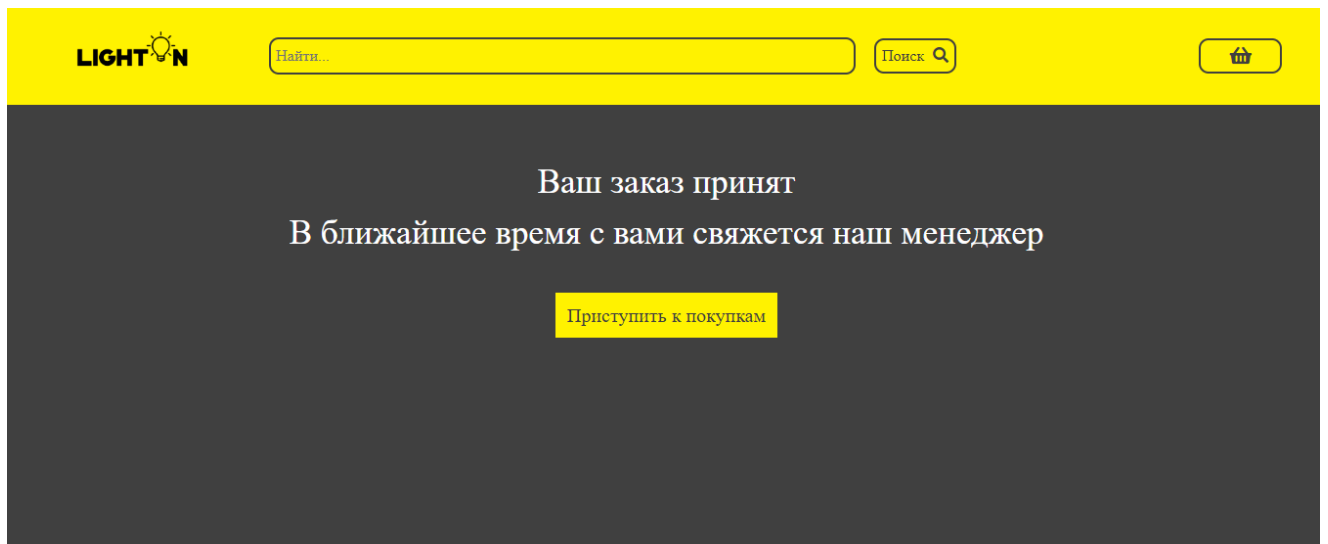


Рисунок 2.20. Повідомлення про прийнятий заказ

I, нарешті, останній фрагмент сайту - це «Підвал» сайту, де знаходяться контактні данні та посилання на соціальні мережі

## РОЗДІЛ 3 ЕКОНОМІЧНИЙ РОЗДІЛ

### 3.1. Розрахунок трудомісткості та вартості розробки програмного продукту

Початкові дані:

1. передбачуване число операторів програми – 1200;
2. коефіцієнт складності програми – 1,6;
3. коефіцієнт корекції програми в ході її розробки – 0,05;
4. годинна заробітна плата програміста – 120 грн/год;
5. коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі – 1,2;
6. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 1,2;
7. вартість машино-години ЕОМ – 13 грн/год.

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

Трудомісткість розробки ПЗ можна розрахувати за формулою:

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

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

$t_u$  - витрати праці на дослідження алгоритму рішення задачі;

$t_a$  - витрати праці на розробку блок-схеми алгоритму;

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

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

$t_{\partial}$  - витрати праці на підготовку документації.

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

Умовне число операторів (підпрограм):

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

де  $q$  - передбачуване число операторів (1200);

$C$  - коефіцієнт складності програми (1,6);

$p$  - коефіцієнт корекції програми в ході її розробки (0,05).

Звідси умовне число операторів в програмі:

$$Q = 1,6 \cdot 1200 \cdot (1 + 0,05) = 2016$$

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

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

де  $B$  - коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

$k$  - коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності. При стажі роботи від 3 до 5 років він складає 1,2.

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

$$t_u = (2016 \cdot 1,2) / (75 \cdot 1,2) = 26,88 \text{ людино-годин}$$

Витрати праці на розробку алгоритму рішення задачі визначаються за формулою:

$$t_a = \frac{Q}{(20 \dots 25) \cdot k}, \text{ людино-годин, (3.2)}$$

де  $Q$  – умовне число операторів програми;

$k$  – коефіцієнт кваліфікації програміста.

Підставивши відповідні значення в формулу (3.2), отримаємо:

$$t_a = 2016 / (20 \cdot 1,2) = 84 \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

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

$$t_n = 2016 / (25 \cdot 1,2) = 67,2 \text{ людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

- за умови автономного налагодження одного завдання:

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

$$t_{oml} = 2016 / (5 \cdot 1,2) = 336 \text{ людино-годин.}$$

- за умови комплексного налагодження завдання:

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

$$t_{oml}^k = 1,5 \cdot 336 = 504 \text{ людино-годин.}$$

Витрати праці на підготовку документації визначаються за формулою:

$$t_{\partial} = t_{\partial p} + t_{\partial o}, \text{ людино-годин,}$$

де  $t_{\partial p}$  - трудомісткість підготовки матеріалів і рукопису:

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

$t_{\partial o}$  - трудомісткість редагування, печатки й оформлення документації:

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

Підставляючи відповідні значення, отримаємо:

$$t_{\partial p} = 2016 / (18 \cdot 1,2) = 93,33 \text{ людино-годин.}$$

$$t_{\partial o} = 0,75 \cdot 93,33 = 70 \text{ людино-годин.}$$

$$t_{\partial} = 93,33 + 70 = 163,33 \text{ людино-годин.}$$

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

$$t = 50 + 26,88 + 84 + 67,2 + 336 + 163,33 = 727,41 \text{ людино-годин.}$$

### 3.2. Розрахунок витрат на створення програми

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

$$K_{ПО} = Z_{ЗП} + Z_{МВ}, \text{ грн.}$$

Заробітна плата виконавців визначається за формулою:

$$Z_{ЗП} = t \cdot C_{ПР}, \text{ грн,}$$

де:  $t$  - загальна трудомісткість, людино-годин;

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

З урахуванням того, що середня годинна зарплата програміста становить 120 грн / год, отримуємо:

$$Z_{ЗП} = 727,41 \cdot 120 = 87\,289,2 \text{ грн.}$$

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

$$Z_{МВ} = t_{отл} \cdot C_{мч}, \text{ грн, (3.3)}$$

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

$C_{мч}$  - вартість машино-години ЕОМ, грн/год (13 грн/год).

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



$$Z_{mv} = 336 \cdot 13 = 4368 \text{ грн.}$$

Звідси витрати на створення програмного продукту:

$$K_{ПО} = 87\,289,2 + 4368 = 91\,657,2 \text{ грн.}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k \cdot F_p} \text{ міс.}$$

де  $B_k$  - число виконавців (дорівнює 1);

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

Звідси витрати на створення програмного продукту:

$$T = 727,41 / 1 \cdot 176 \approx 4 \text{ міс.}$$

**Висновок.** Програмне забезпечення призначене для продажу електротоварів. Вартість даного програмного забезпечення становить 91 657,2 грн. і не вимагає додаткових витрат як при розробці програми. Очікуваний час розробки становить 4 місяці. Цей термін пов'язаний зі значним числом операторів, і включає час на дослідження і розробку алгоритму вирішення поставленого завдання, програмування по готовому алгоритму, налагодження програми і підготовку документації.

## ВИСНОВКИ

Під час створення інтернет-магазину електро товарів були засвоєні функції мови програмування CSS, мови розмітки HTML та мови JavaScript, а також функції роботи у Adobe Photoshop. Створений інтернет-магазин надає можливість користувачам переглянути, обрати та замовити електро товари. На сайті для зручного користування закріплене головне меню, створені кнопки швидкого пересування між розділами та кнопка швидкого повернення до початку сторінки. Для дизайну мною були застосовані команди мови програмування CSS.

Також слід зазначити, що інтернет-магазину є ще доволі багато напрямків для розвитку. Наприклад, можна додати ще більше товарів, розширити функціональні можливості, далі працювати з дизайном для більшої кількості клієнтів, також треба більше розвинути searching optimization сайту.

Для розробки були використані такі програмні засоби: Sublime Text, MySQL WorkBench 8.0 FTP Client та Adobe Photoshop CC.

Наприкінці можна зробити висновок про те, що створений сайт відповідає наданим йому цілям та характеристикам.

Також у кваліфікаційній роботі було визначено трудомісткість розробленої системи (727 люд-год), проведений підрахунок вартості роботи по створенню програми (91 657,2 грн) та розраховано час на її створення (4 міс).

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Продукт Sublime Text [Електронний ресурс]– Режим доступу: <https://www.sublimetext.com/>
2. Корпорація Oracle \ \ Продукт MySql Workbench [Електронний ресурс] – Режим доступу: <https://www.mysql.com>.
3. Этан Браун. Изучаем JavaScript: руководство по созданию современных веб-сайтов: Навчальний посібник / Этан Браун - Диалектика: ISBN 2018. – 368 с.
4. Mysql.ru. Довідковий посібник з MySQL [Електронний ресурс] – Режим доступу: [http://www.mysql.ru/docs/man/Table\\_types.html](http://www.mysql.ru/docs/man/Table_types.html)
5. М.П Малихіна. Базы даних: основи, проектування, використання: Навчальний посібник / М.П Малихіна – bhv 2004. – 499 с.
6. Metanit.com. Сайт про програмування [Електронний ресурс] – Режим доступу: <https://metanit.com/web/nodejs/>
7. Expressjs.com. Офіційний сайт ExpressJS \ \ Документація для фреймворку [Електронний ресурс] – Режим доступу: <https://expressjs.com/ru/guide/routing.html>
8. Developer.mozilla.org. Електронний довідник по JavaScript [Електронний ресурс] – Режим доступу <https://developer.mozilla.org/ru/docs/Web/JavaScript>
9. Деніел Еймор «Internet-магазини та закупівельна діяльність» / «Електронний бізнес. Еволюція та/або революція» вид. "Вільямс", 2007 р.
10. Пирогов С.В. «Огляд великих проєктів»/«Електронна комерція»
11. Юрасов А.В. Електронна комерція: навч. посібник./А.В. Юрасов - М.: Справа. 2009. - 480 с.
12. Комісарова І, Перелетова І, Петрова Е. К. Інтернет-магазин від А до Я: повне покрокове керівництво зі створення, управління / Ексмо - 2008. - 487с.
13. Управління проєктом створення інтернет-сайту Видавництво: Альпіна Паблішер, 2008 р. - 337 с.

14. Якоб Нільсен, Хоа Лоранжер «Веб-дизайн. Зручність використання веб-сайтів» Видавництво «Вільямс». 2007р.- 376 с.
15. Дроздєв Н. - Створи свій власний WordPress сайт, 2011.
16. Джон Дакетт «Основи веб-програмування з використанням HTML, XHTML та CSS» 2010 Видавництво: Ексмо, - 768 с.
17. Білл Скотт, Тереза Нейл «Проектування веб-інтерфейсів» 2010 Видавництво: Символ-Плюс, - 352 с.
18. Метью МакДональд Створення Web-сайтів. Основне керівництво» 2010 р. Видавництво: Ексмо, - 768 с.
19. Дизайн ефективного комерційного сайту Переклад П.Берестнєв, 2011, - 120с.

## КОД ПРОГРАМИ

Лістинг сайту

```
const express = require("express");
const mysql = require('mysql2');
const bodyParser = require("body-parser");

//подключеніе бд
const connection = mysql.createConnection({
  host:'localhost',
  user: 'root',
  database: 'diplom',
  password: 'aB123456789aB'
});

let total;

// тестирование подключения
const server = connection.connect((err)=>{
  if (err) {
    return console.error("Ошибка: " + err.message);
  }
  else{
    console.log("Подключеніе к серверу MySQL успешно установлено");
  }
});
//подключеніе бд

const app = express();

app.use('/public', express.static('public'));
```

```

app.use(bodyParser.urlencoded({ extended: false }));
app.use(bodyParser.json());

get_date = () => {
  let today = new Date();
  let day = String(today.getDate()).padStart(2,'0');
  let month = String(today.getMonth()+1).padStart(2,'0');
  let year = today.getFullYear();
  today = year+'-'+month+'-'+day;
  return today;
}

// url адреса страниц

app.get("/", (req, res)=>{ // главая страница
  connection.query("SELECT img.link, product.* FROM img, product WHERE
product.id_product = img.id_product", (err, result)=>{
    if(err) return console.log("Ошибка: "+err);
    res.render("index.hbs", {
      product:result
    });
  });
});

app.all("/search", (req, res)=>{ //поиск
  let search_err = req.body.test;
  let search_res = req.body.test;
  search_res = "%" + search_res + "%";
  connection.query("SELECT img.link, product.* FROM img, product WHERE
product_name LIKE ? AND product.id_product = img.id_product", search_res, (err,
result)=>{

```

```

if (err) return console.log("Ошибка: "+err);
if (result.length != 0) {
  res.render('search.hbs', {
    product: result
  })
}
else {
  res.render('search.hbs', {
    error: `По запросу "${search_err}" ничего не найдено`
  });
}
});
});

```

```

connection.query("SELECT * FROM product", (err, res)=>{ //страница товара
  if(err) return console.log("Ошибка: "+err);
  const id = res;
  const url=[];
  const prod_info = [];
  for(let i=0; i < id.length; i++){
    url[i] = "/product/"+id[i].id_product;
    prod_info[i] = id[i].id_product;
    app.get(url, (req,res)=>{
      connection.query("SELECT img.link, img.describe, product.* FROM product, img
WHERE product.id_product = ? AND product.id_product=img.id_product", prod_info[i] ,
(err, result)=>{
        if(err) return console.log("Ошибка: "+err);
        res.render('product.hbs', {
          product: result
        });
      });
    });
  });
});

```

```

    }
  });

app.get("/basket", (req,res)=>{
  connection.query("SELECT basket.*, product.product_name, img.link FROM basket,
product, img WHERE basket.id_prod = product.id_product AND
product.id_product=img.id_product", (err,result)=>{
  if(result.length == 0) {
    total = 0;
    res.render('empty_basket.hbs', {
      empty_basket: "Ваша корзина пуста!"
    })
  }
  else {
    res.render('basket.hbs', {
      order:result,
      total:result[result.length-1].total_cost,
      today_date: get_date(),
    })
  }
})
})

app.post('/basket/del', (req,res)=>{
  let id = req.body.id;
  console.log(id);
  connection.query("DELETE FROM basket WHERE id_prod = ?",id , (err,result)=>{
    if(err) return console.log("Ошибка: "+err);
    res.redirect('/basket');
  });
});
});

```



```

app.post('/basket/order', (req,res)=>{
  connection.query("SELECT * FROM clients", (err,result)=>{
    if(err) return console.log("Ошибка: "+err);
    let name = req.body.FIO;
    let phone = req.body.phone;
    let address = req.body.address;
    total = req.body.total;
    let date = req.body.date;
    let id = []
    id[0] = result[result.length-1].id_client +1;
    let clients_describe = [id[0],name,phone,address];
    const clients_array = result.filter(function (client) {
      return client.client_fio == name && client.client_phone == phone &&
client.client_address == address;
    });
    if (clients_array.length == 0) {
      connection.query("INSERT INTO clients VALUE (?,?=?,?)", clients_describe,
(err,result)=>{
        if(err) return console.log("Ошибка: "+err);
        console.log("Пользователь добавлен в БД");
        res.redirect('/complete');
      })
    }
    else {
      console.log('Пользователь уже зарегистрирован в БД')
      res.redirect('/complete');
    }
    connection.query("SELECT id_waybill FROM waybill ORDER BY id_waybill DESC
LIMIT 1", (err,result)=>{
      if(err) return console.log("Ошибка: "+err);
      let id_waybill = result[0].id_waybill+1;
      let tmp_arr = [name, phone, address];

```

```

connection.query("SELECT id_client FROM clients WHERE client_fio = ? AND
client_phone = ? AND client_address = ?",tmp_arr,(err,result)=>{
    if(err) return console.log("Ошибка: "+err);
    let served_client = result[0].id_client;
    let waybill_describe = [id_waybill,served_client,date,total];
    connection.query("INSERT INTO waybill VALUE (?,?=?,?)", waybill_describe
,(err,result)=>{
        if(err) return console.log("Ошибка: "+err);
        console.log('Создана новая накладная')
        console.log("В обработке накладная №",id_waybill);
    })
})
})
})
})
})
})

```

```

connection.query("CALL basket_procedure()", (err, res)=>{
    if(err) return console.log("Ошибка: "+err);
    total = 0;
    app.post('/basket/add', (req, res)=>{
        let prod_id = req.body.prod_id;
        let name = req.body.prod_name;
        let count = req.body.product_count;
        let cost = req.body.cost*count;
        total = total+cost;
        let product_describe =[prod_id,name,count,cost,total];
        connection.query("INSERT INTO basket VALUE (?,?=?,?,?)", product_describe,
(err,result)=>{
            if(err) return console.log("Ошибка: "+err);
            res.redirect('/basket');
        })
    })
})

```

```

})

app.get("/complete", (req,res)=>{
  connection.query("SELECT id_prod, count FROM basket",(err,result)=>{
    if(err) return console.log("Ошибка: "+err);
    let waybill_prod;
    waybill_prod= result;
    connection.query("SELECT id_waybill from waybill ORDER BY id_waybill DESC
LIMIT 1", (err,result)=>{
      if(err) return console.log("Ошибка: "+err);
      let served_waybill;
      served_waybill = result[0].id_waybill;
      console.log(served_waybill);
      for (let i=0; i<waybill_prod.length; i++){
        connection.query("INSERT INTO waybill_prod VALUES
(?,?,?)",[served_waybill,waybill_prod[i].id_prod,waybill_prod[i].count],(err,result)=>{
          if(err) return console.log("Ошибка: "+err);
        })
      }
    })
  })
})
total = 0;
connection.query("TRUNCATE TABLE basket", (err,result)=>{
  if(err) return console.log("Ошибка: "+err);
  res.render('complete.hbs', {
  })
})
})
})

```

//Порт для сервера

```
app.listen(3000, ()=> {  
    console.log("server is listening")  
})  
//Порт для сервера
```

**ВІДГУК**

**керівника економічного розділу  
на кваліфікаційну роботу бакалавра  
на тему:**

**«Розробка web-сайту інтернет - магазину електротоварів «Лайтон» з  
використанням технології Note JS»  
студента групи 122-19ск-2 Соломатіна Владислава Максимовича**

Керівник економічного розділу  
доцент каф. ПЕП та ПУ, к.е.н

Л.В. Касьяненко

## ПЕРЕЛІК ФАЙЛІВ НА ОПТИЧНОМУ НОСІЇ

Ім'я файла	Опис
Пояснювальні документи	
Диплом.doc	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Диплом.pdf	Пояснювальна записка до кваліфікаційної роботи. в форматі PDF
Програма	
Diplom.zip	Архів. Містить коди програми і откомпільовану програму
Презентація	
Презентація.ppt	Презентація кваліфікаційної роботи.