

Міністерство освіти і науки України

Тернопільський національний технічний університет імені Івана Пулюя

Комп'ютерно-інформаційних систем і програмної інженерії

(повна назва факультету )

Програмної інженерії

(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка веб-каталогу для магазину оптики

з використанням штучного інтелекту

Виконав(ла): студент(ка) 4 курсу, групи СП-41  
спеціальності 121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

|                   |          |                        |
|-------------------|----------|------------------------|
|                   |          | Здріль М.С.            |
|                   | (підпис) | (прізвище та ініціали) |
| Керівник          |          | Сиротюк О.Б.           |
|                   | (підпис) | (прізвище та ініціали) |
| Нормоконтроль     |          | Стоянов Ю.М.           |
|                   | (підпис) | (прізвище та ініціали) |
| Завідувач кафедри |          | Петрик М.Р.            |
|                   | (підпис) | (прізвище та ініціали) |
| Рецензент         |          |                        |
|                   | (підпис) | (прізвище та ініціали) |

|           |                                                                                            |
|-----------|--------------------------------------------------------------------------------------------|
| Факультет | <u>Комп'ютерно-інформаційних систем і програмної інженерії</u><br>(повна назва факультету) |
| Кафедра   | <u>Програмної інженерії</u><br>(повна назва кафедри)                                       |

«                      »  
2024 р.

на здобуття освітнього ступеня бакалавр  
(назва освітнього ступеня)

за спеціальністю 121 Інженерія програмного забезпечення  
(шифр і назва спеціальності)

студенту Здрілю Максиму Сергійовичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка веб-каталогу для магазину оптики  
з використанням штучного інтелекту

Керівник роботи Сиротюк Оксана Богданівна  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 15 » квітня 2024 року № \_\_\_\_\_

2. Термін подання студентом завершеної роботи

3. Вихідні дані до роботи Вимоги замовника, технічні умови, технічне завдання

4. Зміст роботи (перелік питань, які потрібно розробити)

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

| Розділ                                        | Прізвище, ініціали та посада консультанта | Підпис, дата   |                  |
|-----------------------------------------------|-------------------------------------------|----------------|------------------|
|                                               |                                           | завдання видав | завдання прийняв |
| Безпека життєдіяльності, основи охорони праці | Лазарюк В.В., доцент кафедри МТ           |                |                  |
|                                               |                                           |                |                  |
|                                               |                                           |                |                  |
|                                               |                                           |                |                  |
|                                               |                                           |                |                  |
|                                               |                                           |                |                  |

## КАЛЕНДАРНИЙ ПЛАН

[illegible]

Здріль М.С.  
(прізвище та ініціали)

Сиротюк О.Б.  
(прізвище та ініціали)

## РЕФЕРАТ

Кваліфікаційна робота бакалавра складається: с. \_\_\_\_, рис. \_\_\_\_, табл. \_\_\_\_, джер. \_\_\_\_

Об'єктом дослідження є сучасні методи автоматичного складання веб-сайтів з використанням штучного інтелекту. Основним завданням є реалізація дизайну та структури веб-додатку, що використовує штучні нейронні мережі.

Метою роботи є створення автоматизованої системи розробки веб-сайтів з використанням штучного інтелекту. Для досягнення поставленої мети потрібно створити алгоритм побудови веб-додатку, за допомогою якого будуть генеруватися готові рішення в реальному часі.

Методи розробки базуються на технології Python, програмна бібліотека для машинного навчання TensorFlow.

У результаті роботи був проведений аналіз предметної галузі, виявлено проблемні місця та поставлено задачу розробки готового рішення. Були сформовані вимоги та реалізовано дизайн веб-додатку та досліджено методи автоматичної розробки веб-сайтів.

Ключові слова: інформаційні технології, програмне забезпечення, веб-каталог, UML Class Diagram; UML Activity Diagram, база даних.

## ABSTRACT

The object of research is the modern methods of automatic compilation of websites using artificial intelligence. The main task is to implement the design and structure of a web application that uses artificial neural networks.

The purpose of the work is to create an automated system for developing websites using artificial intelligence. To achieve this goal you need to create an algorithm for building a web application that will generate ready-made solutions in real time.

Development methods are based on Python technology, a software library for TensorFlow machine learning.

As a result of the work, the analysis of the subject area was carried out, the problem areas were identified and the task of developing a ready-made solution was set.

Requirements were formulated and the Web application design was implemented and the methods for the automatic development of web sites were explored.

Keywords: information technologies, software, web directory, UML Class Diagram; UML Activity Diagram, database.

## ЗМІСТ

### ВСТУП

### 1 РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ

#### 1.1 Предметна область

Порівняння Web 1.0, Web 2.0, Web 3.0.

#### 1.2 Штучний інтелект

#### 1.3 Застосування ІІ у роздрібній торгівлі

##### 1.3.1. ІІ-боти.

##### 1.3.2. "Живий" інструктаж по роботі з клієнтами.

##### 1.3.3. Аналіз настрою клієнтів.

##### 1.3.4. Прогнозування попиту і аналітика.

##### 1.3.5. Рекомендаційний ІІ.

##### 1.3.6. Автоматизоване керування запасами.

##### 1.3.7. Інтелектуальні персональні рекламні оголошення.

#### 1.4. Переваги використання ІІ у роздрібній торгівлі

#### 1.5 Опис програмних інструментів та засобів розробки

##### 1.5.1. Мова PHP.

##### 1.5.2. Мова JavaScript .

##### 1.5.3. MySQL.

##### 1.5.4. Фреймворки.

#### 1.6 Структура бази даних веб-каталогу

##### 1.6.1. Характеристика використовуваної моделі даних

##### 1.6.2. Сценарії взаємодії.

##### 1.6.3. Концептуальна схема веб-додатку.

##### 1.6.4. Опис користувацьких або програмних інтерфейсів.

### 2 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАСОБУ

### 3 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

ЗАГАЛЬНІ ВИСНОВКИ

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

## ВСТУП

Інтернет розвивається, створюється безліч веб-додатків, надаються різноманітні послуги, продаються товари, обробляються файли. В даний час в Інтернеті можна знайти практично будь-який онлайн-сервіс під свої потреби. Онлайн-сервіс – це сайт, який дозволяє нам швидко і ефективно вирішити своє завдання. Працюючи в онлайн-сервісі, нам не потрібно встановлювати програми на свій комп'ютер, відсутні проблеми у сумісності між тією чи іншою програмою та операційною системою на комп'ютері користувача, немає необхідності відстежувати оновлення, онлайн-сервіс – це завжди актуальна версія програми, доступна завжди і скрізь, вона не прив'язаний до конкретного комп'ютера. Все, що потрібно для роботи — це підключення до Інтернету, всі дії можна здійснювати через браузер.

Незважаючи на різноманітність онлайн-сервісів, одним із найактуальніших у сфері інтернет-торгівлі є інтернет-магазин. Інтернет магазин дуже зручний тим, що він працює цілодобово і без вихідних, подивитися каталоги, придбати товар, може покупець з будь-якої точки світу, де є доступ до інтернету, на відміну від звичайних магазинів, у яких площа приміщення обмежена, на сайтах можна розмістити скільки завгодно товару, реклами та інформації.

Метою даної випускної кваліфікаційної роботи є, розробка веб-каталогу магазину «Оптика» із сучасним інтерфейсом, в якому будуть:

- Каталог товарів.
- Кошик покупця.
- Новини.
- Відгуки користувачів.
- Особистий кабінет.

Веб-каталог – веб-ресурс, в якому є необхідні дані компанії, а також представлені безпосередньо товари.

На сайті відсутні обмеження щодо найменувань товарів. Важливим є вже на початковому етапі враховувати зручність та гнучкість сайту для працівників, які будуть поповнювати каталог і користувачів-відвідувачів – потенційних клієнтів.

Структурно сайт-каталог є схожим на інтернет магазин. Порівнюючи його з більшим масштабованим проєктом, він є простішим при обслуговуванні, оскільки в ньому не проводиться опрацювання замовлень. Теж у ньому відсутня функція опрацювання оплати замовлення, тобто користувач має право вибору. Користувач сам переглядає, шукає, та вибирає потрібний товар, його ціну, після цього відбувається зв'язок із менеджером, з яким уточнюються всі питання стосовно оплати та доставки. Здійснення покупки проходить лише після цього. Така процедура є зручною для певних категорій товару, оскільки при їх виборі потрібно враховувати багато особливостей.

Каталогізований сайт дає можливість користувачеві отримати достатню інформацію про той чи інший товар. Важливим є інформативність, простота, зручний та швидкий пошук, дизайн сайту. Щодо власника продукту, то невід'омою частиною є зручність керування адміністративною частиною.

Об'єктом розробки є веб-каталог інтернет-магазину «Оптика».

Інтернет-магазин розроблений на CMS1 системі, в якій адміністратору сайту будуть доступні можливості зміни даних на сайті, створення нових розділів, модерація замовлень та відгуків користувачів сайту

Для досягнення поставленої мети випускної кваліфікаційної роботи необхідно вирішити наступний ряд завдань:

- Описати, що таке інтернет-торгівля, інтернет-магазин, чому актуально мати інтернет-магазин, плюси та мінуси інтернет-магазину, що таке сайт, що таке каталог, що таке система CMS.
- Описати структуру створюваного інтернет-магазину, використані технології та мови програмування.
- Вивчити структуру та документацію CMS системи.
- Розгорнути систему CMS на веб-сервері.
- Розробити веб-каталог інтернет-магазину.

## 1 РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ

### 1.1 Предметна область

Веб-стандарти - це технології, які використовуються для створення web-сторінок. Стандарти існують у вигляді технічної документації (специфікацій), яка точно описує, як це, або ця технологія повинна працювати. Документація ніяк не допоможе вивчити те, як користуватися описуваними в її технологіях (ось чому існує сайт MDN Web Docs). Вона використовується розробниками ПО для впровадження технологій (наприклад, у веб-браузерах) [1-3].

В якості прикладу наведемо стандарт HTML Living Standard. Він описує як HTML (усі елементи HTML, взіємопов'язані з ними API та інші близькі технології) повинні бути реалізовані.

Веб-стандарти створюються організаціями стандартів — інститутами, які залучають групи людей з різних компаній для узгодження того, як технології повинні застосовуватися найбільш ефективним способом у розглянутих випадках. Сама відома організація веб-стандартів - W3C. Існують і інші: WHATWG (відповідальні за модернізацію мови html), ECMA (випускають стандарти мови ECMAScript, на якому побудований JavaScript), Khronos (створюють технології для 3D-графіки, наприклад WebGL) [4-6].

Один із ключових аспектів веб-стандартів, які TimBL і W3C погодили з самого початку, полягає в тому, що сеть (і веб-технології) повинні бути вільними як для внесення вкладу, так і для використання, і не повинні бути обмежені патентами/ліцензуванням. Таким чином, будь-який може безкоштовно написати код для створення свого веб-сайту, і будь-який може внести свій вклад у процес створення стандартів, у яких прописані специфікації.

Оскільки веб-технології створюються відкрито, у співробітництві безлічі різних компаній, це означає, що жодна компанія не може їх контролювати, і це дійсно добре. Вряд чи вам сподобалося, якщо яка-то одна компанія позачергово вирішила поставити весь набір під платний доступ або випустила нову версію

HTML, кожен повинен купити, щоб продовжувати створювати веб-сайти, або, що ще більше, в який-то момент вирішила, що вона більше не зацікавлена в технології HTML і просто відключила її [4].

Це дозволяє інтернету залишатися вільним доступним суспільним ресурсом.

Популярна в області веб-стандартів фраза: "не розірви мережу". Це означає, що будь-яка нова веб-технологія повинна бути сумісна з усіма попередніми технологіями (по цьому старі сайти до них пор працюють), і з усіма наслідками (розробляються в останні технології, у свою чергу, повинні бути сумісні з наявними) [5].

#### 1.1.1. Порівняння Web 1.0, Web 2.0, Web 3.0.

Web 1.0, Web 2.0, Web 3.0 – це, можна сказати, умовні історичні етапи, які виділяються в розвитку Всесвітньої павутини. Перехід між ними не можна прив'язувати до конкретної дати або навіть року, так як він відбувається повільно і залишає багато з попередніх етапів. Виділення цих етапів може бути досить спорним [6-10].

Технології, особливості створення сайтів і поведінка користувачів Всесвітньої павутини (WWW), характерні для 90-х і початку 2000-х років, прийнято називати Веб 1.0 (Web 1.0). У цей період у WWW переважали статичні сайти. Такі сайти призначені в основному для читання, отримання інформації; не вважаються гіперпосиланнями, вони майже не містили інтерактивних елементів, мультимедіа, не надавали можливості користувачам вести діалог, обмінюватися файлами і т.д. п.

Для створення сайтів використовувались теги HTML, які виконують формувальну функцію. Часто розмітку сайту роблять шляхом створення HTML-таблиць. Це приводило до збільшення розмірів коду і ускладнювало пошук.

При застосуванні Web 1.0 мережеве підключення було повільним, що накладало відповідно обмеження. Спілкування в мережі відбувалось за допомогою форумів та чатів [6-10].

Десь приблизно в середині 2000-х методи і цілі створення сайтів почали змінюватися. З'явилась орієнтація на динамічне створення вмісту, коли користувачі самі наповнюють ресурс, спілкуються між собою і виказують відгуки прямо на сайті. Почали виникати блоги, соціальні мережі, вікі-проекти. Насьогодні така форма організації та створення контенту займає істотну долю WWW, а користуються такими сайтами більшість користувачів Інтернету.

З появою високошвидкого доступу до мережі Інтернет у Всесвітній павутині стала популярною мультимедійна інформація (відео, музика, графіка).

Для Web 2.0 характерно поява нових веб-служб, розвиток веб-програмування, покращення дизайну та зручності сайтів, нівелюється анонімність користувачів. Але при цьому, з'являється значна кількість фейкової інформації. Тому, при розробленні веб-застосунку потрібно звертати увагу на якість інформації, яка буде розміщена [11].

Web 3.0 був створений з врахуванням недоліків Web 2.0. Особливу увагу звернуто на якісну сторону сервісу та контенту. Недоліками є відсутність анонімності користувача, чітке дотримання правил веб сервісу, тощо. Як приклад можна розглядати використання хмарних технологій [11].

## 1.2 Штучний інтелект

ШІ став універсальним терміном для програм, які виконують складні завдання, які коли-то вимагали участі людини, наприклад, спілкування з клієнтами в Інтернеті або гра в шахмати. Цей термін часто використовується взаємозамінюємо з його подобластями, які включають машинне навчання (ML) і глибоке навчання [12].

Науки про штучний інтелект і мають свою специфіку. Наприклад, машинне навчання фокусується на створених системах, які навчаються і розвиваються шляхом обробки та аналізу даних. Різниця полягає в тому, що машинне навчання

завжди передбачає використання ШІ, однак ШІ не завжди розуміє машинне навчання.

Щоб використовувати можливості ШІ з максимальною вигідністю для бізнесу, необхідно наняти спеціалістів по вивченню даних. Наука про дані об'єднує статистику, інформатику та бізнес-знання для отримання цінності з різних джерел даних [14].

Розробники застосовують штучний інтелект, щоб ефективніше виконувати завдання, які в цьому випадку прийшло б робити вручну, взаємодіяти із замовниками, виявляти закономірності та вирішувати проблеми. Для початку роботи з ШІ розробниками потрібні математичні знання та вміння користуватись алгоритмами.

ШІ дає можливість відновити і поліпшити те, як ми сприймаємо навколишній світ і реагуємо на нього. Ця властивість ШІ лежить в основі інновацій. ШІ заснований на різних технологіях машинного навчання, які розпізнають шаблони в даних і формують прогнози. Він створює приватну вартість для бізнесу завдяки таким можливостям: допомагає використовувати весь потенціал даних; складає надійні прогнози і автоматизує складні завдання.

Відповідно до останнього звіту Harvard Business Review, компанії вигідно використовувати ШІ для: виявлення та недопущення порушень безпеки (44 %); усунення технічних проблем користувачів (41 %); скорочення задач по управлінню продукцією (34 %); оцінка внутрішньої відповідності нормативам у затверджених постачальників (34 %) [15].

Цінність ШІ для бізнесу підтверджує безліч прикладів успіху. Додавання технологій машинного навчання та когнітивних операцій у традиційні бізнес-процеси та додатки забезпечує підвищення зручності та продуктивності.

Тим не менше впровадження ШІ пов'язано з визначеними труднощами. Лише небагато компаній задіюють повний потенціал ШІ, і тому є кілька причин. Наприклад, якщо вони не використовують хмарні обчислення, проекти машинного навчання часто вимагають більших обчислювальних ресурсів. Вони також складні в створенні і вимагають досвіду, який користується великим

запитом, але його не хватає. Знання того, коли і де включити ці проекти, а також коли звертатися до третьої сторони, допоможе звести до мінімуму ці труднощі.

### 1.3 Застосування ШІ у роздрібній торгівлі

#### 1.3.1. ШІ-боти.

Чатботи зі штучним інтелектом, які мають доступ до різної інформації, можуть бути вбудовані в сайти електронної комерції, соціальні мережі та інші програми. Виходячи за рамки чатботів, які покладаються на кілька закладених вручну робочих процесів для конкретних тем розмови, ШІ-чатботи навчаються з різних тем — часто з використанням організаційної бази даних або бази знань — щоб вони могли більш ефективно розуміти та відповідати на різні запитання та запити клієнтів [17].

#### 1.3.2. "Живий" інструктаж по роботі з клієнтами.

Деякі ШІ-інструменти, наприклад Nuance, Gridspace і Zenarate, не беруть на себе всі функції обслуговування клієнтів, а надають «живі» консультації та пропозиції представникам служби підтримки. Ці пропозиції часто поєднуються з детальними інформаційними панелями, які дають представникам компанії інструменти та мову, необхідні для продуктоорієнтованішого спілкування з клієнтами по телефону [18].

#### 1.3.3. Аналіз настрою клієнтів.

Хоча роздрібна компанія, можливо, вже регулярно отримує та аналізує відгуки клієнтів, часто буває складно зіставити ці настрої по користувачам та в

масштабі, а ще складніше застосувати ці дані для необхідних покращень бізнес-моделі.

За допомогою моделей ШІ всі відгуки та взаємодії клієнтів автоматично збираються та аналізуються, що дозволяє ритейлерам швидше та краще зрозуміти, як клієнти ставляться до бренду. Зокрема, за допомогою великих мовних моделей та моделей генеративного ШІ можна збирати неструктуровані дані, такі як текст, з різних джерел та одночасно відсіювати спамерські відгуки, щоб вони не каламутили воду в аналізі.

Крім покращення аналізу настроїв покупців, ШІ дозволяє ритейлерам застосовувати отримані інсайти для більш персоналізованої роботи з покупцями, включаючи рекламу з урахуванням їх побажань та більш точну сегментацію аудиторії.

#### 1.3.4. Прогнозування попиту і аналітика.

Аналітика даних на основі ШІ дозволяє користувачам, у тому числі і тим, хто не є фахівцями в науці про дані, краще розуміти все різноманіття наявних у їх розпорядженні даних [19-21].

Ці моделі можуть аналізувати дані в різних форматах і великих обсягах, вивчаючи історичні та поточні метадані про закупівлі в різних категоріях для більш точного прогнозування попиту. Наприклад, власники бізнесу можуть використовувати ШІ-аналіз, щоб дізнатися, що окуляри, які вони продавали минулого року, не тільки швидко розійшлися, а й отримали позитивну оцінку у відгуках та розмовах покупців.

Вони також можуть дізнатися, що хоча окуляри добре продавались в одних регіонах, це не стало хітом продажів в інших. Озброївшись цими даними, ритейлер зрозуміє, куди йому слід перенаправити цей товар, на яких ринках потрібно більше запасів, а на яких менше.

Для тих, хто не займається вивченням даних професійно, інструменти ШІ-аналізу особливо ефективні в плані аналітики, тобто аналітики, яка дає

рекомендації щодо коригування бізнес-тактики в майбутньому на основі поточних даних. Завдяки природно-мовному підходу ці інструменти ІІІ допомагають бізнес-користувачам з різних підрозділів та областей знань використовувати ці дані для досягнення найкращих результатів.

#### 1.3.5. Рекомендаційний ІІІ.

Грунтуючись на метаданих окремих користувачів, їх минулих покупках, чуйності до реклами, аналізу настроїв та інших даних, ІІІ у роздрібній торгівлі тепер може рекомендувати покупцям товари та послуги, які вони, швидше за все, захочуть придбати, але, можливо, не розглядали б в іншому випадку.

Дійсно, рекомендаційний ІІІ є одним з напрямків ІІІ, що найбільш швидко розвиваються, в ритейлі завдяки тому, що він добре вловлює переваги покупців і дозволяє їх монетизувати [19-21].

#### 1.3.6. Автоматизоване керування запасами.

Прогнозування попиту з допомогою ІІІ — одне із способів, з допомогою якого роздрібні компанії тепер точніше пророкують, скільки товарних запасів їм потрібно і де і коли вони мають бути розміщені. ІІІ-аналітика даних також може допомогти ритейлерам визначити, коли слід змінити ціни, як сезонні закупівлі впливають на товарні запаси та рух по ланцюжку поставок, і в яких випадках повернення товару покупцями потребує частішого оновлення запасів.

Фізично більш відчутним є те, що робототехніка, керована ІІІ, також може бути використана для підтримки автоматизованого управління запасами. Дедалі більше складів роздрібної торгівлі покладаються на роботів-помічників з ІІІ, які сканують запаси і відстежують їх рівень, та був поповнюють чи скорочують запаси за необхідності.

### 1.3.7. Інтелектуальні персональні рекламні оголошення.

Маркетологи та менеджери з реклами працюють за лаштунками, аналізуючи ефективність реклами та приймаючи рішення про те, які зміни слід внести для підвищення залученості аудиторії. ШІ повсюдно бере на себе це завдання і приймає більш точні рішення про націлення, насамперед тому, що здатний швидше аналізувати більшу кількість точок даних про залучення.

Крім того, ШІ частіше помічає закономірності в даних, які людина може не помітити, і може приймати рішення на основі минулих даних, чи це оновлення рекламного тексту на основі настрою користувачів або зміна місця оголошення на веб-сторінці на основі попередніх даних теплової картки.

ШІ також може націлювати рекламні оголошення на окремих людей, ґрунтуючись на їх метаданих, що дозволяє підвищити ймовірність їхньої зацікавленості та залученості. Нарешті, ШІ навчається та оновлює рекламні оголошення в режимі реального часу, забезпечуючи їхню постійну оптимізацію під поточну аудиторію.

Спрощення процесу оформлення замовлення. Завдяки поєднанню біометрії та технології розпізнавання за допомогою ШІ вітрини магазинів починають спрощувати процедуру оформлення замовлення, у тому числі і в офлайн-магазинах.

Наприклад, деякі магазини тепер дозволяють постійним покупцям просто взяти свої товари та вийти за двері; технологія розпізнавання та сканування з використанням ШІ ідентифікує покупця та автоматично виставляє рахунок, не вимагаючи від нього участі в оформленні покупки в магазині.

Для покупців, які бажають отримати більш зручний досвід віртуального шопінгу, багато компаній додають до своїх програм для роздрібної торгівлі допоміжні елементи ШІ. Ці функції можуть робити пропозиції щодо покупки або інтегруватися з віртуальними гаманцями для зручнішого здійснення покупок.

## 1.4 Переваги використання ІІІ у роздрібній торгівлі

Багато ритейлерів побоюються, що використання ІІІ в їхній бізнес-моделі призведе до втрати особистого контакту з покупцями, проте на сьогоднішній день воно вже дає ряд переваг, включаючи допомогу у створенні більш зручного клієнтського досвіду [27].

- Акцент на досвіді клієнта. Рекламні оголошення націлені на те, що дійсно потрібно користувачам, чати можуть більш чітко відповідати на запитання клієнтів у зручний для них час, а додатки, керовані ІІІ, відкривають користувачам доступ до нових типів купівельного досвіду, які відповідають їх перевагам. Незважаючи на те, що з впровадженням ІІІ в ритейлі може зменшитися кількість контактів між людьми, покупці все одно залишаються у вигравшій від орієнтації на клієнта.

- Більше можливостей для автоматизації. ІІІ дозволяє автоматизувати розмовні робочі процеси, управління запасами та ланцюжками поставок, а також інші завдання, що повторюються роздрібною торгівлю, які традиційно вимагали участі людини. Це знижує ймовірність людських помилок і вивільняє час співробітників на вирішення більш стратегічних завдань, що може призвести до підвищення продуктивності організації.

- Зниження впливу помилок персоналу. ІІІ може відстежувати весь життєвий цикл ланцюжка поставок та управління запасами та виявляти помилку відразу ж після її виникнення. Це дозволяє усувати помилки при складуванні та відвантаженні до того, як вони призведуть до невдоволення клієнтів або нестачі запасів.

- Оптимізація цифрового маркетингу та аналітики. Аналітичні інструменти на базі ІІІ демократизують процес аналізу, дозволяючи використовувати природну мову та контекстні пояснення та отримувати більш детальну та точну прогнозну аналітику, корисну для відділів маркетингу та продажу. Крім того, ці

інструменти можуть аналізувати більше обсягів і типів даних, ніж більшість традиційних інструментів маркетингової аналітики.

- Зменшення числа точок зіткнення з людиною. Оскільки ШІ бере на себе різні функції обслуговування клієнтів та управління запасами, рітейлери можуть скоротити штат співробітників або зосередити їхню увагу на більш стратегічних завданнях. В умовах дефіциту на ринку праці ШІ допоможе заповнити виробничі прогалини, скорочуючи вимоги до перепідготовки та найму персоналу.

## 1.5 Опис програмних інструментів та засобів розробки

У процесі вибору було проаналізовано різні мови програмування та технології.

Розглянемо дві найбільш популярні мови програмування, що застосовуються у веб-розробці: PHP, Javascript, а також найбільш популярну систему управління базами даних: MySQL [28, 29, 30, 33].

### 1.5.1. Мова PHP.

PHP (Hypertext PreProcessor, препроцесор гіпертексту) – мова програмування, що виконується на стороні веб-сервера, спроектована Расмусом Лердорфом (Rasmus Lerdorf) як інструмент створення динамічних та інтерактивних веб-сайтів.

Ця мова виявилася досить гнучкою і потужною, тому набула великої популярності і використовується в проектах будь-якого масштабу: від простого блогу до найбільших веб-додатків в Інтернеті [28, 33].

Переваги PHP:

- 1) це вільне програмне забезпечення, яке розповсюджується під особливою ліцензією (PHP license);

- 2) усі етапи його освоєння досить прості;
- 3) цією мовою користується широке коло користувачів та розробників, що забезпечує мову серйозну підтримку;
- 4) у PHP глибоко розвинена підтримка баз даних;
- 5) для нього розроблено широкий спектр розширень та бібліотек;
- 6) PHP можна застосовувати в ізольованому середовищі;
- 7) володіє програмним інтерфейсом розширень та засобами для організації веб-сесій;
- 8) PHP можна використовувати як повноцінну заміну пропрієтарного середовища ASP (Active Server Pages), розробленого Microsoft, розгортається практично на будь-якому сервері;
- 9) вже на сьогодні PHP портований на дуже велику кількість операційних систем та апаратних платформ.

#### Недоліки PHP:

- 1) його не можна використовувати для розробки desktop-додатків чи системних компонентів;
- 2) невеликий та слабкий набір коштів для роботи з винятками;
- 3) налаштування сервера та розгортання додатків утруднені через те, що на базовий синтаксис мови впливають глобальні параметри конфігурації;
- 4) у PHP передача об'єктів відбувається за значенням, до чого доводиться пристосовуватися більшості програмістів, оскільки в більшості інших мов об'єкти передаються за посиланням;
- 5) веб-програми, написані цією мовою, зазвичай мають низький рівень безпеки.

PHP застосовується у таких відомих проектах: 1) Zend; 2) Yahoo; 3) Facebook; 4) Google; 5) NASA; 6) W3C.

### 1.5.2. Мова JavaScript .

JavaScript - це легка, кросплатформна та інтерпретована мова сценаріїв. Її часто застосовують для розроблення web-сторінок, також вона використовується багатьма небраузерними середовищами. JavaScript можна використовувати як для розробки на стороні клієнта, так і для розробки на стороні сервера. JavaScript містить стандартну бібліотеку об'єктів (Array, Date, Math) та базовий набір мовних елементів (оператори, управляючі структури тощо).

На стороні клієнта: JavaScript надає об'єкти для керування браузером та його об'єктною моделлю документа (DOM). Так, розширення клієнтів дають можливість застосувати розміщення елементів у HTML-форматі та реагувати на дії користувача (клікання, ввід даних, навігація тощо). Корисними бібліотеками для клієнтської частини є AngularJS, ReactJS, VueJS та ін.

Сервер JavaScript видає об'єкти, які належать до запускання на сервері. Наприклад, якщо розширення на стороні сервера дозволяють застосунок взаємодіяти з базою даних і забезпечувати безперервність інформації між викликами або працювати на сервері із файлами. Найвідоміший у наш час корисний фреймворк — це node.js.

JavaScript можна додати до HTML-файлу двома способами.

Внутрішній JS: можна додати JavaScript безпосередньо до HTML-файлу, написавши код усередині тега `<script>`, який може бути розміщений всередині тега `<head>` або `<body>` в залежності до вимог.

Зовнішній JS: можна написати код JavaScript в іншому файлі з розширенням .js, а потім зв'язати цей файл усередині тегу `<head>` HTML-файлу, до якого необхідно додати код [28-33].

### 1.5.3. MySQL.

Це вільна система керування БД (СУБД). Вона належить організації SunMicrosystems, яка займається її розробкою та підтримкою. Поширення MySQL

профодиться за ліцензією GNU, або з використанням власної комерційної ліцензії. Крім того, ліцензійні користувачі можуть надсилати замовлення, за якими розробляється відповідна функціональність, що зумовило появу механізму реплікації в ранніх версіях MySQL [28, 30, 32, 33].

MySQL найефективніше використовується у додатках малого та середнього масштабу. Він входить до стандартних пакетів розробки веб-застосунків для таких операційних систем, як Windows (або WAMP) та Linux (LAMP).

У більшості випадків MySQL виконує роль сервера, до якого звертаються із запитамі локальні або віддалені клієнти, проте MySQL може бути включений і в автономні програми за рахунок дистрибутива, включеного в бібліотеку внутрішнього сервера.

MySQL здатний підтримувати кілька механізмів бази даних. Ці механізми визначають, яким чином він у конкретний момент часу виробляє зберігання, вилучення та обробку даних. Відповідно, кожен із механізмів має свій індивідуальний набір можливостей та переваг. Поступово потужність та швидкість наявних механізмів зростає [28, 30, 32, 33].

#### 1.5.4. Фреймворки.

Використовувався фреймворк з відкритим кодом Laravel. Він призначений для розробок із використанням архітектурної моделі MVC (Model View Controller). Laravel випущено під ліцензією MIT. Вихідний код проекту розміщується на GitHub. Логіка програми – працює з використанням шаблону проєктування HMVC .

Зворотня маршрутизація поєднує генеровані додатком посилання та маршрути. Це дозволяє замінювати останні та автоматично оновлювати посилання. У Laravel автоматично генеруються кінцеві URL - адреси з допомогою іменних маршрутів.

Laravel не потребує підключення файлів визначень класів PHP в include, вони завантажуються автоматично (тільки ті, які потрібно використати).

Даний фреймворк має можливість вносити зміни в код програми та поєднує їх зі змінами, які вносяться у структуру БД. Він суттєво спрощує розкриття і оновлення програми.

Також, за допомогою вбудованого в Laravel механізму, спрощено процедуру генерування сторінок

## 1.6 Структура бази даних веб-каталогу

База даних є складовою сучасного web-додатку. В ній зберігають інформацію, які підтримує працездатність програмного продукту. Вона забезпечує простоту і швидкість керування даними. Її використання дає можливість забезпечити пошук інформації, поділ на сторінки, авторизувати користувача [28, 30, 32, 33].

Логічна модель даних є початковим зразком майбутньої БД. Основним засобом розробки логічної моделі даних зараз є різні варіанти ER-діаграм. Розробка БД починається з розробки структури її таблиць [28, 30, 32, 33].

Структура бази даних веб-застосування залежить від конкретної задачі та функціональності програми, але в загальному випадку може включати такі таблиці:

1. Користувачі – містить інформацію про зареєстрованих користувачів, таку як ім'я, прізвище, електронну пошту, пароль та дату реєстрації.
2. Профіль користувача – містить додаткову інформацію про користувача, таку як адресу, телефон тощо.
3. Продукти – містить інформацію про продукти, таку як найменування, опис, зображення, ціна тощо.
4. Категорії - містить інформацію про категорії продуктів, які використовують для групування продуктів.

5. Замовлення – містить інформацію про замовлення користувачів, таку як ідентифікатор користувача, дату замовлення, вартість замовлення тощо.

6. Елементи замовлення - містить інформацію про продукти, включені до замовлення, таку як ідентифікатор замовлення, ідентифікатор продукту, кількість і т. д.

7. Відгуки - містить інформацію про відгуки користувачів на продукти, такі як ідентифікатор користувача, продукту, текст відкликання, дату відкликання тощо.

Крім того, можуть бути додаткові таблиці, які залежать від функціональності програми, наприклад таблиця з інформацією про доставку, таблиця з інформацією про платежі тощо.

Додаткові таблиці можуть включати:

8. Сесії - містить інформацію про поточні сеанси користувачів, такі як ідентифікатор сеансу, ідентифікатор користувача, дату початку та дату закінчення сеансу.

9. Повідомлення - містить інформацію про повідомлення, що надсилаються користувачам, такі як ідентифікатор користувача, тип повідомлення, текст повідомлення та дату надсилання.

10. Журнали дій – містить інформацію про дії користувачів у додатку, такі як ідентифікатор користувача, тип дії, дату дії тощо. Ця інформація може бути корисна для аналізу поведінки користувачів і поліпшення досвіду користувача.

11. Кошик містить інформацію про продукти, додані в кошик користувачем, такі як ідентифікатор користувача, ідентифікатор продукту, кількість і т.д.

12. Адміністратори – містить інформацію про адміністраторів програми, таку як ім'я користувача, електронну пошту, пароль та права доступу.

Таблиці залежать від конкретної задачі та функціональності програми, і можуть змінюватись в залежності від вимог програми.

### 1.6.1. Характеристика використовуваної моделі даних

У кваліфікаційній роботі було використано реляційну модель даних у структурі БД [28, 30, 32, 33].

Реляційна модель є найбільш поширеною на сьогоднішній день. Об'єкти та зв'язки між ними за такої моделі даних представлені таблицями, причому зв'язки в них розглядаються як об'єкти. Стовпці в такій таблиці називаються полями, а рядки записами.

### 1.6.2. Сценарії взаємодії.

- Короткий опис функції: Пошук товару.

Первинні учасники: Користувач незареєстрований.

Передмова: Відкрито сторінку «головна» або «каталог».

Результат: Необхідний товар знайдено.

Основний сценарій:

1. Користувач відкриває сторінку "магазин" або "каталог".
2. Користувач вбиває у пошук потрібний товар.
3. Система пошуку виводить необхідний товар початку каталогу.
4. Користувач переходить на внутрішню сторінку товару.
5. Користувачеві показано сторінку з інформацією про товар, ціну та способами оформлення.

- Короткий опис функції: Оформлення замовлення

Первинні учасники: Користувач незареєстрований

Передмова: Відкрита сторінка «магазин» / «каталог» / внутрішня сторінка товару

Результат: Замовлення оформлене з урахуванням внесених даних користувачем

Основний сценарій:

1. Користувач вибирає потрібний товар на сторінці «магазин»/«каталог»/внутрішня сторінка товару, натискає кнопку «додати в кошик».

2. Користувачеві показано сторінку «кошик».
3. Користувач натискає кнопку "оформити замовлення".
4. Користувачеві показано сторінку оформлення замовлення.
5. Користувач вводить свої дані у форму заявки та натискає кнопку «оформити».
6. Користувачеві показано сторінку успішного оформлення замовлення.

- Короткий опис функції: Додавання товару

Первинні учасники: Зареєстрований адміністратор

Передумова: Відкрито модуль додавання/редагування/видалення товару

Результат: Новий товар доданий до бази даних

Основний сценарій:

1. Адміністратор натискає кнопку «створити товар» у модулі додавання/редагування/видалення товару.
2. Адміністратору показано сторінку з полями введення характеристик майбутнього товару.
3. Адміністратор вводить потрібні дані у форму та натискає кнопку «зберегти».
4. Користувачеві показано сторінку збереженого товару.

### 1.6.3. Концептуальна схема веб-додатку.

На перших етапах розробки була відображена концептуальна схема веб-застосунку, на якій зображено основний маршрут користувача.

### 1.6.4. Опис користувацьких або програмних інтерфейсів.

Головна сторінка інтернет магазину відображена на рисунку 1.2. На цій сторінці розміщені наступні елементи: меню сайту, рубрикатор з розділами каталогу та брендами зліва, основний текст, 2 каталоги з фільтрами «Новинки» та «Лідери продажу».

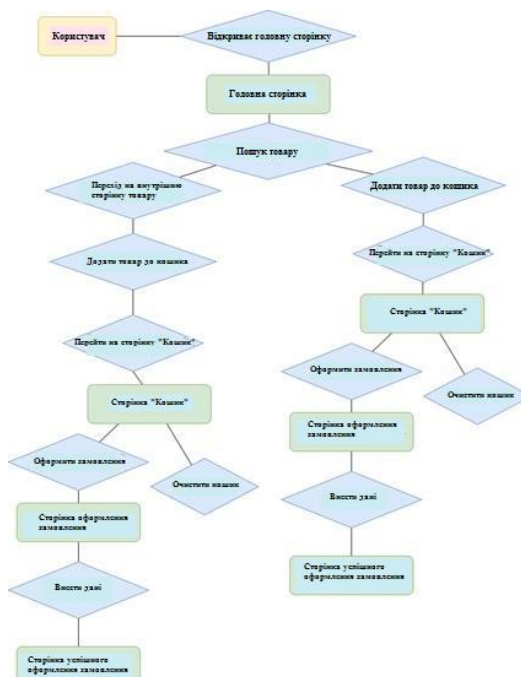


Рисунок 1.1 –Концептуальна схема

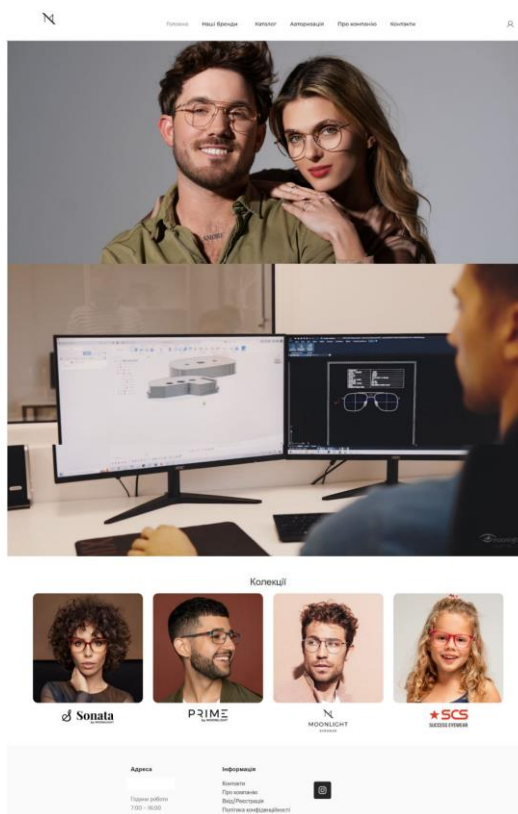


Рисунок 1.2 – Головна сторінка

Сторінка кошика показана на рисунку 1.3. Тут елементи навігації залишаються ідентичними з головною сторінкою. Є можливість збільшити кількість товару або видалити товар.

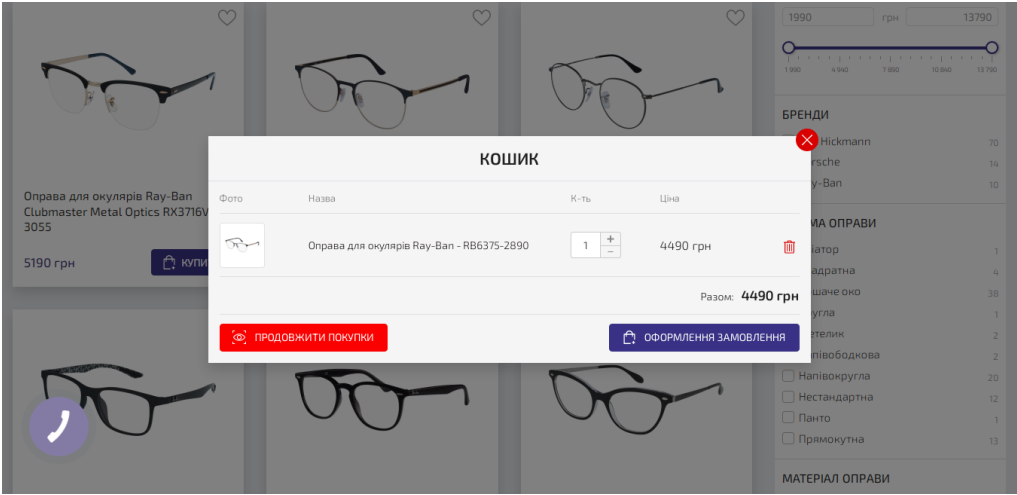


Рисунок 1.3 – «Кошик»

Функціональний код сторінки показано далі.

```
andTableSeeder.php | CategoryTableSeeder.php | Basket.php | Brand.php | Category.php | TrustProxies.php | Cate...

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\ModelNotFoundException;
use Illuminate\Support\Facades\Cookie;

class Basket extends Model {
    use HasFactory;

    /**
     * Связь «многие ко многим» таблицы 'baskets' с таблицей 'products'
     */
    public function products() {
        return $this->belongsToMany(Product::class)->withPivot('quantity');
    }

    /**
     * Увеличивает кол-во товара $id в корзине на величину $count
     */
    public function increase($id, $count = 1) {
        $this->change($id, $count);
    }

    /**
     * Уменьшает кол-во товара $id в корзине на величину $count
     */
    public function decrease($id, $count = 1) {
        $this->change($id, -1 * $count);
    }

    /**
     * Изменяет количество товара $id в корзине на величину $count;
     * если товара еще нет в корзине - добавляет этот товар; $count
     * может быть как положительным, так и отрицательным числом
     */
    private function change($id, $count = 1) {
        if ($count == 0) {
            return;
        }
        $id = (int)$id;
```

Рисунок 1.4 — Модель «Кошик»

```

use ...

class BasketController extends Controller {

    private $basket;

    public function __construct() {
        $this->basket = Basket::getBasket();
    }

    /**
     * Показывает корзину покупателя
     */
    public function index() {
        $products = $this->basket->products;
        $amount = $this->basket->getAmount();
        return view('basket.index', compact('products', 'amount'));
    }

    /**
     * Форма оформления заказа
     *
     * @param Request $request
     * @return Illuminate\Http\Response
     */
    public function checkout(Request $request) {
        $profile = null;
        $profiles = null;
        if (auth()->check()) { // если пользователь аутентифицирован
            $user = auth()->user();
            // ...и у него есть профили для оформления
            $profiles = $user->profiles;
            // ...и был запрошен профиль для оформления
            $prof_id = (int)$request->input('profile_id');
            if ($prof_id) {
                $profile = $user->profiles()->whereIdAndUserId($prof_id, $user->id)->first();
            }
        }
    }
}

```

Рисунок 1.5 — Контролер «Кошик»

```

@extends('layout.site', ['title' => 'Ваша корзина'])

@section('content')
<h1>Ваша корзина</h1>
@if (count($products))
<form action="{{ route('basket.clear') }}" method="post" class="text-right">
    @csrf
    <button type="submit" class="btn btn-outline-danger mb-4 mt-0">
        Очистить корзину
    </button>
</form>
<table class="table table-bordered">
    <tr>
        <th>№</th>
        <th>Наименование</th>
        <th>Цена</th>
        <th>Кон-во</th>
        <th>Стоимость</th>
    </tr>
    @foreach($products as $product)
    <tr>
        <td>{{ $loop->iteration }}</td>
        <td>
            <a href="{{ route('catalog.product', ['product' => $product->slug]) }}">
                {{ $product->name }}
            </a>
        </td>
        <td>{{ number_format($product->price, 2, '.', '') }}</td>
        <td>
            <form action="{{ route('basket.minus', ['id' => $product->id]) }}"
                method="post" class="d-inline">
                @csrf
                <button type="submit" class="m-0 p-0 border-0 bg-transparent">
                    <i class="fas fa-minus-square"></i>
                </button>
            </form>
            <span class="mx-1">{{ $product->pivot->quantity }}</span>
            <form action="{{ route('basket.plus', ['id' => $product->id]) }}"

```

Рисунок 1.6 — Представлення «Кошик»

Сторінку каталог відображено на рисунках 1.7 і 1.8. На цій сторінці елементи навігації також залишаються ідентичними з головною сторінкою. У центрі розташована інформація про розділи каталогу та популярні бренди з функцією переходу на внутрішню сторінку кожного підрозділу.

У внутрішній сторінці підрозділу знаходяться плитки з товарами (див. рис. 1.9), функціонал дозволяє додати товар у кошик або перейти на внутрішню сторінку товару з детальною інформацією про нього.

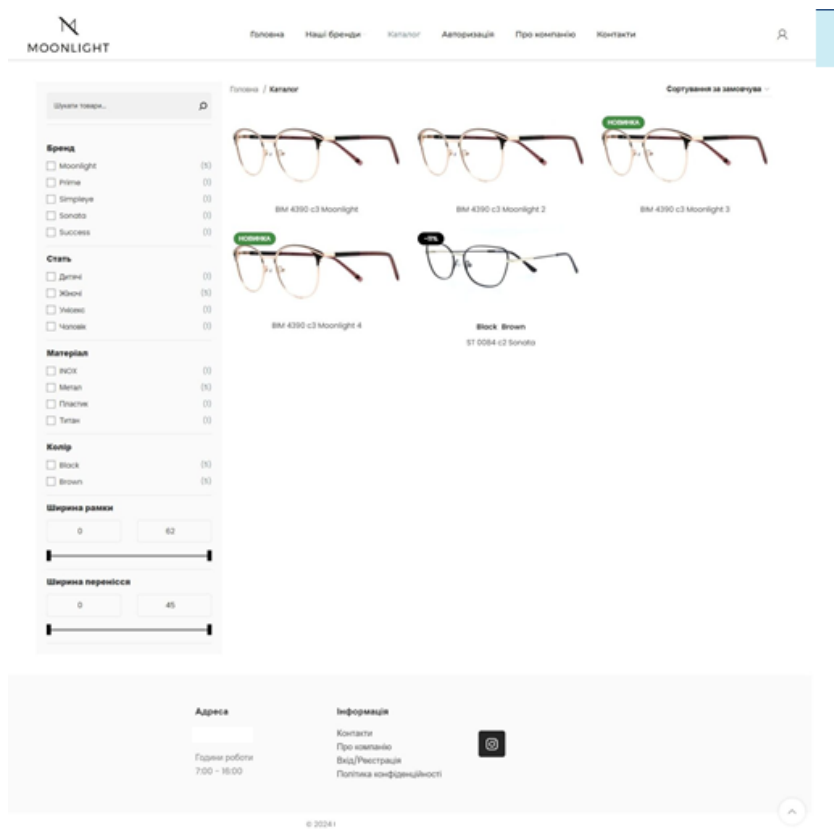


Рисунок 1.7 — Сторінка каталогу 1

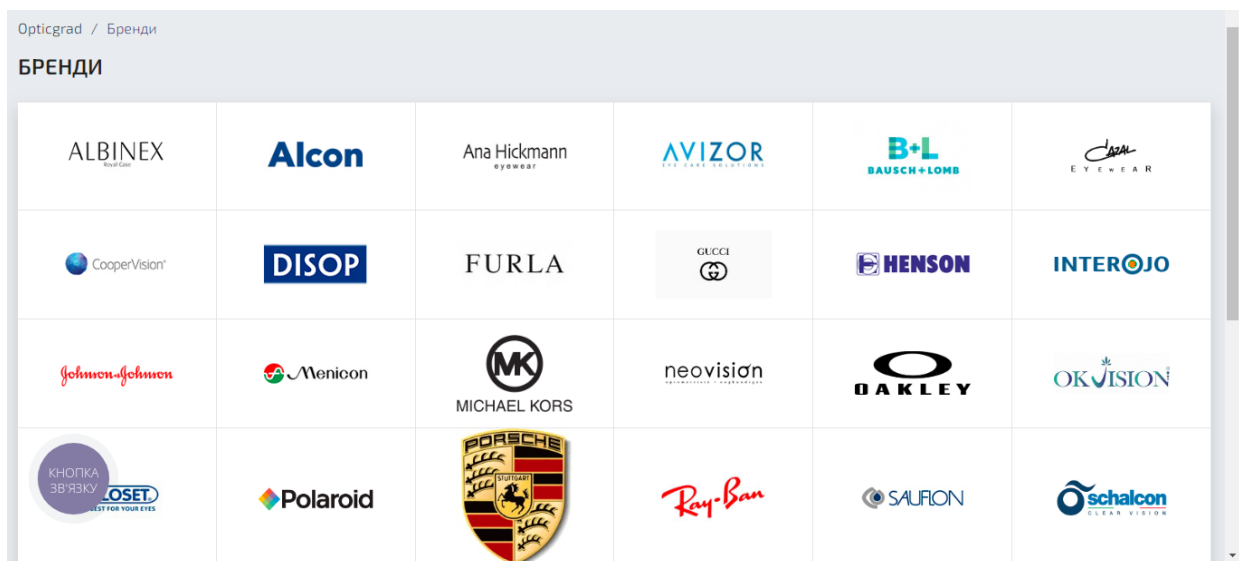


Рисунок 1.8 – Сторінка каталогу 2

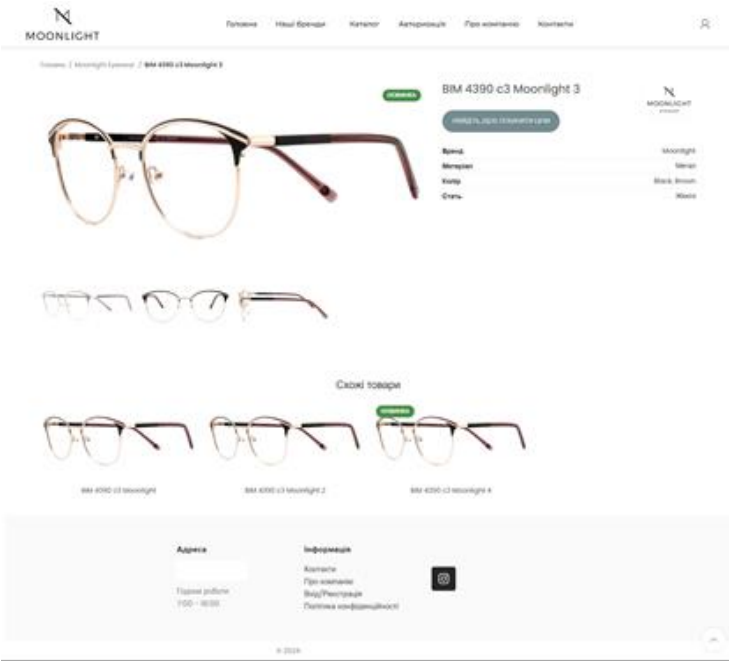


Рисунок 1.9 – Внутрішня сторінка підрозділу каталогу

Функціональний код сторінки каталогу показано на рисунках 1.10-1.12.

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Category extends Model {
    use HasFactory;

    protected $fillable = [
        'parent_id',
        'name',
        'slug',
        'content',
        'image',
    ];

    /**
     * Связь «один ко многим» таблицы 'categories' с таблицей 'products'
     *
     * @return \Illuminate\Database\Eloquent\Relations\HasMany
     */
    public function products() {
        return $this->hasMany(related: Product::class);
    }

    /**
     * Связь «один ко многим» таблицы 'categories' с таблицей 'categories'
     *
     * @return \Illuminate\Database\Eloquent\Relations\HasMany
     */
    public function children() {
        return $this->hasMany(related: Category::class, foreignKey: 'parent_id');
    }

    /**
     * Связь «один ко многим» таблицы 'categories' с таблицей 'categories', но
     * позволяет поличить не только дочерние категории, но и дочерние-дочерние
     */
}
```

Рисунок 1.10 — Модель Категорії

```

<?php

namespace App\Http\Controllers;

use ...

class CatalogController extends Controller {

    public function index() {
        $roots = Category::where('parent_id', 0)->get();
        $brands = Brand::popular();
        return view('catalog.index', compact('roots', ...var_names));
    }

    public function category(Category $category, ProductFilter $filters) {
        $products = Product::categoryProducts($category->id)
            ->filterProducts($filters)
            ->paginate(6)
            ->withQueryString();
        return view('catalog.category', compact('category', ...var_names));
    }

    public function brand(Brand $brand, ProductFilter $filters) {
        $products = $brand
            ->products() // возвращает построитель запроса
            ->filterProducts($filters)
            ->paginate(6)
            ->withQueryString();
        return view('catalog.brand', compact('brand', ...var_names));
    }

    public function product(Product $product) {
        return view('catalog.product', compact('product'));
    }

    public function search(Request $request) {
        $search = $request->input('key: query');
        $query = Product::search($search);
        $products = $query->paginate(6)->withQueryString();
        return view('catalog.search', compact('products', ...var_names));
    }
}

```

Рисунок 1.11 — Контролер Каталог

```

@extends('layout.alpha', ['title' => 'Каталог товаров'])

@section('content')
    <h1>Каталог товаров</h1>

    <div class="row">
        <div class="col">
            <h2 class="mb-4">Разделы каталога</h2>
            <div class="row">
                @foreach ($roots as $root)
                    <div class="col">
                        @include('catalog.part.category', ['category' => $root])
                    </div>
                @endforeach
            </div>
        </div>
        <div class="col">
            <h2 class="mb-4">Популярные бренды</h2>
            <div class="row">
                @foreach ($brands as $brand)
                    <div class="col">
                        @include('catalog.part.brand', ['brand' => $brand])
                    </div>
                @endforeach
            </div>
        </div>
    </div>
@endsection

```

Рисунок 1.12 – Представления «Каталог»

## 2 ТЕСТУВАННЯ ПРОГРАМНОГО ЗАСОБУ

### 2.1 Оптимізація каталогу інтернет-магазину

Автоматизація імпорту даних є важливим для ефективного управління інформаційними системами та інтернет-магазинами.

CSV (Comma - Separated Values) – поширений формат для збереження і передачі даних, особливо у сфері електронної комерції. CSV-файли містять дані, розділені комами або іншими роздільниками, і являють собою таблиці, де кожен рядок відповідає запису, а стовпці відповідають полям даних.

Автоматизація імпорту CSV-файлів до інформаційних систем та інтернет-магазинів може бути досягнута за допомогою різних інструментів. Один із таких інструментів - це спеціальне програмне забезпечення для автоматичного завантаження даних у систему. Таке програмне забезпечення може бути інтегроване з інформаційною системою або інтернет-магазином і налаштоване на автоматичний імпорт CSV-файлів з певної папки або з певної URL-адреси.

Іншим способом автоматизації імпорту CSV - файлів є використання скриптів, написаних мовами програмування, таких як Python або PHP. Такі скрипти можуть завантажувати CSV-файли з певної папки чи URL-адреси, обробляти їх та завантажувати дані в інформаційну систему або інтернет-магазин.

Перевагою автоматизації імпорту CSV-файлів є можливість істотно скоротити витрачений час на ручне введення даних у систему. Замість того, щоб вручну вводити кожен запис, можна завантажити файл CSV і обробити його автоматично. Це дозволяє скоротити час на обробку даних та збільшити продуктивність бізнес-процесів.

Крім того, це також підвищує точність даних, оскільки зменшується імовірність виникнення помилок, пов'язаних із ручним внесенням інформації.

Це суттєво для інтернет-магазинів, де помилки даних можуть призвести до неправильної ціноутворення, неправильного опису товарів або неправильної обробки замовлень. Автоматизація імпорту CSV-файлів також спрощує процес

оновлення даних в інформаційній системі чи інтернет-магазині. Наприклад, якщо постачальник змінив ціни на свої товари або додав новий товар в асортимент, можна просто завантажити оновлений CSV-файл в систему, а скрипти автоматично оновлять дані і ціни на сайті.

Несумісність інструментів автоматичного завантаження даних з інтернет-магазином є однією з потенційних проблем, яку необхідно враховувати під час автоматизації імпорту CSV-файлів.

Одна з причин несумісності може бути пов'язана з різними форматами даних. Наприклад, інтернет-магазин може вимагати певний формат даних для завантаження товарів, а CSV-файл може не відповідати цьому формату.

У ході виконання роботи натрапив на наступну проблему. Менеджер інтернет-магазину заповнює Excel із товарами для імпорту, вказуючи в «Категоріях» та Брендах найменування, а в базі зберігається ідентифікатор. Потрібно було скласти алгоритм зіставлення даних.

Також може виникнути проблема з відповідністю структури даних у CSV-файлі та в інтернет-магазині.

Важливо також тестувати та перевіряти правильність імпортованих даних після кожного оновлення, щоб уникнути помилок та неправильної обробки інформації в інтернет-магазині.

У цій кваліфікаційній роботі було обрано варіант автоматизації імпорту CSV-файлів на основі скриптів, написаних мовами програмування, таких як Python або PHP.

У контролері `\App\Http\Controllers\Admin\ProductController` був створений екшн `import` для відображення уявлення, екшн `importCsv` для обробки POST-запитів та допоміжна функція `UR_exists` для реалізації автоматизованого завантаження зображень за посиланням з CSV.

```

public function importCsv(Request $request) {
    $csvFile = $request->file('csv_file');

    $csvFile = $request->file('csv_file');
    $csv = Reader::createFromPath($csvFile, 'r')->setDelimiter(',');
    $csv->setHeaderOffset(0);

    foreach ($csv->getRecords() as $row) {

        $product = new Product();

        $product->category_id = Category::where('name', $row['Категория'])->first()?->id;
        $product->brand_id = Brand::where('name', $row['Бренд'])->first()?->id;
        $product->name = substr($row['Название товара'], 0, 90);
        $product->content = $row['Текст'];
        $product->slug = Str::slug($row['Название товара']) . rand(1, 999) . time();
        $product->price = $row['Цена'];

        $product->isnew = $row['Новинка?'];
        $product->hit = $row['Хит?'];
        $product->sale = $row['Скидка?'];

        if($row['Ссылка на картинку'] !== null) {
            if (static::UR_exists($row['Ссылка на картинку'])) {
                $image = file_get_contents($row['Ссылка на картинку']);
                $fileName = time() . '-' . Str::random(10) . '.jpg';
                \Storage::put('public/catalog/product/image/' . $fileName, $image);
                \Storage::put('public/catalog/product/thumb/' . $fileName, $image);
                \Storage::put('public/catalog/product/source/' . $fileName, $image);

                $product->image = $fileName;
            }
        }

        $product->save();
    }

    return redirect()->back()->with('success', 'CSV file imported successfully.');
```

Рисунок 2.1 – Метод importCsv

```

public function import() {
    return view('admin.product.import');
}

static function UR_exists($url){
    $headers=get_headers($url);
    return strpos($headers[0],"200 OK")?true:false;
}
```

Рисунок 2.2 – Методи import та допоміжна функція UR\_exists

Основним методом є метод importCsv, він влаштований наступним чином:

1. Спочатку метод приймає файл із POST-запиту
2. Потім створює екземпляр об'єкта CSV
3. Перший рядок пропускає, вважаючи його заголовком
4. Для інших рядків, перебирає їх у циклі та:
  - а. Створює екземпляр моделі товарів
  - б. Наповнює кожне поле моделі даними з рядка файлу CSV

с. У випадку, якщо заповнений стовпчик «Посилання на картинку», алгоритм перевіряє її доступність за віддаленою адресою за допомогою функції UR\_exists

d. Поміщає картинку у сховище

e. Записує найменування картинки до бази даних

## 3 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 3.1 Долікарська допомога при ураженні електричним струмом

Долікарська допомога – це комплекс медичних заходів, спрямованих на надання медичної допомоги при невідкладних станах, які стаються на виробництві, у побуті, під час дорожньо-транспортних пригод, катастроф, техногенних аварій та при гострих неврологічних, терапевтичних, хірургічних та термінальних станах. Ненадання долікарської допомоги при нещасних випадках, раптових гострих захворюваннях людини призводить до тяжких наслідків, аж до летальних. Своєчасна допомога відіграє важливу роль у подальшому лікуванні потерпілих і хворих, сприяє скороченню термінів їх медичної та трудової реабілітації [34].

Під час розробки інформаційної системи програмний інженер постійно перебуває за персональним комп'ютером. Також при впровадженні такої системи на серверному обладнанні необхідно працювати з приладами які перебувають під напругою, це збільшує ймовірність ураження електричним струмом. Важливо в такі моменти знати алгоритм та послідовність кроків які необхідно виконати для надання першої медичної допомоги. Ураження електричним струмом можливе при безпосередньому контакті з провідником електричного струму, електричною дугою, напругою від лежачого на землі дроту, частіше високої напруги. Загалом електротравма становить 1-2 % усіх видів травм. Близько 80 % ураження електричним струмом виникає при контакті зі струмом низької напруги (до 1000 В), до 20 % - струмом високої напруги. У першому випадку летальність становить 3 %, при контакті зі струмом високої напруги – до 30 %. Побутовий струм у 50 Гц особливо небезпечний для нормального функціонування міокарду [18]. Ступінь тяжкості функціональних розладів та ушкоджень залежить також від фізичних характеристик струму, умов контакту та стану постраждалого до моменту отримання електротравми. Також варто враховувати шлях проходження струму в організмі, електропровідність шкіри, загальний стан організму, навколишнє

середовище (сухе, вологе). Так, суха, товста шкіра має опір в 10-200 кОм, тоді, як волога, тонка шкіра створює опір у 100-200 разів менше [18]. Розрізняють легку, середню й тяжку форми ураження електричним струмом. У разі легких уражень постраждалий непритомніє. Електротравми середнього ступеня викликають загальні судоми м'язів, непритомність, розлади дихання й діяльності серця. У разі тяжких уражень дихання та серцева діяльність постраждалого настільки пригнічені, що звичайними методами їх не вдасться відновити (він перебуває в стані клінічної смерті). Ознаки ураження електричним струмом. Втрату свідомості спостерігають у 70 % випадків, вона може тривати від кількох хвилин до години. Водночас в окремих випадках свідомість може не відновлюватися понад добу. Ознаками ураження головного мозку є клонічні і тонічні судоми, головний біль, сонливість, ретроградна амнезія. Рідше виникають вогнищеві ушкодження головного і спинного мозку. Всі ці явища часто спостерігають у випадках, коли місцем входу струму є голова або верхні кінцівки. Якщо на шляху проходження струму знаходиться головний мозок, серце, то наслідки ураження можуть бути фатальними – розвивається стан клінічної смерті, де у 80 % випадків спостерігають фібриляцію шлуночків, а у 20 % - асистолію [19]. Проходячи крізь живий організм, електричний струм чинить дію: - термічну - з опіками відповідних ділянок, нагріванням кровоносних судин, нервів; - електричну - з розкладанням крові та інших органічних речовин; - біологічну - з подразненням та збудливістю живих тканин організму, що супроводжується судомним скороченням м'язів, у тому числі м'язів серця і легень. Як наслідок виникають різні порушення в організмі і навіть повна зупинка роботи серця та легень. Електричний струм викликає місцеві й загальні порушення в організмі. Місцеві зміни проявляються опіками тканини в місцях виходу й входу електричного струму. Залежно від стану ураженого (волога шкіра, стомлення, виснаження тощо), сили й напруги струму можливі різні місцеві прояви - від утрати чутливості до глибоких опіків. За впливу змінного струму силою 15 мА у постраждалого виникають судоми (так званий струм, що не відпускає). У випадку враження струмом силою 25-50 мА настає зупинка дихання. Через спазм

голосових зв'язок постраждалий не може крикнути й покликати на допомогу. Якщо дія струму не припиняється, через кілька хвилин відбувається зупинка серця в результаті гіпоксії й настає смерть постраждалого [34].

Порядок надання першої медичної допомоги при ураженні електричним струмом:

- Першим кроком є забезпечення негайної безпеки рятувальника та потерпілого, від'єднавши джерело живлення або використовуючи ізольовані інструменти, щоб усунути людину від електричного контакту.

- Після того, як потерпілий перебуває в безпечному середовищі, слід провести первинну оцінку, щоб оцінити його життєві функції, чи він перебуває в свідомості та наявність будь-яких небезпечних для життя травм. Якщо потерпілий не реагує, не дихає або у нього відсутній пульс, слід негайно розпочати серцево-легеневу реанімацію (СЛР).

- Етапи, пов'язані з СЛР, включають забезпечення відкритих дихальних шляхів, для цього потрібно нахилити голови назад і підняття підборіддя, виконувати процес штучного дихання та компресії грудної клітки.

- Під час проведення серцево-легеневої реанімації важливо негайно викликати екстринну медичну допомогу або доручити це зробити комусь поблизу. Прибуття кваліфікованих медичних працівників є життєво важливим для подальшого обстеження та лікування потерпілого.

- Очікуючи прибуття екстреної медичної допомоги, важливо постійно контролювати життєво важливі функції потерпілого та заспокоювати його. Якщо жертва починає реагувати, переконайтеся, що вона зайняла зручне положення та запропонуйте їй емоційну підтримку.

При розробці веб-додатку для контролю за веденням документації в медичних закладах з використанням JavaScript технологій не можна не зважати на ризик ураження електричним струмом. Розуміння відповідних кроків для надання першої медичної допомоги у разі ураження електричним струмом має важливе значення для забезпечення безпеки та благополуччя осіб, які беруть участь у розгортанні та обслуговуванні системи.

### 3.2 Вимоги ергономіки до організації робочого місця лікаря

Проектування та розробка веб-додатку передбачає роботу з персональним комп'ютером, тому вкрай важливо правильно організувати робоче місце. Ергономічно оптимізована робоча зона відіграє важливу роль у сприянні здоров'ю, комфорту та продуктивності людей, які проводять тривалий час за комп'ютером. Робочі місця працівників, обладнані комп'ютерами пристроями, повинні відповідати вимогам ДСТУ 8604:2015 «Дизайн і ергономіка. Робоче місце під час виконання робіт стоячи» [34] та ДСанПІН 3.3.2.007-98 «Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електроннообчислювальних машин» [34].

Основні положення ДСТУ 8604:2015:

- Конструкція робочого місця та взаємне розташування всіх його елементів (сидіння, органів керування, засобів відображення інформації тощо) повинні відповідати антропометричним, фізіологічним і психологічним вимогам, а також характеру виконуваної роботи.

- Конструкцією робочого місця повинно бути забезпечено виконання трудових операцій у межах зони досяжності моторного поля.

- Висоти сидіння та підставки для ніг (у разі нерегульованої висоти робочої поверхні). У цьому разі висоту робочої поверхні встановлюють за номограмою для працюючих зростом 1800 мм. Оптимальна робоча поза для менших за зростом працюючих досягається збільшенням висоти робочого сидіння й підставки для ніг на величину, що дорівнює різниці між висотою робочої поверхні для працюючого зростом 1800 мм і висотою робочої поверхні, оптимальної для зросту даного працюючого.

- Мінімальна товщина стільниці залежатиме від характеристик міцності використовуваного матеріалу та інших технічних вимог і має бути не більше ніж 30 мм.

- Підставка для ніг повинна бути регульованою по висоті. Її ширина повинна бути не менше ніж 300 мм, довжина - не менше ніж 400 мм. Поверхня підставки повинна бути рифленою. По передньому краю доцільно передбачати бортик висотою 10 мм. Основні положення ДСанПІН 3.3.2.007-98: - Площа на одне робоче місце має становити не менше ніж 6,0 м<sup>2</sup>, а об'єм не менше ніж 20,0 м<sup>3</sup>.

- Приміщення для роботи повинні мати природне та штучне освітлення. - Природне освітлення має здійснюватись через світлові прорізи, орієнтовані переважно на північ чи північний схід і забезпечувати коефіцієнт природною освітленості (КПО) не нижче ніж 1,5%.

- Для внутрішнього оздоблення приміщень слід використовувати дифузовідбивні матеріали з коефіцієнтами відбиття для стелі 0,7-0,8, для стін 0,5-0,6.

- Яскравість світильників загального освітлення в зоні кутів випромінювання від 50 до 90 градусів з вертикаллю в повздовжній та поперечній площинах має становити не більше ніж 200 кд/м<sup>2</sup>, захисний кут світильників

- не менше ніж 40 градусів

- Показник засліплення у разі використання джерел загального штучного освітлення у виробничих приміщеннях має не перевищувати 20.

- Необхідно обмежувати нерівномірність розподілу яскравості в полі зору працюючих з екранами. При цьому співвідношення яскравості робочих поверхонь має бути не більшим ніж 3:1, а співвідношення яскравості робочих поверхонь та поверхонь стін, обладнання тощо - 5:1.

## ВИСНОВОК

Коли ви подорожуєте Інтернетом сьогодні, ви знайдете багато дуже добре розроблених сайтів, які швидко завантажуються, виглядають привабливо, мають чітко визначені інформаційні розділи та зручні функції навігації. Такі сайти просто запрошують вас відвідати їх, відпочити, зупинитися на деякий час, повеселитися та дізнатися про щось нове.

Якщо хтось довгий час залишається на сайті і навіть не помічає цього, цей сайт здається дуже добре продуманим.

Багато веб-майстрів не витрачають час на обдумування потоку інформації та обмежуються лише розміщенням тексту та зображень на сторінці. Хоча багато сайтів можуть похвалитися тим, що у них набагато більше відвідувачів, ніж у конкурентів, це не кращий показник якості роботи. Найкращим показником є час: якщо користувачі залишаються на сайті досить довго, щоб пройти через різні рівні та розділи, можна бути впевненим у тому, що робота виконана добре.

Рітейлери виявляють творчий підхід і знаходять усілякі способи впровадження ІІІ у свій бізнес. Це може бути як пряма взаємодія з покупцями, так і опосередкований вплив на них, уточнення запасів та контроль за їх станом на розподілених майданчиках, а також детальніше вивчення керівниками поточних показників та прогнозів на майбутнє.

При продуманому впровадженні ІІІ та наявності відповідних політик та обмежень його використання він може принести користь як співробітникам, так і покупцям завдяки автоматизації, персоналізації та «безконтактним» функціям, які покращують загальне враження від роздрібної торгівлі.

У цій роботі розглянуті актуальні питання розробки та створення сучасного веб-додатку із використанням ІІІ.

При цьому мною було вирішено такі окремі завдання:

- ознайомлення з сучасними Інтернет-технологіями та використовувати їх у своїй розробці;

– вивчення основних понять мов програмування PHP та Javascript, СУБД MySQL та фреймворку Laravel, які використовують при розробленні веб-застосунків;

– ознайомлення з методами та способами представлення на веб-сторінках різних видів інформації (текстів та зображень);

– ознайомлення з основними правилами та рекомендаціями щодо розробки та створення веб-додатків та неухильне дотримання їх у своїй практиці;

- Визначення структури веб-сторінок;

- Надання покрокової стратегії розробки інтернет магазину.

У ході виконання роботи було детально вивчено фреймворк Laravel та його доповнення.

Основним етапом у вивченні фреймворку стало створення на його основі повноцінного інтернет магазину. Завдання щодо його виконання також були повністю виконані. Створено веб-додаток, який відповідає всім сучасним нормам. Він поєднує в собі зручність, функціональність та комфортний для користувача зовнішній вигляд. На сайті є можливість пошуку товару за каталогом та категоріями, додавання товару в кошик та оформлення замовлення. Додатком можна легко керувати і надалі, що робить його зручним для адміністрації.

Внаслідок проведених робіт на базі обраних технологій було створено веб-каталог магазину «Оптика».

## СПИСОК ЛІТЕРАТУРИ

1. Radha Guha Toward the Intelligent Web Systems // First International Conference on Computational Intelligence, Communication Systems and Networks. – 2009. с. 459 – 463.
2. Keshab Nath What Comes after Web 3.0? Web 4.0 and the Future [Electronic resource]: International Conference on Computing and Communication Systems (I3CS'15). 2015. URL: [https://www.researchgate.net/profile/Keshab\\_Nath2/publication/281455061\\_What\\_Comes\\_after\\_Web\\_30\\_Web\\_40\\_and\\_the\\_Future/links/55e8b3eb08ae3e1218425040.pdf](https://www.researchgate.net/profile/Keshab_Nath2/publication/281455061_What_Comes_after_Web_30_Web_40_and_the_Future/links/55e8b3eb08ae3e1218425040.pdf)
3. Sareh Aghaei, Mohammad Ali Nematbakhsh, Hadi Khosravi Farsani Evolution of the world wide web: from web 1.0 to web 4.0 [Electronic resource]: International Journal of Web & Semantic Technology. 2012. Vol. 3. №1. с. 110. URL: <https://pdfs.semanticscholar.org/8cb3/93c3229e8f288febfa4dac12a0f6298efb93.pdf>
4. Nupur Choudhury World Wide Web and Its Journey from Web 1.0 to Web 4.0 [Electronic resource]: International Journal of Computer Science and Information Technologies. 2014. Vol. 5(6). с. 8096 – 8100. URL: <http://ijcsit.com/docs/Volume%205/vol5issue06/ijcsit20140506265.pdf>
5. Nath K., Dhar Mr. S., Basishta S. Web 1.0 to Web 3.0 – Evolution of the Web and its Various Challenges [Electronic resource]: International Conference on Optimization, Reliability, and Information Technology (ICROIT). 2014. URL: [https://www.researchgate.net/publication/269310255\\_Web\\_10\\_to\\_Web\\_30\\_-\\_Evolution\\_of\\_the\\_Web\\_and\\_its\\_various\\_challenges](https://www.researchgate.net/publication/269310255_Web_10_to_Web_30_-_Evolution_of_the_Web_and_its_various_challenges)
6. Fernando Almeida Concept and Dimensions of Web 4.0 [Electronic resource]: Concept and Dimensions International journal of Computers and Technology. 2014. Vol. 16. №7. с. 7040 – 7046. URL: <https://pdfs.semanticscholar.org/d38c/0b90e2501a34c8d8da9f985b079dea778d6b.pdf>
7. Kosmayer D. How to build a highly converting website with Artificial Intelligence [Electronic resource]: Crank wheel. 2017. URL:

<https://crankwheel.com/how-to-build-a-highly-converting-website-with-artificialintelligence/>

8. Constandse C. How AI-driven website builders will change the digital landscape [Electronic resource]: Medium. 2018. URL: <https://uxdesign.cc/how-aidriven-website-builders-will-change-the-digital-landscape-a5535c17bbe>

9. Ismail K. The Grid AI-Powered Website Builder Doesn't Check Enough Boxes [Electronic resource]: CMS WiRE. 2017 URL: <https://www.cmswire.com/digitalexperience/the-grid-ai-powered-website-builder-doesnt-check-enough-boxes/>

10.13. Reed L. Will AI Website Builders Ever Replace Human Web Designers?[Electronic resource]: Dzone. 2018. URL: <https://dzone.com/articles/will-ai-websitebuilders-ever-replace-human-web-de>

11. Vukcevic P. A Look at AI Website Builders: The Future or Just a Gimmick [Electronic resource]: COMPARE THE CLOUD LTD. 2017. URL: <https://www.comparethecloud.net/articles/ai/look-ai-website-builders-future-justgimmick/>

12. Agarwal D. An Introduction to AI at LinkedIn [Electronic resource]: LinkedIn Engineering.2018. URL: <https://engineering.linkedin.com/blog/2018/10/anintroduction-to-ai-at-linkedin>

13. Brandon J. This A.I.-Powered Website Creator Stood Up to Rigorous Testing [Electronic resource]: INC. 2018. URL: <https://www.inc.com/johnbrandon/reviewed-this-ai-powered-website-creator-stood-up-to-rigorous-testing.html>

14. Зубик Л. В., Зубик Я. Я. Розвиток систем штучного інтелекту // Вісник Чернігівського національного педагогічного університету. Серія: Педагогічні науки. 2018. Вип. 153. с. 49 – 52. URL: [http://nbuv.gov.ua/UJRN/VchdpuP\\_2018\\_153\\_13](http://nbuv.gov.ua/UJRN/VchdpuP_2018_153_13)

15. Tony Beltramelli Pix2code: Generating Code from a Graphical User Interface Screenshot [Electronic resource]: arXiv:1705.07962v2 [cs.LG]. 19 Sep, 2017. URL: <https://arxiv.org/pdf/1705.07962.pdf>

16. Івченко К. В. Машинне навчання та когнітивні обчислення [Електронний ресурс]: Матеріали міжнародної науково-практичної інтернет-конференції "Використання інноваційних технологій в процесі підготовки фахівців", Вінниця, 28-29 березня 2017 р. 2017. URL: <https://conferences.vntu.edu.ua/index.php/itpf/2017/paper/view/3285>
17. Bala Venkatesha An Introduction to Machine Learning Theory [Електронний ресурс]: Medium, 27 April, 2018. URL: <https://medium.com/@venkateshpnk22/an-introduction-to-machine-learning-theory-2b09a4748187>
18. Орел А.А., Ромакіна О.М. Навчальний посібник з курсу «Проектування бізнес-процесів» для студентів механіко-математичного факультету, 2007. – 123 с.
19. А.В. Варзунов, Є.К. Торосян, Л.П. Сажніва. Аналіз та управління бізнес-процесами. Навчальний посібник, 2010 – 212 с.
20. Лайза Кріспін, Джанет Грегорі. Гнучке тестування: практичний посібник для тестувальників програмного забезпечення та гнучких команд Agile Testing: A Practical Guide for Testers and Agile Teams. - К, 2010. - 464 с.
21. Назаренко Г. І., Гулієв Я. І., Єрмаков Д. Є. Медичні інформаційні системи: теорія та практика. - К, 2015. - 320 с.
22. Шеєр А.В. Моделювання бізнес-процесів. – ІХ., 2000. - 224 с.
23. Буч Р, Рамбо Дж. Введення в UML від творців мови. – 2015. – 493 с.
24. Карл І. Вігерс. Розробка вимог до програмного забезпечення. - 2005. - 576 с.
25. Соммервілл І. Інженерія програмного забезпечення. 2005. - 624 с.
26. Вендер А.М. CASE технології. Сучасні методи та засоби проектування інформаційних систем М.: Фінанси та статистика, 2005. - 352 с.
27. Ульман, Дж. Основи систем баз даних/Дж. Ульман. -, 2017. - 292 с.
28. Методичні вказівки до виконання дипломної роботи освітнього рівня —бакалавр студентами усіх форм навчання для напряму підготовки 121 — Інженерія програмного забезпечення / Укладачі : Петрик М.Р., Михалик Д.М.,

Кінах Я.І., Гладь С.В., Цуприк Г.Б. – Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2016 – 28 с.

29. М.Р. Петрик, Д.М. Михалик, О.Ю. Петрик, Г.Б. Цуприк. Методичні вказівки до виконання атестаційної роботи магістра за спеціальністю 121 – “Інженерія програмного забезпечення” для усіх форм навчання [Текст] – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя – 2020 – 27 с.

30. Бойко І.В., М.Р. Петрик, Г.Б. Цуприк. Інформаційні технології видобутку даних (Data mining, високопродуктивні обчислення у складних системах): навчальний посібник. Тернопіль: : ТНТУ 2020 – 62 с.

31. Бойко І.В., М.Р. Петрик, Г.Б. Цуприк. Моделювання та видобуток даних (високопродуктивні обчислення у великих алгебраїчних та числових системах, комбінаторному аналізі): навчальний посібник. Тернопіль: : ТНТУ 2019 – 62 с.

32. Нога В.В., Цуприк Г.Б. Інформаційні технології при створенні автоматизованих систем для задач збору та збереження інформації. Збірник тез доповідей VII Міжнародної науково-технічної конференції молодих учених та студентів. Актуальні задачі сучасних технологій – Тернопіль 28-29 листопада 2018. — Т. : ТНТУ, 2018. — Том 2. — С. 131.

33. І.В. Бойко, М.Р. Петрик, Г.Б. Цуприк. Дискретні структури (Алгебраїчні та числові системи, комбінаторний аналіз). Навчально-методичний посібник для студентів спеціальності 121 «Інженерія програмного забезпечення», аспірантів та викладачів вищих навчальних закладів. Конспект лекцій : Навчальний посібник / І.В. Бойко, М.Р. Петрик, Г.Б. Цуприк – Тернопіль : ТНТУ , 2017. – 64 с.

34. Стручок В.С. Безпека в надзвичайних ситуаціях. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної бо та заочної (дистанційної) форм навчання / В.С.Стручок. — Тернопіль: ФОП Паляниця В. А., 2022. — 156 с.

## ДОДАТКИ

## ДОДАТОК А

### Архітектура програмної системи

#### A.1 Патерн Model-View-Presenter

Модель–вигляд–контролер (або Модель–представлення–контролер, англ. Model-view-controller, MVC) – архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення.

Модель складається з чистої логіки програми, яка взаємодіє з базою даних. Вона включає в себе всю інформацію для подання даних кінцевому користувачеві.

```
# model_view_controller.py
import basic_backend
import mvc_exceptions as mvc_exc

class ModelBasic(object):
    def __init__(self, application_items):
        self._item_type = 'product'
        self.create_items(application_items)
    @property
    def item_type(self):
        return self._item_type
    @item_type.setter
    def item_type(self, new_item_type):
        self._item_type = new_item_type
    def create_item(self, name, price, quantity):
        basic_backend.create_item(name, price,
quantity)
    def create_items(self, items):
        basic_backend.create_items(items)
    def read_item(self, name):
        return basic_backend.read_item(name)
    def read_items(self):
        return basic_backend.read_items()
    def update_item(self, name, price, quantity):
        basic_backend.update_item(name, price,
```

Представлення відповідає за відображення інформації на зовнішньому інтерфейсі. Інтерфейс – це те, що бачить користувач, і він може бути реалізований за допомогою веб-сторінки (в разі веб-додатків), або набору інструментів з графічним інтерфейсом.

```
# model_view_controller.py
class View(object):
    @staticmethod
    def show_bullet_point_list(item_type, items):
        print('---          {}          LIST          ---')
        print(''.format(item_type.upper()))
        for item in items:
            print('* {}'.format(item))
    @staticmethod
    def show_number_point_list(item_type, items):
        print('---          {}          LIST          ---')
        print(''.format(item_type.upper()))
        for i, item in enumerate(items):
            print('{} . {}'.format(i+1, item))
    ...
    @staticmethod
    def display_change_item_type(older, newer):
        print('-----')
        print('Change item type from "{}" to')
        print('{}' .format(older, newer))
        print('-----')
    @staticmethod
    def display_item_updated(item, o_price, o_quantity,
n_price, n_quantity):
        print('-----')
        print('Change {} price: {} --> {}'.format(item, o_price, n_price))
        print('Change {} quantity: {} --> {}'.format(item, o_quantity, n_quantity))
        print('-----')
    @staticmethod
    def display_item_deletion(name):
        print('-----')
        print('We have just removed {} from our')
        print('list' .format(name))
        print('-----')
```

Контролер приймає вхідні дані користувача і делегує уявлення даних в уявлення і обробку даних в модель.

```

        def show_item(self, item_name):
            try:
                item = self.model.read_item(item_name)
                item_type = self.model.item_type
                self.view.show_item(item_type, item_name,
item)
            except mvc_exc.ItemNotStored as e:

self.view.display_missing_item_error(item_name, e)
...

        def update_item_type(self, new_item_type):
            old_item_type = self.model.item_type
            self.model.item_type = new_item_type

self.view.display_change_item_type(old_item_type,
new_item_type)

        def delete_item(self, name):
            item_type = self.model.item_type
            try:
                self.model.delete_item(name)
                self.view.display_item_deletion(name)
            except mvc_exc.ItemNotStored as e:

self.view.display_item_not_yet_stored_error(name,
item_type, e)

# model_view_controller.py
class Controller(object):

    def __init__(self, model, view):
        self.model = model
        self.view = view

    def show_items(self, bullet_points=False):
        items = self.model.read_items()
        item_type = self.model.item_type
        if bullet_points:

self.view.show_bullet_point_list(item_type, items)
        else:

self.view.show_number_point_list(item_type, items)

```

Оскільки модель, перегляд і контролер відокремлені, кожен з трьох може бути розширений, змінений і замінений без необхідності переписувати два інших.

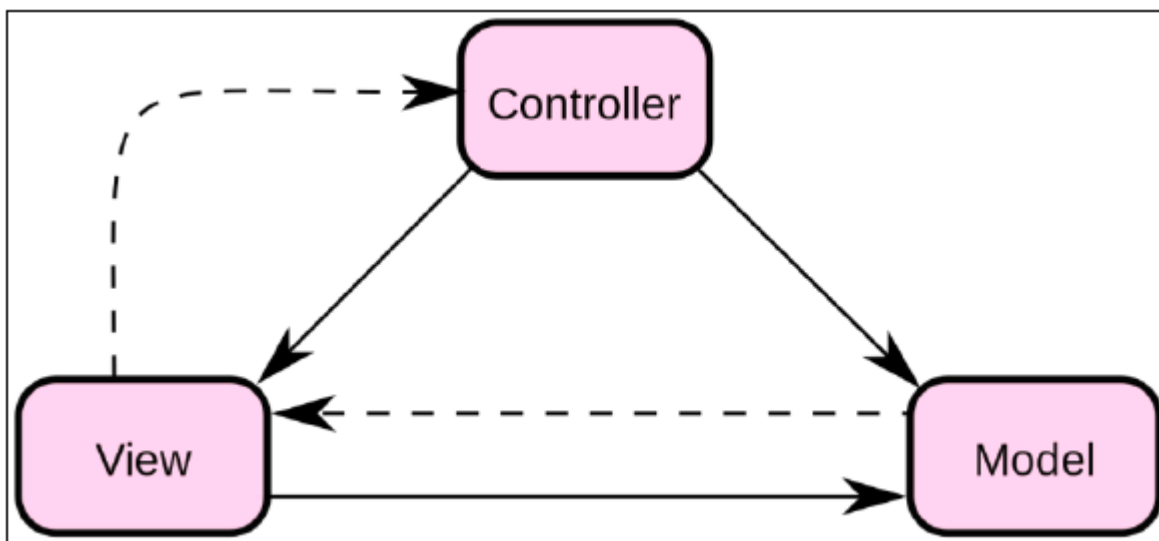


Рисунок А.1 – Схема патерну MVC при використанні в Python-додатках

Додаток  
Диск з роботою