

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

_____ Ігор ШЕЛЕХОВ
(підпис)

червня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня бакалавр

зі спеціальності 122 – Комп'ютерних наук,
освітньо-професійної програми «Інформатика»
на тему: «Веб-орієнтована система для забезпечення інформаційного
обслуговування діяльності магазину одягу»
здобувача групи ІН-91 Литвиненка Дмитра Володимировича

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело.

_____ Дмитро ЛИТВИНЕНКО
(підпис)

Керівник,
асистентка кафедри комп'ютерних наук,
кандидат фізико-математичних наук

Ольга ШУТИЛЄВА

_____ (підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

(підпис)

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавра

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»
здобувача групи ІН-91 Литвиненка Дмитра Володимировича

1. Тема роботи: «Веб-орієнтована система для забезпечення інформаційного обслуговування діяльності магазину одягу»

затверджую наказом по СумДУ від «01» червня 2023 р. № 0475-VI

2. Термін здачі здобувачем кваліфікаційної роботи до 09 червня 2023 року

3. Вхідні дані до кваліфікаційної роботи

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1) Аналіз проблеми предметної області, постановка й формування завдань дослідження.

2) Огляд технологій, що використовуються для розробки web-додатку. 3) Розробка web-додатку. 4) Аналіз результатів

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

6. Консультанти до проекту (роботи), із зазначенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання «8» лютого 2022 р.

Завдання прийняв до виконання _____

(підпис)

Керівник _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	Аналіз проблеми предметної області, постановка й формування завдань дослідження		
2	Аналіз аналогів web-додатків		
3	Проведення етапу моделювання		
4	Розробка структури web-додатку		

5	Оформлення пояснювальної записки до кваліфікаційної роботи		
---	--	--	--

Здобувач вищої освіти

(підпис)

Керівник

(підпис)

АНОТАЦІЯ

Записка: пояснювальна записка складається зі вступу, розділів, висновків, списку використаних джерел із 20 найменувань, трьох додатків. Загальний обсяг роботи складає 121 сторінки, у тому числі 45 сторінки основного тексту, 2 сторінки списку використаних джерел, 67 сторінок додатків.

Об'єкт дослідження: web-орієнтована система для забезпечення інформаційного обслуговування діяльності магазину одягу.

Мета роботи: розробка web-додатку для підтримки роботи магазину з продаж одягу.

Методи дослідження: аналіз аналогів web-додатків, аналіз інструментів для реалізації web-додатку.

Результат роботи: розроблений web-додаток «Clothes Shop», який надає всі необхідні інструменти для підтримки продажу одягу.

Ключові слова: ІНТЕРНЕТ МАГАЗИН, HTML, CSS, BOOSTRAP, JAVASCRIPT, JQUERY, AJAX, PHP.

ЗМІСТ

ВСТУП	Ошибка! Закладка не определена.
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	8
1.1 Огляд останніх досліджень та публікацій.....	Ошибка! Закладка не определена.
1.2 Аналіз існуючих додатків-аналогів	9
1.3 Постановка задачі	15
1.4 Визначення середовища розробки	16
2 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ВЕБ-ДОДАТКУ	18
2.1 Проєктування веб-додатку	18
2.2 Реалізація бази даних.....	21
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ.....	23
3.1 Архітектура веб-додатку	23
3.2 Розробка користувацької частини веб-додатку	33
ВИСНОВКИ.....	51
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	52
ДОДАТОК А.....	54
ДОДАТОК Б	60
ДОДАТОК В.....	76

ВСТУП

Актуальність. Розвиток сучасних технологій і зростання популярності електронної комерції вимагають від магазинів одягу адаптуватися до нових вимог і надавати ефективне інформаційне обслуговування своїм клієнтам. Веб-орієнтована система для забезпечення інформаційного обслуговування стає актуальною, оскільки вона надає можливість магазинам одягу покращити свою присутність в Інтернеті, залучити нових клієнтів та підвищити рівень задоволення існуючих клієнтів.

Об'єкт дослідження. Об'єктом дослідження є процес інформаційного обслуговування діяльності магазину одягу, зокрема, його онлайн-присутність, електронна комерція, зберігання і обробка даних про товари та клієнтів, забезпечення зручного інтерфейсу для користувачів та ефективного управління бізнес-процесами.

Предмет дослідження. Предметом дослідження є розробка веб-орієнтованої системи для забезпечення інформаційного обслуговування діяльності магазину одягу. Це включає аналіз вимог користувачів, проектування та реалізацію системи, інтеграцію з існуючими інформаційними системами, тестування, впровадження та підтримку системи.

Гіпотеза. Гіпотезою даного дослідження є те, що розробка веб-орієнтованої системи для забезпечення інформаційного обслуговування діяльності магазину одягу сприятиме покращенню ефективності продажів, збільшенню клієнтської бази та покращенню задоволення клієнтів.

Новизна. Новизною даної роботи є розробка веб-орієнтованої системи, спеціально адаптованої для потреб магазинів одягу. Вона враховує особливості даної галузі, забезпечує зручний інтерфейс для користувачів, ефективне управління товарами та замовленнями, а також надає засоби аналізу та звітності.

Структура. Дана робота складається зі вступу, аналітичного огляду, постановки задачі, вибору методу розв'язання поставленої задачі, опису

програмного забезпечення інформаційної системи, висновків, списку використаних джерел та додатків. Вступ містить обґрунтування актуальності теми, визначає об'єкт, предмет, гіпотезу та структуру роботи. Аналітичний огляд включає огляд існуючих рішень у сфері веб-орієнтованих систем для магазинів одягу. Постановка задачі визначає конкретні цілі та завдання дослідження. Опис програмного забезпечення включає опис функціональності та особливостей розробленої системи. Висновки містять підсумки проведеного дослідження та висновки щодо досягнення поставлених цілей.

1. АНАЛІТИЧНИЙ ОГЛЯД

1.1 Сучасний стан індустрії web-додатків для комерційного бізнесу

Сьогодні онлайн-покупки є одним із найпопулярніших видів діяльності в Інтернеті. У 2017 році обсяг продажів електронної комерції в усьому світі склав \$2,3 трлн, і, за прогнозами, до 2023 року він зросте до \$6,88 трлн. Покупки онлайн мають багато переваг і вигод, не кажучи вже про більший вибір, і все більше людей починають повідомляти про те, що віддають їм перевагу перед традиційними фізичними магазинами.

У сучасному світі спостерігається тенденція, коли люди зручно здійснюють покупки без виходу зі своїх домівок, використовуючи послуги різноманітних компаній, які функціонують на глобальному рівні. Цей процес дозволяє споживачам залишатись у комфортних умовах власного житла, одночасно забезпечуючи можливість придбання широкого спектру товарів і послуг через Інтернет.

У даний час не є рідкістю факт, коли люди використовують онлайн-платформи для замовлення різних товарів, починаючи від одягу від іноземних брендів до продуктів харчування. Цей тренд значно полегшує процес покупок, оскільки споживачам більше не потрібно фізично відвідувати різні магазини або місцеві ринки. Замість цього, вони можуть замовити все необхідне через Інтернет і мати його доставлене безпосередньо до своїх дверей. Такий спосіб покупок є зручним та ефективним для споживачів, дозволяючи їм економити час та зусилля, а також мати доступ до більш широкого асортименту товарів, який пропонується на міжнародному ринку.

Завдяки появі інтернету та можливості здійснювати покупки онлайн, процес прийняття рішень покупцями швидко змінився. Покупці зараз краще поінформовані, ніж будь-коли раніше, і є завзятими дослідниками, коли справа доходить до речей, які вони хочуть придбати. Сьогодні не рідкість, коли

споживач придивляється до товару в магазині, а потім шукає в інтернеті кращу пропозицію від інтернет-магазину.[2]

Тому, розробка веб-додатку для підтримки роботи онлайн-магазину для продажів одягу, який би задовольняв потреби як для користувача так і для власника сайту та створював умови для збільшення прибутку підприємства, є актуальною задачею для сучасного бізнесу.

1.2 Аналіз існуючих додатків-аналогів

На сьогоднішній день, веб-додатки для підтримки роботи магазину продажів одягу користуються високою популярністю серед середньо-статистичних користувачів мережі інтернет. Це змінило ринок надавання послуг і тепер реалізація даного додатку потребує детального аналізу ринку конкурентів.

Отже, існує кілька факторів, які визначають успішність веб-додатку на ринку. Першим з них є адаптивність додатку до всіх сучасних браузерів, які доступні на ринку у 2023 році, а також можливість переглядати додаток на мобільних пристроях. Це дозволяє користувачам зручно отримувати доступ до додатку з будь-якого пристрою, що робить його більш доступним та підходящим для різних типів користувачів.

Другим важливим фактором є створення максимально зручного інтерфейсу, який дозволить користувачам легко навігуватись у веб-додатку. Простота використання та логічна організація функцій допоможуть забезпечити зручну та позитивну взаємодію з користувачем.

Крім того, вимоги до веб-додатку повинні включати функціонал, який допоможе здійснити замовлення та переглянути результати сесії на сайті. Зручний процес оформлення замовлення та наявність інструментів для перегляду результатів дозволять користувачам ефективно використовувати веб-додаток і отримувати бажані результати. За результатами пошуку схожих веб-додатків для аналізу було обрано наступні сайти:

- 1) garne.com.ua; [3]
- 2) cultboutique.com.ua; [4]
- 3) kasta.ua; [5]

Отже, розглянемо більш детально кожний із ресурсів.

1.2.1 Веб-додаток garne.com.ua

На Рисунку 1.1 представлений дизайн головної сторінки веб-додатку «garne.com.ua». Даний веб-додаток має сучасний та адаптивний дизайн, а також повний функціонал, необхідний для оформлення замовлень, як показано на Рисунку 1.2. Кольорова гамма була вдало підібрана разом з відповідними шрифтами. Крім того, наявність особистої сторінки надає можливість користувачам відстежувати історію своїх замовлень і зв'язуватися з адміністрацією сайту у разі виникнення питань (рис.1.3).

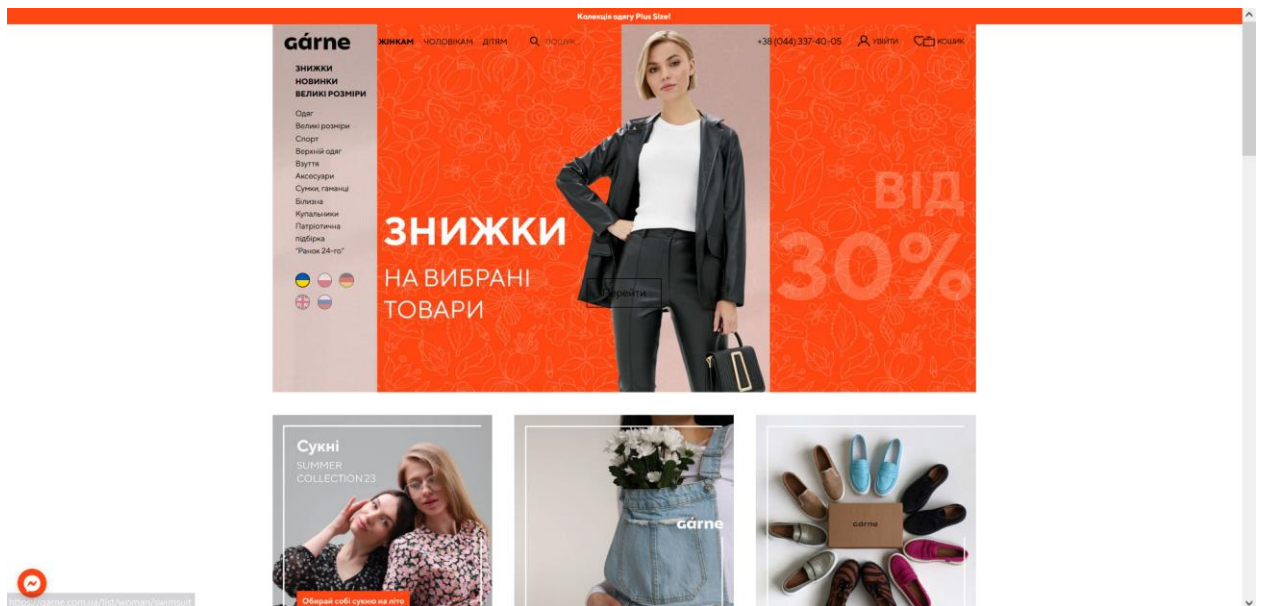


Рисунок 1.1 – Вигляд головної сторінки сайту «garne.com.ua»

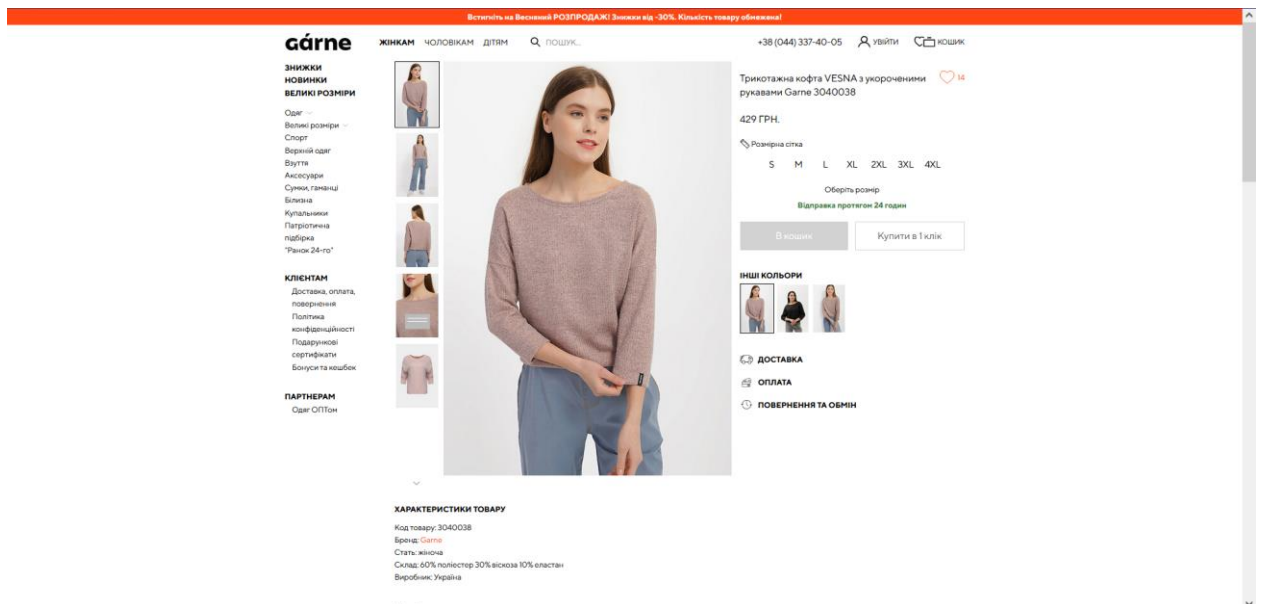


Рисунок 1.2 – Вигляд сторінки з обраним товаром

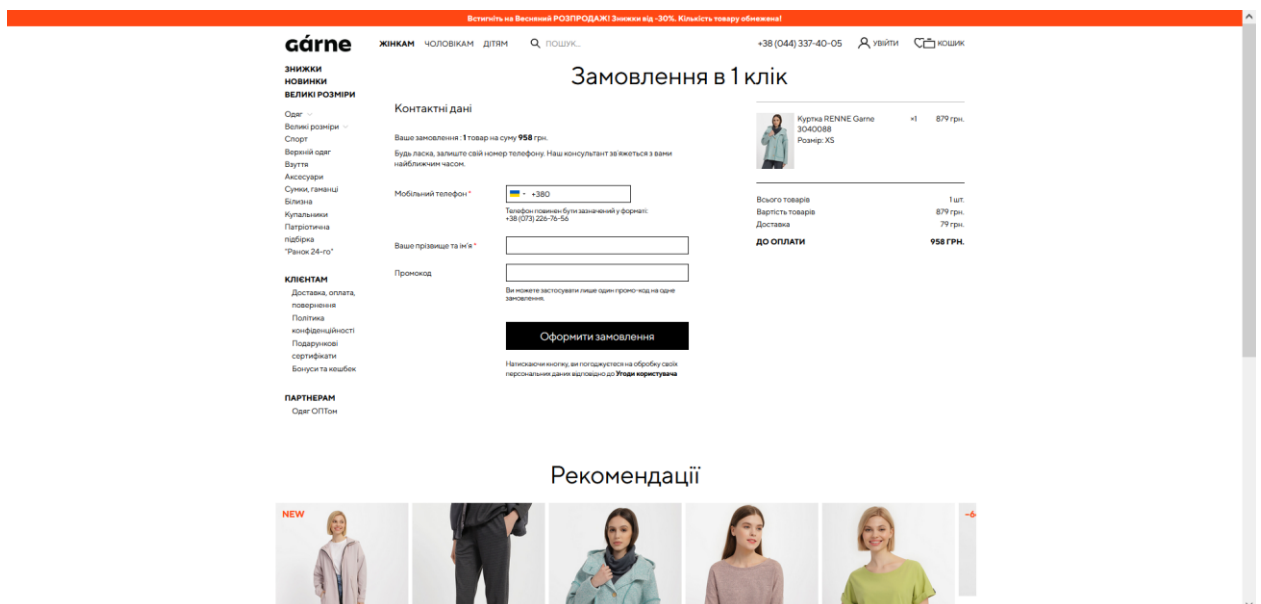


Рисунок 1.3 – Вигляд сторінки замовлення товару

1.2.2 Веб-додаток cultboutique.com.ua

Веб-додаток «cultboutique.com.ua» має схожі характеристики, які були описані при аналізі сайту «garne.com.ua». Однак, основна відмінність полягає у тому, що на головній сторінці веб-додатку представлені товари лише від одного бренду, як показано на Рисунку 1.4. Крім того, магазин акцентує увагу на товарах, які мають актуальність або відносяться до трендових категорій, що підкреслюється на Рисунку 1.5.

Процес замовлення товару веб-додатку не потребує додаткових дій з боку користувача, достатньо заповнити форму, навіть без обов'язкової реєстрації, як показано на Рисунку 1.6.

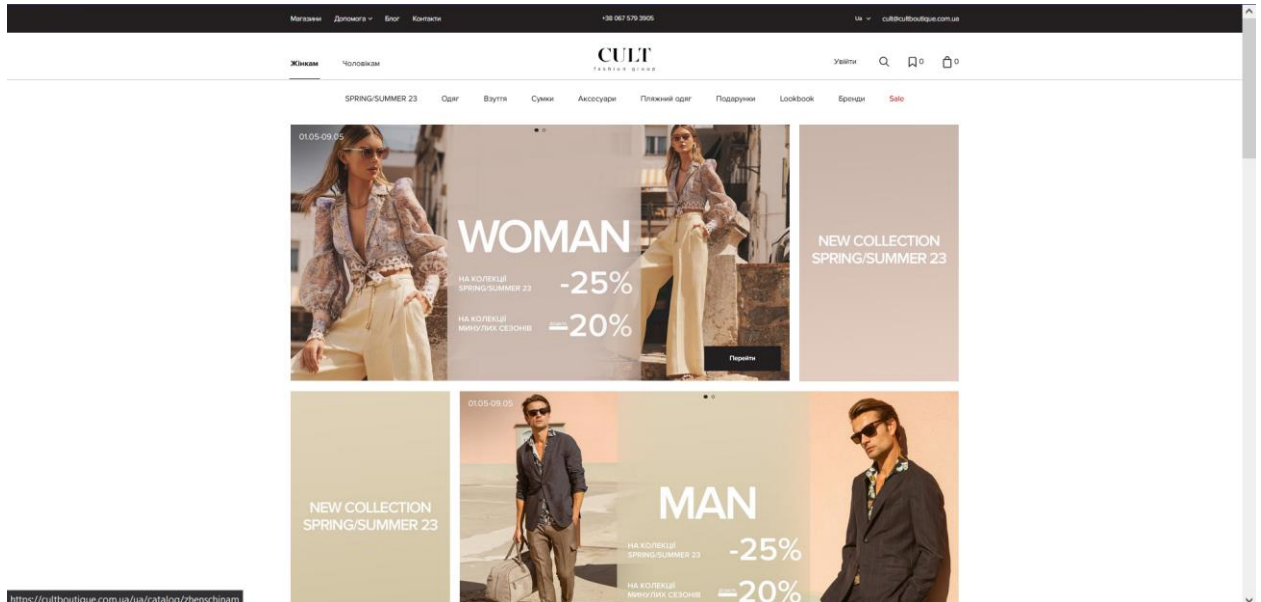


Рисунок 1.4 – Вигляд головної сторінки сайту «cultboutique.com.ua»

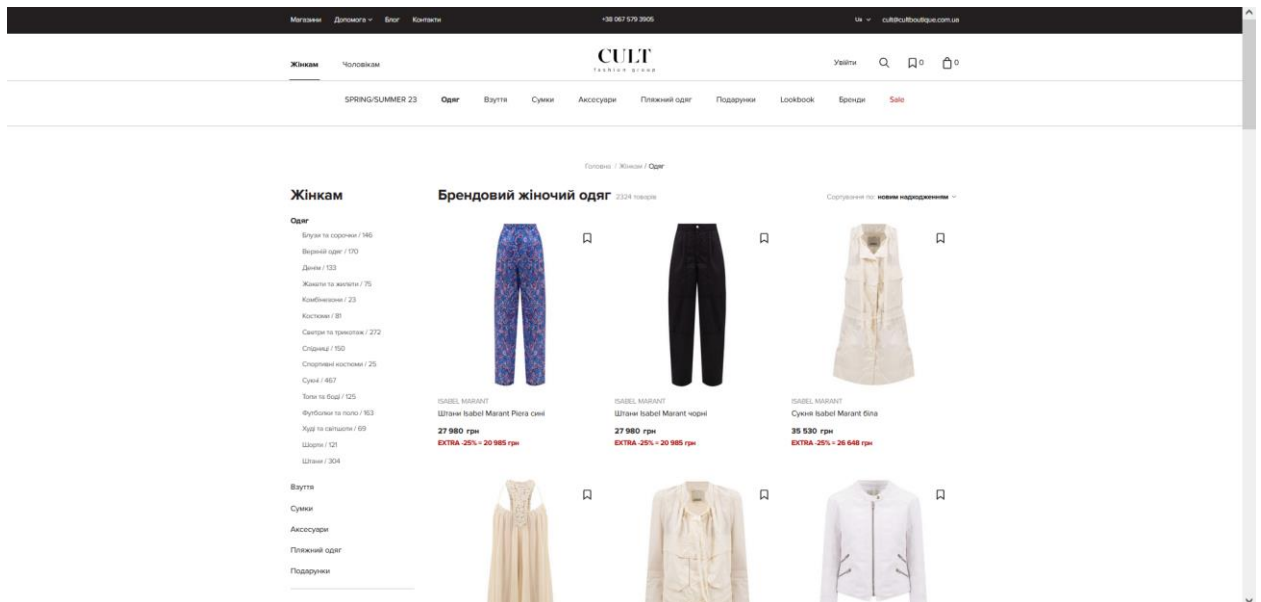


Рисунок 1.5 – Вигляд сторінки з товарами

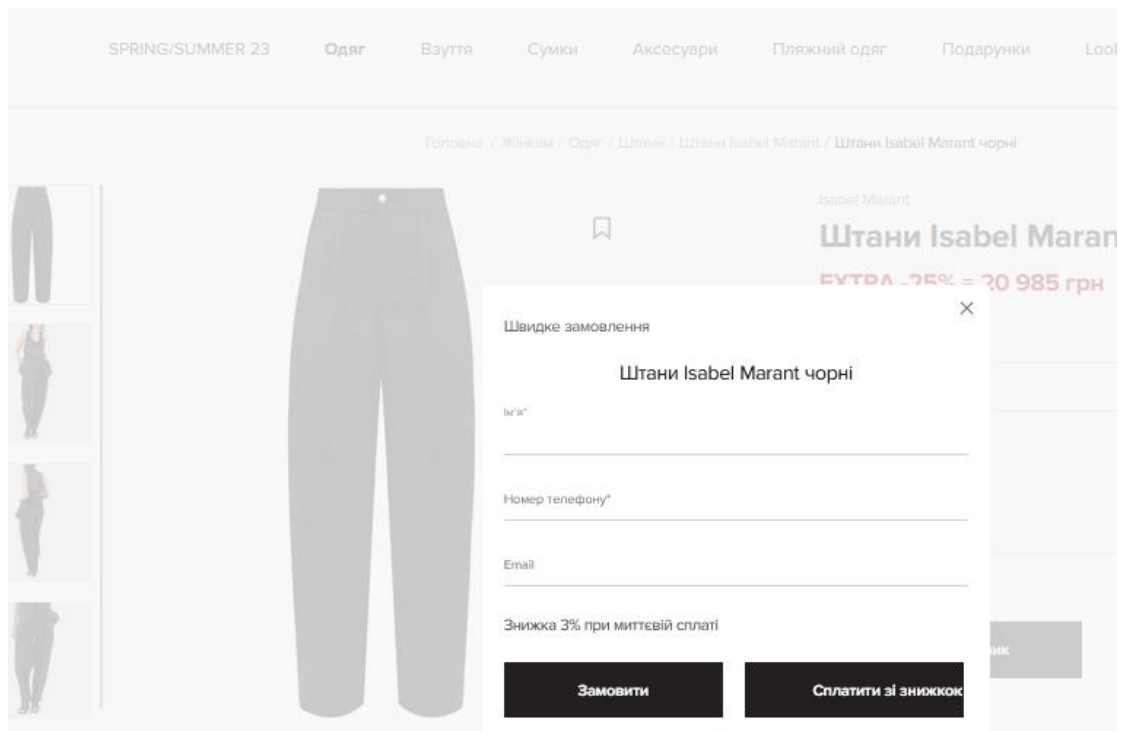


Рисунок 1.6 – Приклад швидкого замовлення товару

1.2.3 Веб-додаток kasta.ua

Даний веб-додаток має ряд проблем, які варто врахувати. Головна сторінка додатку перенасичена інформацією і спочатку не пропонує основну інформацію про роботу додатку, а замість цього одразу відображає товари, як показано на Рисунку 1.7. Це може викликати плутанину та збивати увагу користувача, коли вони намагаються ознайомитися з функціоналом додатку.

Крім того, головна сторінка містить два меню – горизонтальне і вертикальне, що також може викликати збиття уваги користувача, коли вони спробують вивчити веб-додаток (рис.1.7).

У веб-додатку відсутня опція швидкого замовлення товару, що означає, що обов'язкова реєстрація в системі є необхідною для здійснення покупки, як видно на Рисунку 1.8. Це може створювати перешкоди для користувачів, які хочуть здійснити швидку покупку без витрати часу на реєстрацію.

Крім того, веб-додаток також не має адаптивного дизайну, що є значним недоліком. Адаптивний дизайн є важливим для забезпечення коректного відображення і зручного використання додатку на різних пристроях,

включаючи мобільні пристрої. Ці питання варто врахувати та внести відповідні зміни в веб-додаток для поліпшення його користувацького досвіду.

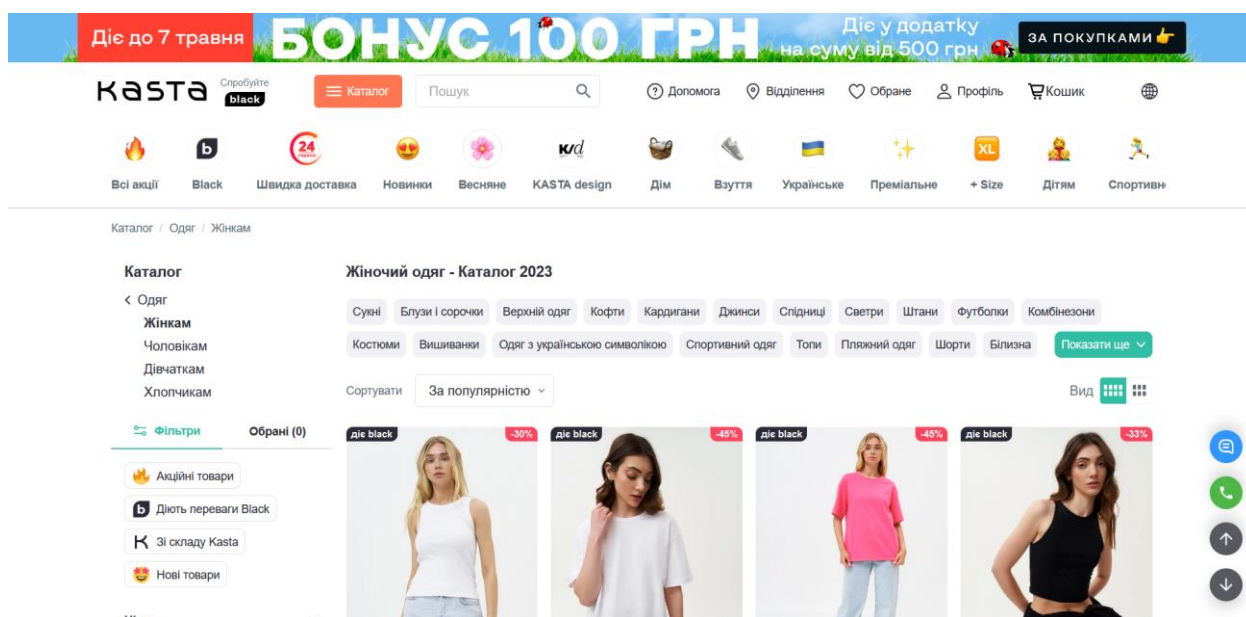


Рисунок 1.7 – Вигляд головної сторінки

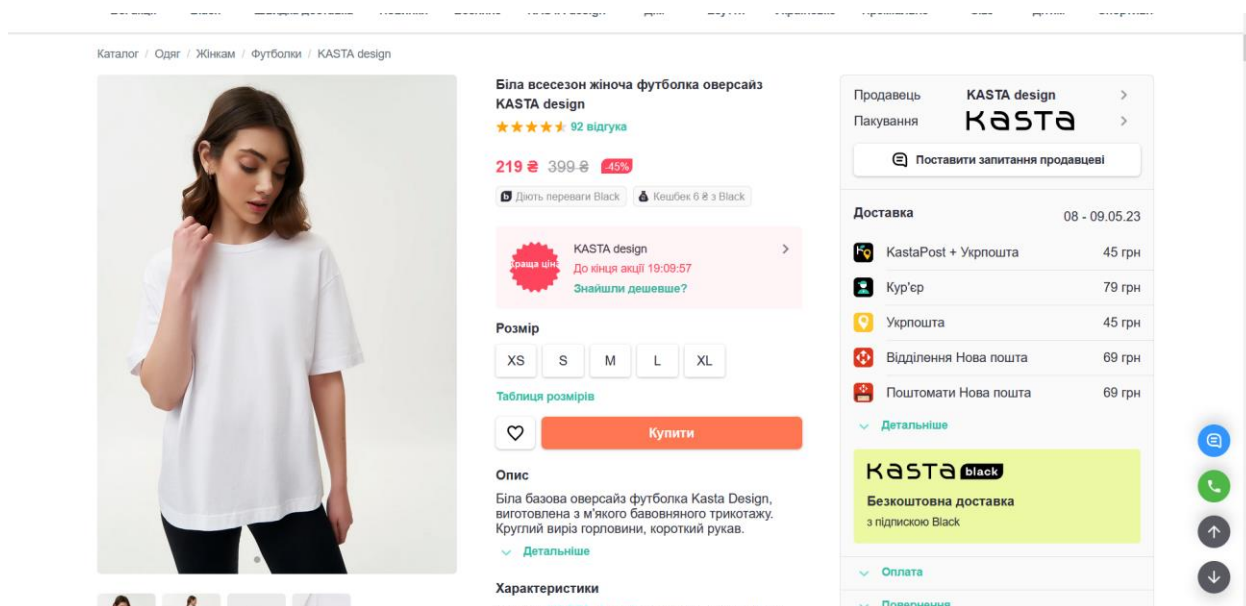


Рисунок 1.8 – Вигляд сторінки замовлення

Після детального аналізу аналогів веб-додатків для підтримки діяльності магазину одягу було визначено їх переваги та недоліки. Результати представлені в таблиці 1.1.

Таблиця 1.1. – Порівняльна характеристика аналогів веб-додатку

Характеристика	garne.com.ua	cultboutique.com.ua	kasta.ua
Сучасний дизайн	+	+	-
Зручна навігація	+	+	-
Функціональність	+	+	+
Інтерактивність	+	+	-
Підбір товару за критеріями	+	+	+
Швидке замовлення	+	+	-
Додаткові знижки	+	+	+

Після аналізу аналогічних веб-додатків було визначено, що всі розділи і сторінки сайту повинні бути виконані в єдиному оформленні, за винятком окремих функціональних блоків, наприклад, форм. Це допоможе забезпечити єдність і стилістичну цілісність додатку.

Наявність навігації на всіх сторінках сайту є обов'язковою. Користувачам повинно бути зручно та легко переходити між різними розділами та сторінками, щоб вони могли швидко знайти потрібну інформацію або функціонал.

Однією з обов'язкових умов на сайті є наявність форм зворотного зв'язку. Це дає користувачам можливість звернутися до адміністрації сайту, задати питання, висловити свої пропозиції або подати скаргу. Форми зворотного зв'язку допомагають створити ефективний канал комунікації з користувачами та підтримувати взаємодію з ними.

1.3 Постановка задачі

Метою кваліфікаційної роботи є розробка веб-додатку для підтримки роботи магазину з продаж одягу. Загальні вимоги до проєкту є наступними:

- забезпечити швидку навігацію користувача;
- забезпечити наявність авторизації;

- забезпечити швидку обробку даних;
- наявність панелі адміністратора для виконання операцій з товарами.

Для досягнення мети проєкту необхідно виконати наступні задачі:

- визначити актуальність роботи, дослідити предметну область та провести аналіз аналогів веб-додатків для підтримки роботи магазину з продажу одягу;
- виконати розробку веб-додатку;
- розробити базу даних;
- розробити та реалізувати компоненти для взаємодії додатку з клієнтом;

Вимоги до веб-додатку, необхідного програмного забезпечення та функціональності системи описані у технічному завданні (Додаток А).

Також для проєктування веб-додатку потрібно використовувати методи моделювання макета веб-додатку та бази даних (БД).

1.4 Визначення середовища розробки

У контексті розробки веб-додатку для підтримки роботи магазину з продажу одягу, було прийнято рішення використовувати наступні технології: HTML, CSS, JS, PHP та MySQL. Цей вибір базується на їхній популярності, широкому застосуванні та відповідності вимогам проєкту.

HTML (HyperText Markup Language) є основним стандартним мовою розмітки для створення веб-сторінок. Використання HTML дозволяє структурувати та представити інформацію на сторінках веб-додатку. Він забезпечує можливість розмітки тексту, вставки зображень, створення посилань та інших елементів, необхідних для побудови інтерфейсу.

CSS (Cascading Style Sheets) є мовою опису стилів і дозволяє визначити зовнішній вигляд веб-сторінок, стиль і розташування елементів, кольори, шрифти та інші аспекти дизайну. Використання CSS допомагає забезпечити єдність та консистентність вигляду веб-додатку.

JavaScript (JS) є мовою програмування, яка використовується для динамічного взаємодії з користувачем та створення інтерактивних елементів на веб-сторінках. Використання JS дозволяє реалізувати функціональність,

таку як валідація даних, асинхронні запити на сервер, динамічне оновлення вмісту сторінки та багато іншого.

PHP є серверною мовою програмування, спеціально створеною для розробки веб-додатків. Використання PHP дозволяє обробляти дані на серверній стороні, виконувати операції з базою даних, здійснювати обробку форм та забезпечувати функціональність, яка потрібна для роботи магазину, таку як керування запасами, обробка замовлень та автентифікація користувачів.

MySQL є однією з найпоширеніших систем управління базами даних (СУБД). Використання MySQL дозволяє зберігати та керувати даними, необхідними для роботи магазину, такими як інформація про товари, клієнтів, замовлення та інші відомості.

Обрані технології, враховуючи їхню широку популярність, гнучкість та сумісність між собою, надають розробникам можливість створити функціональний, естетичний та ефективний веб-додаток, що відповідає потребам магазину з продажу одягу.

1 МОДЕЛЮВАННЯ ТА ПРОЄКТУВАННЯ ВЕБ-ДОДАТКУ

2.1 Проєктування веб-додатку

Для моделювання роботи веб-системи було побудовано контекстну діаграму методології IDEF0.

Структурно-функціональне моделювання в нотації IDEF0 відіграє важливу роль у розробці веб-додатків для продажу одягу. Ця методологія забезпечує систематичний підхід до аналізу, проєктування та впровадження веб-додатків, зосереджуючись на функціях, процесах та взаємозв'язках між ними. Використання структурно-функціонального моделювання дозволяє зрозуміти бізнес-процеси, пов'язані з продажем одягу, та уявити їх у вигляді системи компонентів та взаємодій. Це дає можливість визначити ключові функції, які повинен виконувати веб-додаток, такі як керування запасами, обробка замовлень, взаємодія з клієнтами тощо. [7].

Для побудови діаграми було визначено наступні дані:

- Вхідні дані: інформація про тренди, мета реалізації додатку;
- Управління: ТЗ, методики розробки, вимоги до виконання;
- Механізми: системне програмне забезпечення, студент-виконавець;
- Вихідні дані: реаліований веб-додаток, створена документація.

Функціональне моделювання веб-додатку представлено на рисунку 2.1.

Для деталізації внутрішніх процесів у діаграмі IDEF0 було виконано декомпозицію, яка представлена на рисунку 2.2.

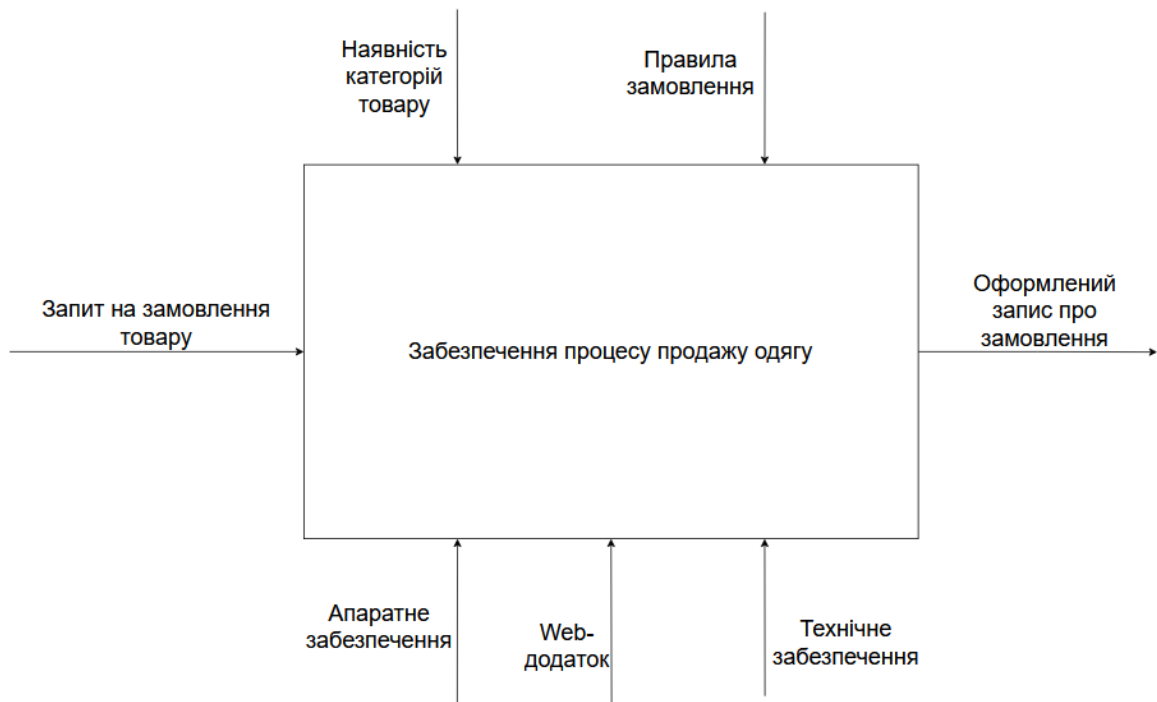


Рисунок 2.1 – Контекстна діаграма

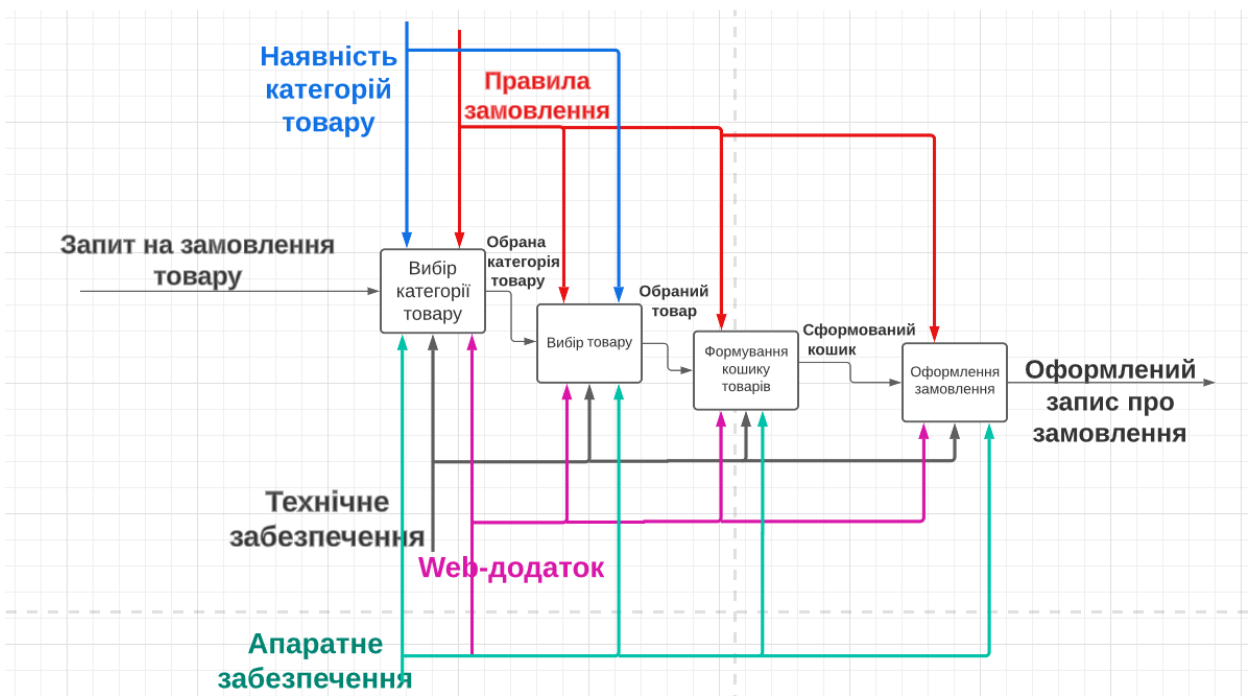


Рисунок 2.1 – Діаграма декомпозиції першого рівня

UML діаграми дозволяють розробникам візуалізувати та зрозуміти структуру системи. Зокрема, діаграми класів UML використовуються для відображення класів, їхніх взаємозв'язків та атрибутів. У контексті веб-додатків для продажу одягу це може включати класи, такі як "Товар",

"Замовлення", "Користувач" та інші. Використання цих діаграм допомагає встановити зв'язки між класами, ідентифікувати необхідні методи та атрибути, а також забезпечити однорідність та належну організацію коду [8].

Крім того, UML діаграми послугують для моделювання поведінки системи. Наприклад, діаграми послідовності UML можуть бути використані для уявлення взаємодії між різними компонентами системи в конкретних сценаріях, таких як процес замовлення або оплата товару. Це допомагає розробникам уточнити послідовність дій, ідентифікувати ролі та взаємодіючі компоненти, а також забезпечити відповідність бізнес-процесам та вимогам клієнтів [9].

Діаграму Use-Case представлено на рисунку 2.3.

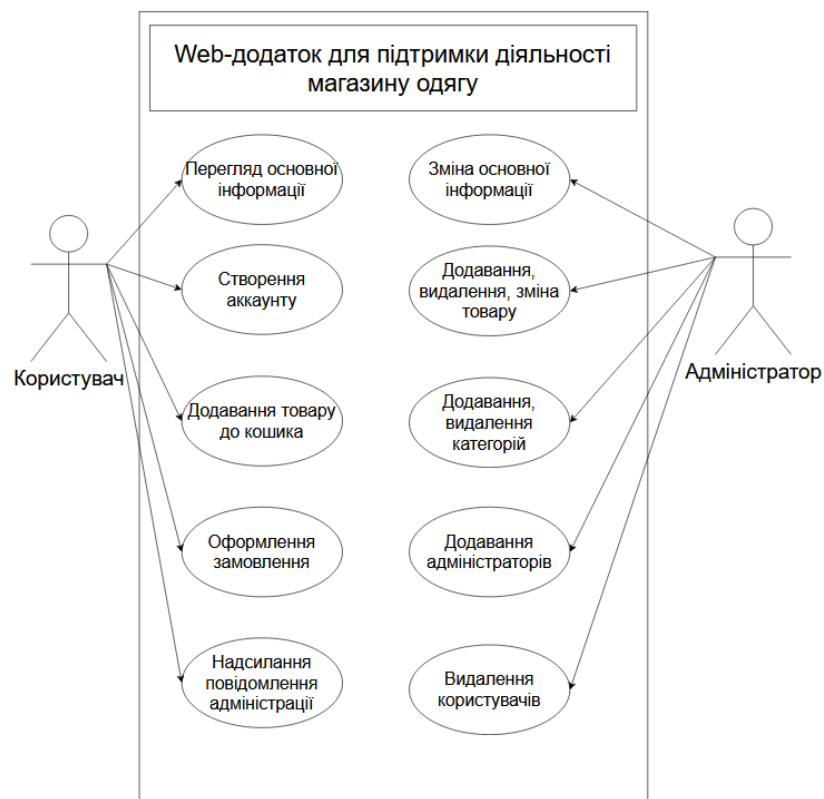


Рисунок 2.3 – Use-Case діаграма

2.2 Реалізація бази даних

База даних – це життєво важливий інструмент для зберігання, управління та пошуку інформації. Вона також має вирішальне значення для побудови системи веб-додатку. Добре структурована база даних забезпечує роботу сайту та керує всіма взаємодіями в системі [10]. Хороший дизайн бази даних для підтримки діяльності магазину одягу включає в себе:

- Просту, функціональну структуру бази даних: Структура таблиць бази даних проста, але охоплює всю необхідну функціональність без шкоди для користувацького досвіду.

- Висока продуктивність: Запити до бази даних виконуються швидко, щоб полегшити взаємодію з клієнтами в реальному часі і підтримувати безперебійний процес покупок. Тому обрана база даних повинна мати хороші можливості індексації та оптимізації продуктивності.

- Висока доступність і масштабованість: Хороший дизайн бази даних забезпечує високу доступність за допомогою автоматичних знімків і дозволяє автоматично масштабувати її для підтримки майбутнього зростання платформи, а також раптових стрибків трафіку.

У процесі проектування БД було створено наступні таблиці з атрибутами:

- about_us (інформація про сайт);
- admins (дані для входу в адмін панель);
- bundle_product_relation (товари, які продаються комплектами);
- card (кошик доданих товарів);
- categories (категорії товарів);
- contact_us (інформація про контакти для зв'язку);
- coupons (купони для знижок);
- customers (користувачі сайту);
- customer_orders (замовлення користувачів);
- manufacturers (виробники або постачальники товару);

- payments (оплата товару);
- pending_orders (завершені замовлення);
- products (наявні товари);
- products_categories (категорії товарів);
- terms (правила користування);
- wishlist (товар, який сподобався);

На рисунку 2.4 зображено реалізовану модель БД веб-додатку для підтримки діяльності магазину одягу.

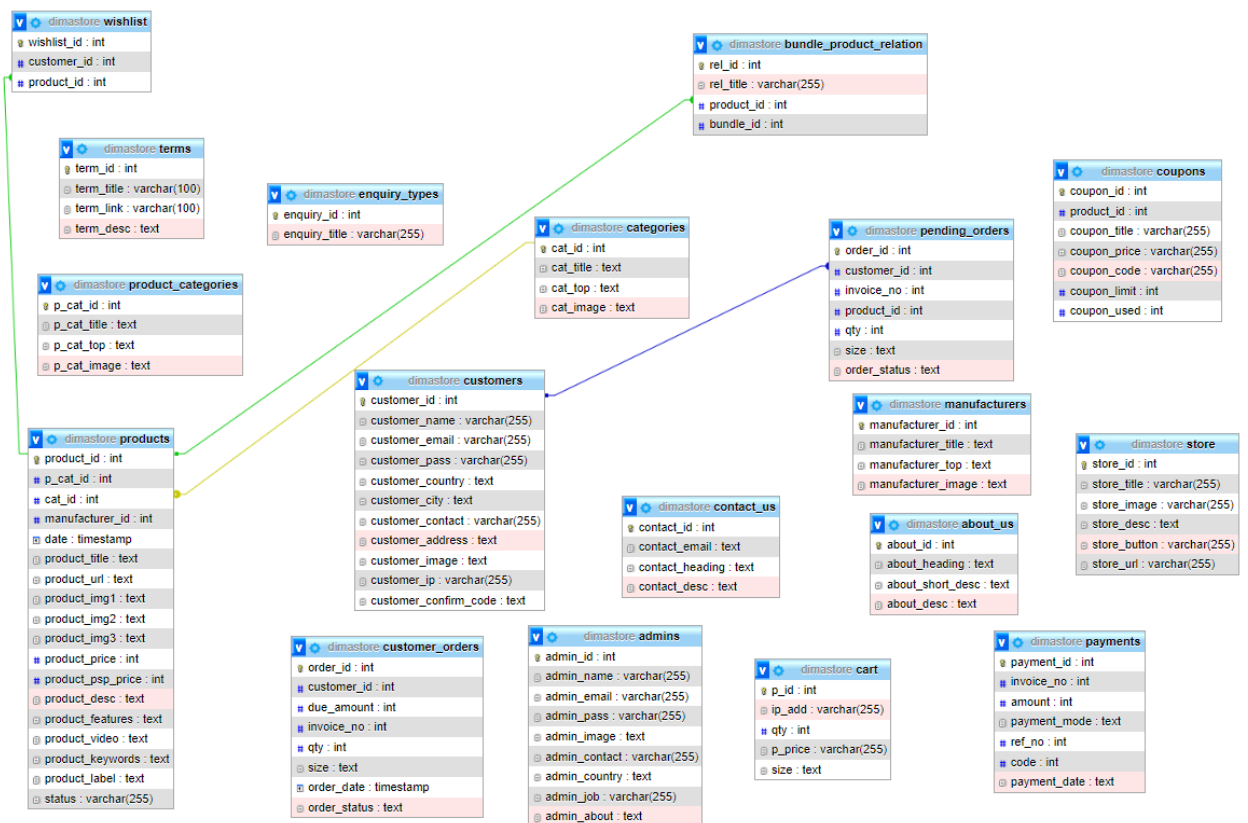


Рисунок 2.2 – Реалізована модель бази даних

3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Архітектура веб-додатку

Для більш чіткого розуміння архітектури веб-додатку, було побудовано діаграму проєктів High Level Design (HLD) [11], яка надає всебічний огляд процесу розробки програмного забезпечення разом з архітектурою системи, додатками, управлінням базами даних, а також повною блок-схемою системи та навігації. Це план, який об'єднує різні етапи та модулі, їхні цілі, змінні компоненти, результати, архітектуру та часові рамки розробки програмного забезпечення. HLD перетворює бізнес-план на програмний продукт або послугу. На рисунку 3.1 представлено результат створення.

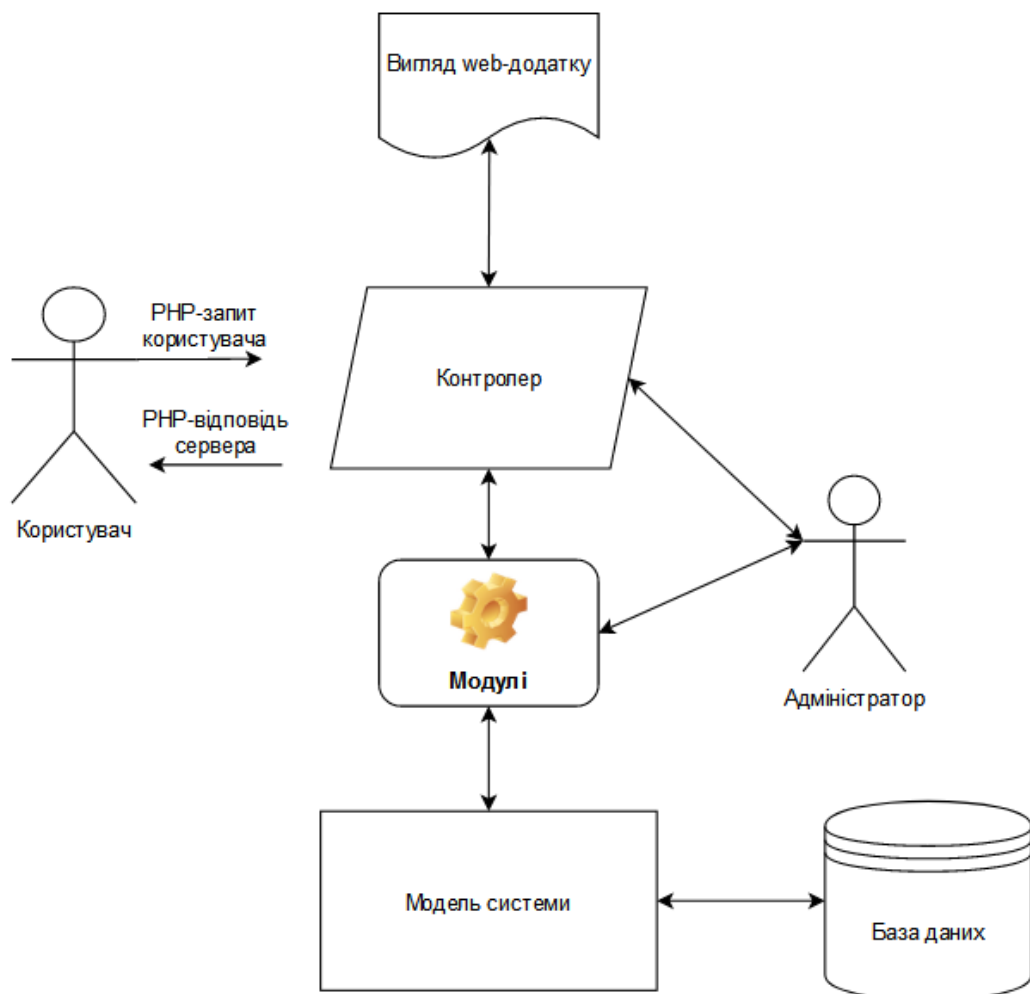


Рисунок 3.1 – High Level Design діаграма веб-додатку

Дана діаграма містить наступні компоненти:

1. Користувацький інтерфейс «Вигляд», який надає представлення даних системи за допомогою стилів та графічних матеріалів.
2. Контролер обробки даних.
3. Модулі веб-додатку для виконання певних операцій.
4. Модель інформаційної системи, яка організовує взаємозв'язок контролеру та БД.
5. База даних, яка містить всю інформацію для наповнення сайту.

Для того, щоб розпочати реалізацію веб-додатку для підтримки діяльності магазину одягу, необхідно обрати редактор коду, який допоможе полегшити процес розробки за допомогою безкоштовних плагінів для редагування та автоматизації створення сторінок веб-додатку [12]. У якості редактору було обрано Visual Studio Code [13], який має наступні переваги:

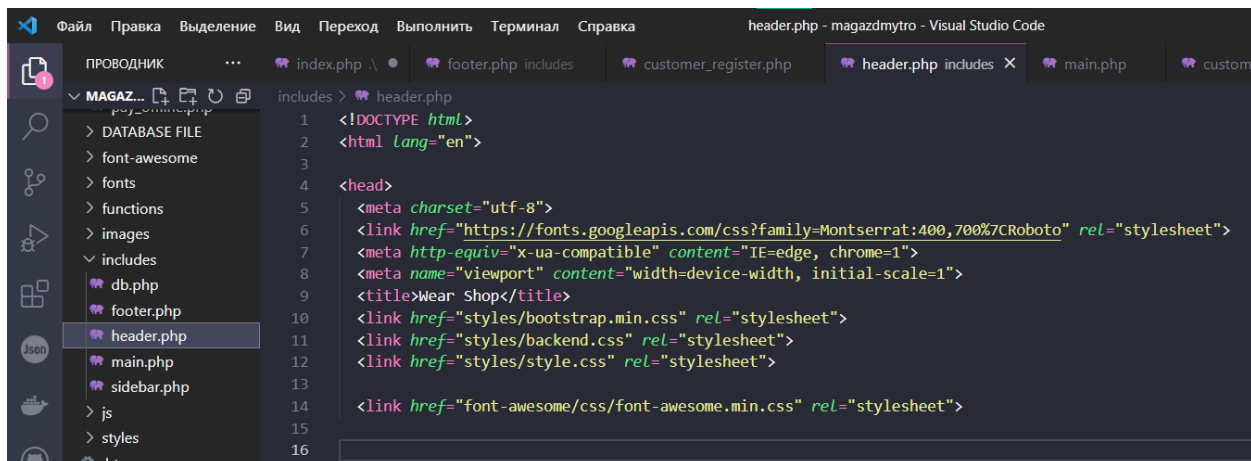
- Простота в використанні
- Швидкий в ознайомленні
- Розширення

На ринку існує значна кількість розширень для редактора коду VS Code, які пропонують широкий спектр функціональності. Ці розширення можуть включати різні можливості, від зміни теми інтерфейсу до підтримки різних мов програмування, налагодження коду, інтеграції з Git та навіть взаємодії з плеєрами музики, такими як Spotify.

VS Code надає користувачам можливість створювати власні розширення за допомогою JavaScript або TypeScript та доступу до детальної документації. Розширення відіграють важливу роль у розвитку VS Code, надаючи йому велику потужність і гнучкість. Крім того, VS Code має розширений набір інструментів для розробки веб-технологій, таких як JSX/React, HTML, CSS, SCSS, Less та JSON.

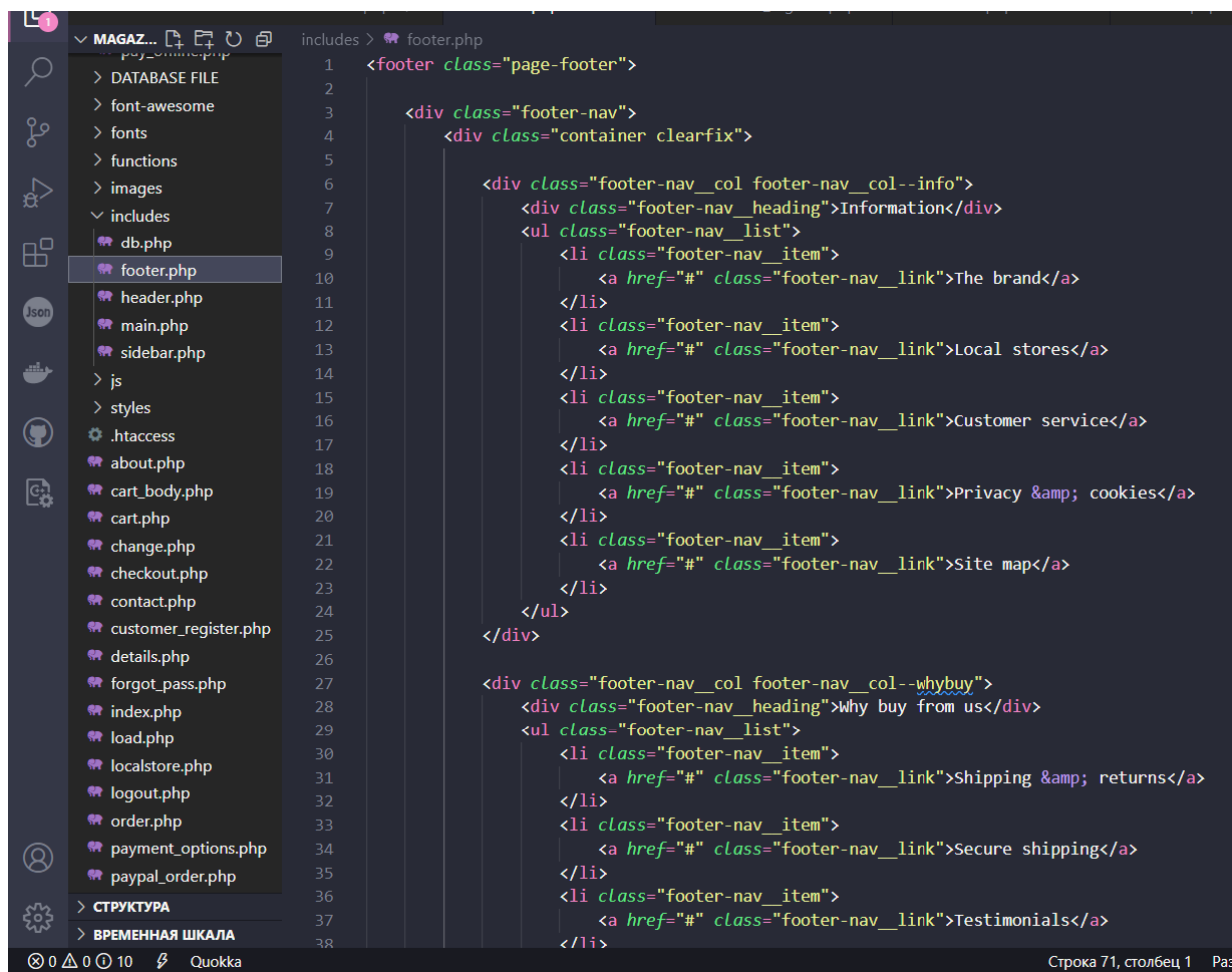
Макет сторінки веб-додатка складається з роздільних блочних елементів, включаючи верхню частину (header), основний контент (content) та

нижню частину (footer). На рисунках 3.2 – 3.3 зображено реалізацію блоків «header» та «footer» за допомогою редактору коду VS Code.



```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5 <meta charset="utf-8">
6 <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700%7CRoboto" rel="stylesheet">
7 <meta http-equiv="x-ua-compatible" content="IE=edge, chrome=1">
8 <meta name="viewport" content="width=device-width, initial-scale=1">
9 <title>Wear Shop</title>
10 <link href="styles/bootstrap.min.css" rel="stylesheet">
11 <link href="styles/backend.css" rel="stylesheet">
12 <link href="styles/style.css" rel="stylesheet">
13
14 <link href="font-awesome/css/font-awesome.min.css" rel="stylesheet">
```

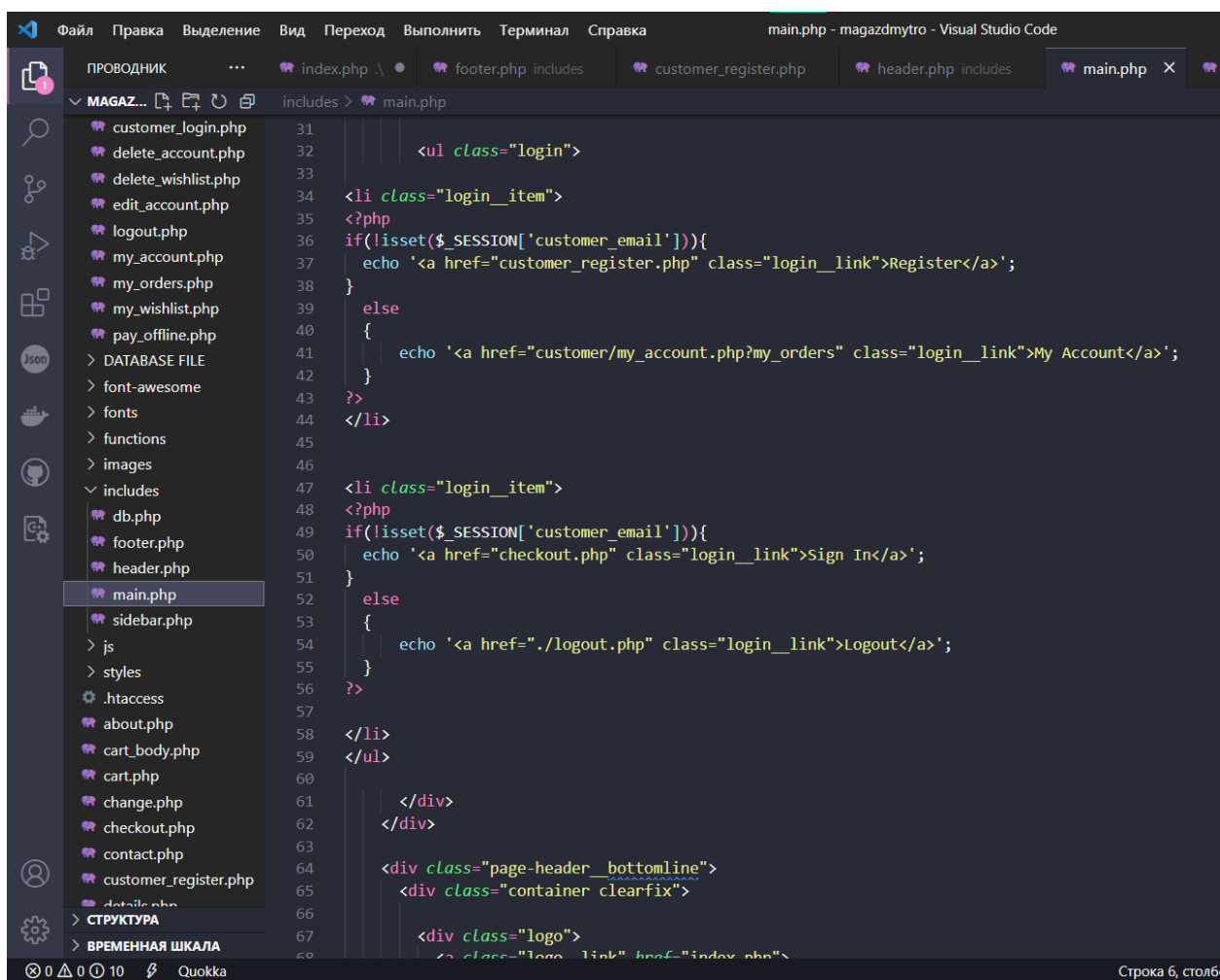
Рисунок 3.2 – Реалізація блоку «header» в VS Code



```
1 <footer class="page-footer">
2
3 <div class="footer-nav">
4 <div class="container clearfix">
5
6 <div class="footer-nav_col footer-nav_col--info">
7 <div class="footer-nav_heading">Information</div>
8 <ul class="footer-nav_list">
9 <li class="footer-nav_item">
10 <a href="#" class="footer-nav_link">The brand</a>
11 </li>
12 <li class="footer-nav_item">
13 <a href="#" class="footer-nav_link">Local stores</a>
14 </li>
15 <li class="footer-nav_item">
16 <a href="#" class="footer-nav_link">Customer service</a>
17 </li>
18 <li class="footer-nav_item">
19 <a href="#" class="footer-nav_link">Privacy & cookies</a>
20 </li>
21 <li class="footer-nav_item">
22 <a href="#" class="footer-nav_link">Site map</a>
23 </li>
24 </ul>
25 </div>
26
27 <div class="footer-nav_col footer-nav_col--whybuy">
28 <div class="footer-nav_heading">Why buy from us</div>
29 <ul class="footer-nav_list">
30 <li class="footer-nav_item">
31 <a href="#" class="footer-nav_link">Shipping & returns</a>
32 </li>
33 <li class="footer-nav_item">
34 <a href="#" class="footer-nav_link">Secure shipping</a>
35 </li>
36 <li class="footer-nav_item">
37 <a href="#" class="footer-nav_link">Testimonials</a>
38 </li>
```

Рисунок 3.3 – Реалізація блоку «footer» в VS Code

Основний блок «content» містить всю основну інформацію про наповнення сайту та відрізняється від інших сторінок. На головній сторінці знаходиться логотип сайту, меню, навігація з допомогою якої користувач зможе перейти на сторінку реєстрації, авторизації, або переглянути вміст кошику товарів. Також блок доданими товарами надає базове розуміння про наповнення сайту. На рисунку 3.4 представлено частину реалізації блоку «content» за допомогою VS Code. Повний лістинг коду «main.php» наведено в додатку В.



```
31 <ul class="login">
32 <li class="login_item">
33 <?php
34 if(isset($_SESSION['customer_email'])){
35 echo '<a href="customer_register.php" class="login_link">Register</a>';
36 }
37 else
38 {
39 echo '<a href="customer/my_account.php?my_orders" class="login_link">My Account</a>';
40 }
41 </li>
42 <li class="login_item">
43 <?php
44 if(isset($_SESSION['customer_email'])){
45 echo '<a href="checkout.php" class="login_link">Sign In</a>';
46 }
47 else
48 {
49 echo '<a href="./logout.php" class="login_link">Logout</a>';
50 }
51 </li>
52 </ul>
53 </div>
54 </div>
55 <div class="page-header_bottomline">
56 <div class="container clearfix">
57 <div class="logo">
58 <a class="logo_link" href="index.php">
```

Рисунок 3.4 – Реалізація блоку «content» в VS Code

Для більш інтерактивної взаємодії між елементами, а також створення анімаційних блоків були використані технології JavaScript в поєднанні з JQuery та CSS з використанням фреймворку Bootstrap [14]. На рисунках 3.5 –

3.6 представлено частину реалізації скриптів та оформлення стилів в редакторі VS Code.

```
function(a,b){
  "use strict";
  "object"==typeof module && "object"==typeof module.exports ? module.exports=a.document ? b(a, !0) :
  function(a){
    if(!a.document) throw new Error("jQuery requires a window with a document");
    return b(a) : b(a)
  }
  ("undefined" != typeof window?window: this, function(a,b)
  {"use strict";
  var c=[],
  d=a.document,
  e=object.getPrototypeOf,
  f=c.slice,g=c.concat,
  h=c.push,
  i=c.indexOf,
  j={},
  k=j.toString,
  l=j.hasOwnProperty,
  m=l.toString,
  n=m.call(Object),
  o={};
  function p(a,b){
    b=b || d;
    var c=b.createElement("script");
    c.text=a,
    b.head.appendChild(c).parentNode.removeChild(c)
    var q="3.1.1",
    r=function(a,b){return new r.fn.init(a,b)},
    s=/^\s*(?:\u201c\u201d|[\s\u201c\u201d])+/g,
    t=/^\s*(?:\u201c\u201d|[\s\u201c\u201d])+/g;
```

Рисунок 3.5 – Частина скрипта для веб-додатку в VS Code

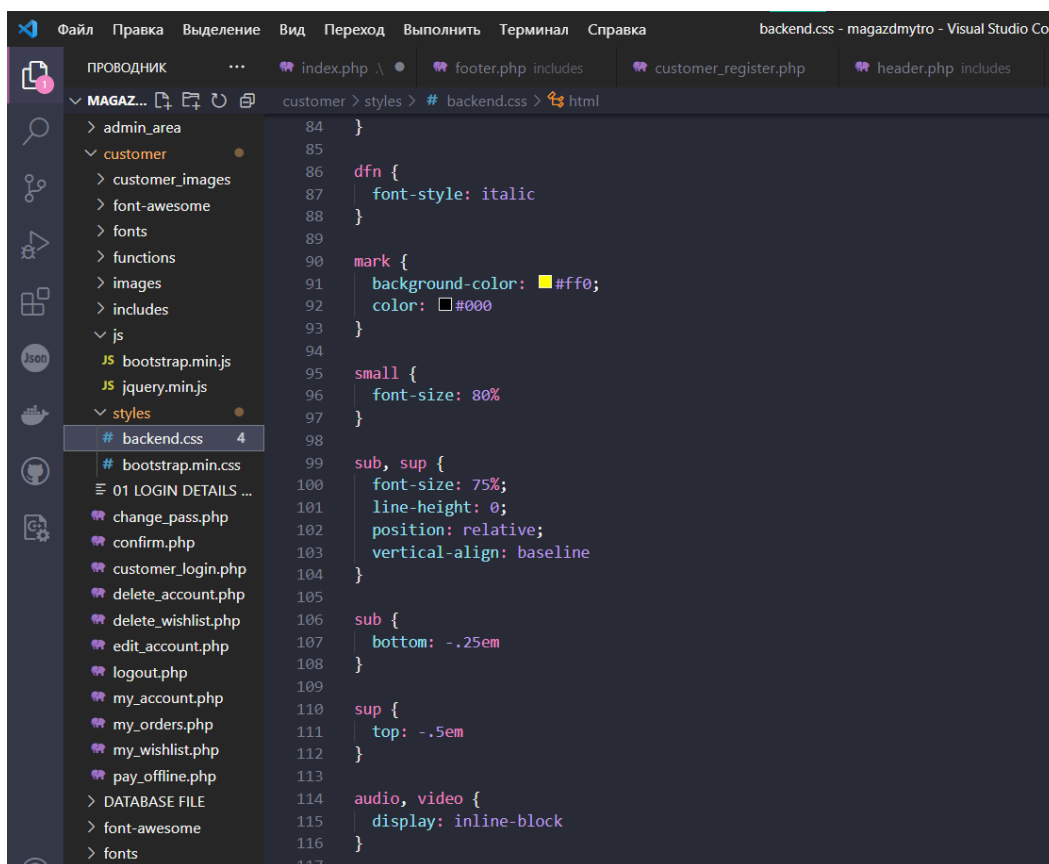


Рисунок 3.6 – Реалізація стилів в VS Code

База даних «ecom_store.sql» була створена для зберігання основної інформації яка використовується в якості наповнення веб-додатку. Для розробки було обрано локальний сервер «OpenServer» з використанням інструменту «PHPMyAdmin». Таблиці БД представлено на рисунках 3.7 – 3.24. Детальний опис таблиць було наведено в етапі проектування бази даних.

Таблица	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
<input type="checkbox"/> about_us	★ [иконки]	1	InnoDB	latin1_swedish_ci	16.0 КиБ	-
<input type="checkbox"/> admins	★ [иконки]	1	InnoDB	latin1_swedish_ci	16.0 КиБ	-
<input type="checkbox"/> bundle_product_relation	★ [иконки]	0	InnoDB	latin1_swedish_ci	16.0 КиБ	-
<input type="checkbox"/> cart	★ [иконки]	3	InnoDB	latin1_swedish_ci	16.0 КиБ	-
<input type="checkbox"/> categories	★ [иконки]	4	InnoDB	latin1_swedish_ci	16.0 КиБ	-
<input type="checkbox"/> contact_us	★ [иконки]	1	InnoDB	latin1_swedish_ci	16.0 КиБ	-
<input type="checkbox"/> coupons	★ [иконки]	2	InnoDB	latin1_swedish_ci	16.0 КиБ	-
<input type="checkbox"/> customers	★ [иконки]	2	InnoDB	latin1_swedish_ci	16.0 КиБ	-
<input type="checkbox"/> customer_orders	★ [иконки]	10	InnoDB	latin1_swedish_ci	16.0 КиБ	-
<input type="checkbox"/> enquiry_types	★ [иконки]	3	InnoDB	latin1_swedish_ci	16.0 КиБ	-
<input type="checkbox"/> manufacturers	★ [иконки]	6	InnoDB	latin1_swedish_ci	16.0 КиБ	-
<input type="checkbox"/> payments	★ [иконки]	15	InnoDB	latin1_swedish_ci	16.0 КиБ	-
<input type="checkbox"/> pending_orders	★ [иконки]	10	InnoDB	latin1_swedish_ci	16.0 КиБ	-
<input type="checkbox"/> products	★ [иконки]	7	InnoDB	latin1_swedish_ci	16.0 КиБ	-
<input type="checkbox"/> product_categories	★ [иконки]	6	InnoDB	latin1_swedish_ci	16.0 КиБ	-
<input type="checkbox"/> store	★ [иконки]	2	InnoDB	latin1_swedish_ci	16.0 КиБ	-
<input type="checkbox"/> terms	★ [иконки]	3	InnoDB	latin1_swedish_ci	16.0 КиБ	-
<input type="checkbox"/> wishlist	★ [иконки]	6	InnoDB	latin1_swedish_ci	16.0 КиБ	-

Рисунок 3.7 – Список таблиць «ecom_store.sql»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/> 1	about_id 🔑	int			Нет	Нет		AUTO_INCREMENT	[иконки]
<input type="checkbox"/> 2	about_heading	text	utf8_general_ci		Нет	Нет			[иконки]
<input type="checkbox"/> 3	about_short_desc	text	utf8_general_ci		Нет	Нет			[иконки]
<input type="checkbox"/> 4	about_desc	text	utf8_general_ci		Нет	Нет			[иконки]

Рисунок 3.8 – Структура таблиці «about_us»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 admin_id	int			Нет	Нем		AUTO_INCREMENT	
<input type="checkbox"/>	2 admin_name	varchar(255)	utf8_general_ci		Нет	Нем			
<input type="checkbox"/>	3 admin_email	varchar(255)	utf8_general_ci		Нет	Нем			
<input type="checkbox"/>	4 admin_pass	varchar(255)	utf8_general_ci		Нет	Нем			
<input type="checkbox"/>	5 admin_image	text	utf8_general_ci		Нет	Нем			
<input type="checkbox"/>	6 admin_contact	varchar(255)	utf8_general_ci		Нет	Нем			
<input type="checkbox"/>	7 admin_country	text	utf8_general_ci		Нет	Нем			
<input type="checkbox"/>	8 admin_job	varchar(255)	ucs2_general_ci		Нет	Нем			
<input type="checkbox"/>	9 admin_about	text	utf8_general_ci		Нет	Нем			

Рисунок 3.9 – Структура таблиці «admins»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 rel_id	int			Нет	Нем		AUTO_INCREMENT	
<input type="checkbox"/>	2 rel_title	varchar(255)	utf8_general_ci		Нет	Нем			
<input type="checkbox"/>	3 product_id	int			Нет	Нем			
<input type="checkbox"/>	4 bundle_id	int			Нет	Нем			

Рисунок 3.10 – Структура таблиці «bundle_product_relation»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 p_id	int			Нет	Нем			
<input type="checkbox"/>	2 ip_add	varchar(255)	utf8_general_ci		Нет	Нем			
<input type="checkbox"/>	3 qty	int			Нет	Нем			
<input type="checkbox"/>	4 p_price	varchar(255)	utf8_general_ci		Нет	Нем			
<input type="checkbox"/>	5 size	text	utf8_general_ci		Нет	Нем			

Рисунок 3.11 – Структура таблиці «card»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 cat_id	int			Нет	Нем		AUTO_INCREMENT	
<input type="checkbox"/>	2 cat_title	text	utf8_general_ci		Нет	Нем			
<input type="checkbox"/>	3 cat_top	text	utf8_general_ci		Нет	Нем			
<input type="checkbox"/>	4 cat_image	text	utf8_general_ci		Нет	Нем			

Рисунок 3.12 – Структура таблиці «categories»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 contact_id	int			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 contact_email	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	3 contact_heading	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	4 contact_desc	text	utf8_general_ci		Нет	Нет			

Рисунок 3.13 – Структура таблицы «contact_us»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 coupon_id	int			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 product_id	int			Нет	Нет			
<input type="checkbox"/>	3 coupon_title	varchar(255)	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	4 coupon_price	varchar(255)	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	5 coupon_code	varchar(255)	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	6 coupon_limit	int			Нет	Нет			
<input type="checkbox"/>	7 coupon_used	int			Нет	Нет			

Рисунок 3.14 – Структура таблицы «coupons»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 customer_id	int			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 customer_name	varchar(255)	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	3 customer_email	varchar(255)	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	4 customer_pass	varchar(255)	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	5 customer_country	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	6 customer_city	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	7 customer_contact	varchar(255)	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	8 customer_address	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	9 customer_image	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	10 customer_ip	varchar(255)	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	11 customer_confirm_code	text	utf8_general_ci		Нет	Нет			

Рисунок 3.15 – Структура таблицы «customers»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 order_id	int			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 customer_id	int			Нет	Нет			
<input type="checkbox"/>	3 due_amount	int			Нет	Нет			
<input type="checkbox"/>	4 invoice_no	int			Нет	Нет			
<input type="checkbox"/>	5 qty	int			Нет	Нет			
<input type="checkbox"/>	6 size	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	7 order_date	timestamp		on update CURRENT_TIMESTAMP	Нет	CURRENT_TIMESTAMP		DEFAULT_GENERATED ON UPDATE CURRENT_TIMESTAMP	
<input type="checkbox"/>	8 order_status	text	utf8_general_ci		Нет	Нет			

Рисунок 3.16 – Структура таблицы «customers_orders»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 enquiry_id	int			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 enquiry_title	varchar(255)	utf8_general_ci		Нет	Нет			

Рисунок 3.17 – Структура таблицы «enquiry_types»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 manufacturer_id	int			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 manufacturer_title	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	3 manufacturer_top	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	4 manufacturer_image	text	utf8_general_ci		Нет	Нет			

Рисунок 3.18 – Структура таблицы «manufacturers»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 payment_id	int			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 invoice_no	int			Нет	Нет			
<input type="checkbox"/>	3 amount	int			Нет	Нет			
<input type="checkbox"/>	4 payment_mode	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	5 ref_no	int			Нет	Нет			
<input type="checkbox"/>	6 code	int			Нет	Нет			
<input type="checkbox"/>	7 payment_date	text	utf8_general_ci		Нет	Нет			

Рисунок 3.19 – Структура таблицы «payments»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 order_id	int			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 customer_id	int			Нет	Нет			
<input type="checkbox"/>	3 invoice_no	int			Нет	Нет			
<input type="checkbox"/>	4 product_id	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	5 qty	int			Нет	Нет			
<input type="checkbox"/>	6 size	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	7 order_status	text	utf8_general_ci		Нет	Нет			

Рисунок 3.20 – Структура таблицы «pending_orders»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 product_id	int			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 p_cat_id	int			Нет	Нет			
<input type="checkbox"/>	3 cat_id	int			Нет	Нет			
<input type="checkbox"/>	4 manufacturer_id	int			Нет	Нет			
<input type="checkbox"/>	5 date	timestamp		on update CURRENT_TIMESTAMP	Нет	CURRENT_TIMESTAMP		DEFAULT_GENERATED ON UPDATE CURRENT_TIMESTAMP	
<input type="checkbox"/>	6 product_title	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	7 product_url	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	8 product_img1	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	9 product_img2	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	10 product_img3	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	11 product_price	int			Нет	Нет			
<input type="checkbox"/>	12 product_psp_price	int			Нет	Нет			
<input type="checkbox"/>	13 product_desc	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	14 product_features	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	15 product_video	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	16 product_keywords	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	17 product_label	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	18 status	varchar(255)	utf8_general_ci		Нет	Нет			

Рисунок 3.21 – Структура таблиці «products»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 p_cat_id	int			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 p_cat_title	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	3 p_cat_top	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	4 p_cat_image	text	utf8_general_ci		Нет	Нет			

Рисунок 3.21 – Структура таблиці «product_categories»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 store_id	int			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 store_title	varchar(255)	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	3 store_image	varchar(255)	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	4 store_desc	text	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	5 store_button	varchar(255)	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	6 store_url	varchar(255)	utf8_general_ci		Нет	Нет			

Рисунок 3.22 – Структура таблиці «store»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 term_id	int			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 term_title	varchar(100)	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	3 term_link	varchar(100)	utf8_general_ci		Нет	Нет			
<input type="checkbox"/>	4 term_desc	text	utf8_general_ci		Нет	Нет			

Рисунок 3.23 – Структура таблиці «terms»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно	Действие
<input type="checkbox"/>	1 wishlist_id	int			Нет	Нет		AUTO_INCREMENT	
<input type="checkbox"/>	2 customer_id	int			Нет	Нет			
<input type="checkbox"/>	3 product_id	int			Нет	Нет			

Рисунок 3.24 – Структура таблиці «wishlist»

3.2 Розробка користувацької частини веб-додатку

Процес роботи з веб-додатком розпочинається з головної сторінки, яка зображена на рисунках 3.25 – 3.27. У шапці сайту міститься привітання користувача, навігаційний розділ для авторизації та реєстрації, а також блок з інформацією про додані товари в кошик. Основний інформаційний блок веб-додатку містить логотип сайту, банер для переходу до перегляду товарів, та блок з частиною доданих товарів. Підвал сайту містить меню, в якому користувач може перейти на сторінки авторизації та отримати основну інформацію про діяльність магазину. Для більш широкого охоплення аудиторії як в Україні так і за межами, веб-додаток подає всю інформацію англійською мовою.

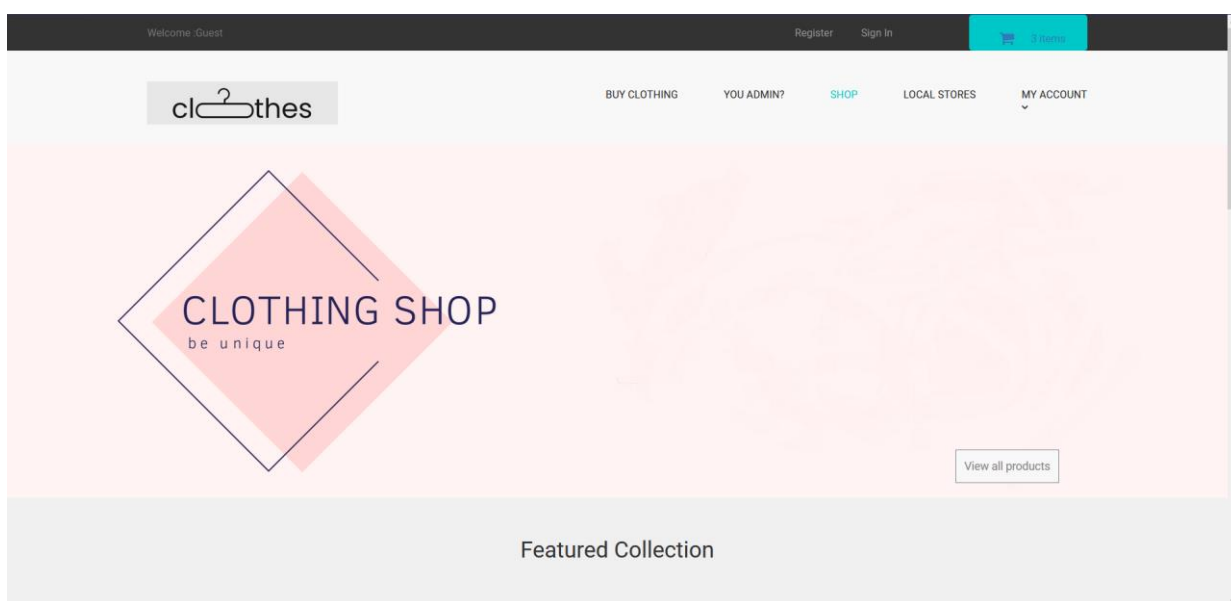


Рисунок 3.25 – Перша частина головної сторінки

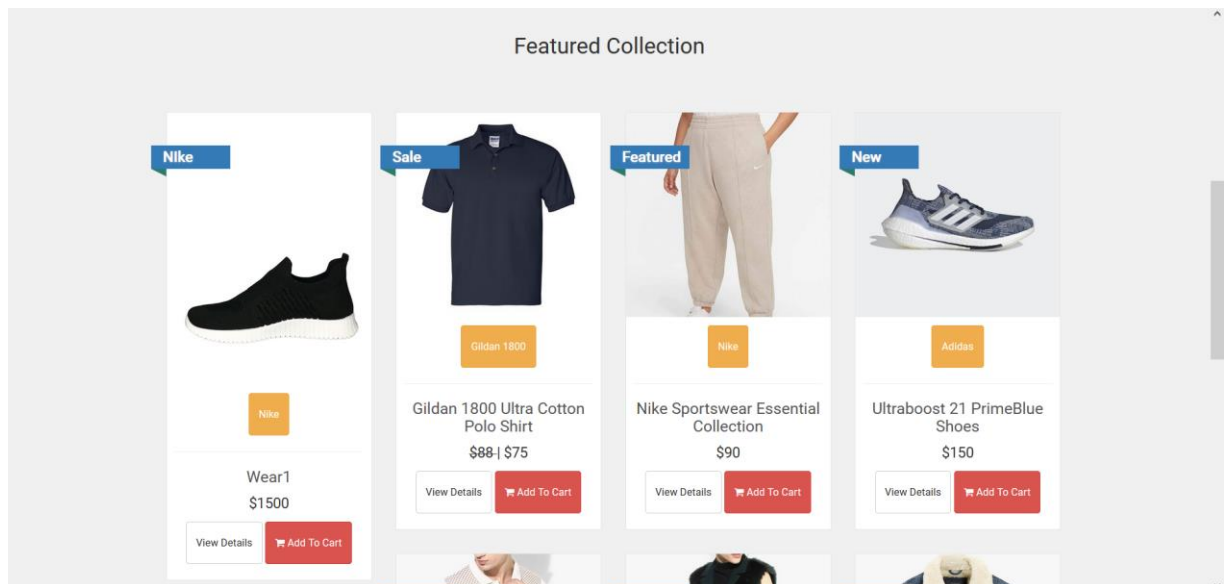


Рисунок 3.26 – Друга частина головної сторінки

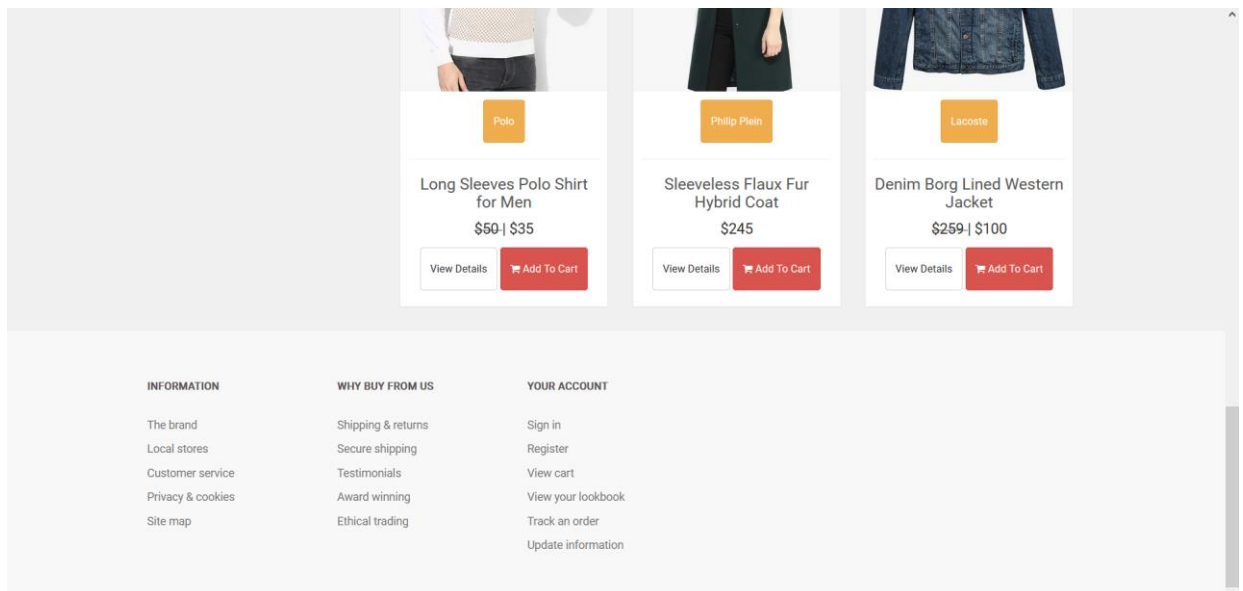


Рисунок 3.27 – Третя частина головної сторінки

Для того, щоб переглянути весь асортимент товарів, необхідно натиснути на елемент «BUY CLOTHING» щоб перейти на сторінку. На сторінці надано основну інформацію про товар, можливість перейти на сторінку обраного товару, а також додати товар до кошику. На рисунках 3.28 – 3.30 представлено вигляд сторінки «shop.php».

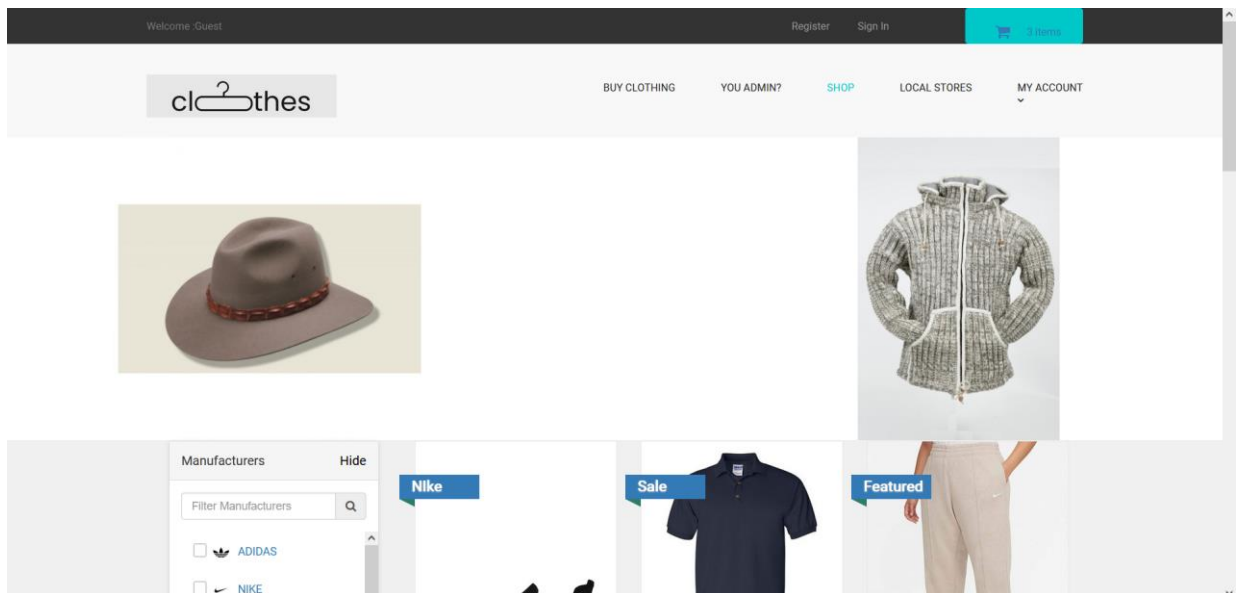


Рисунок 3.28 – Перша частина сторінки «shop.php»

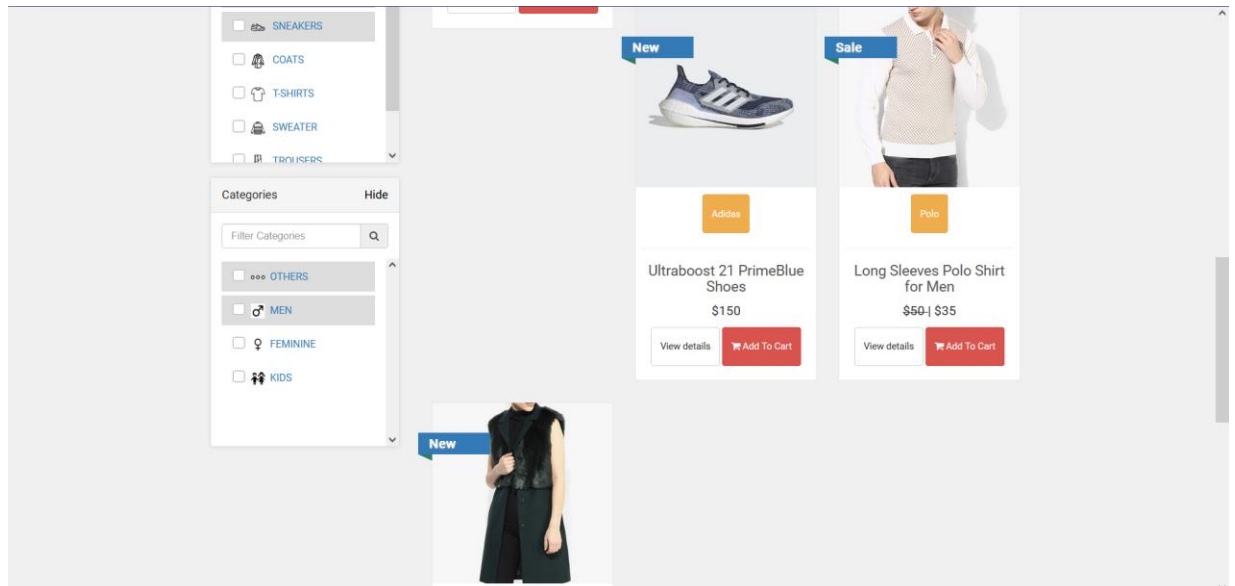


Рисунок 3.29 – Друга частина сторінки «shop.php»

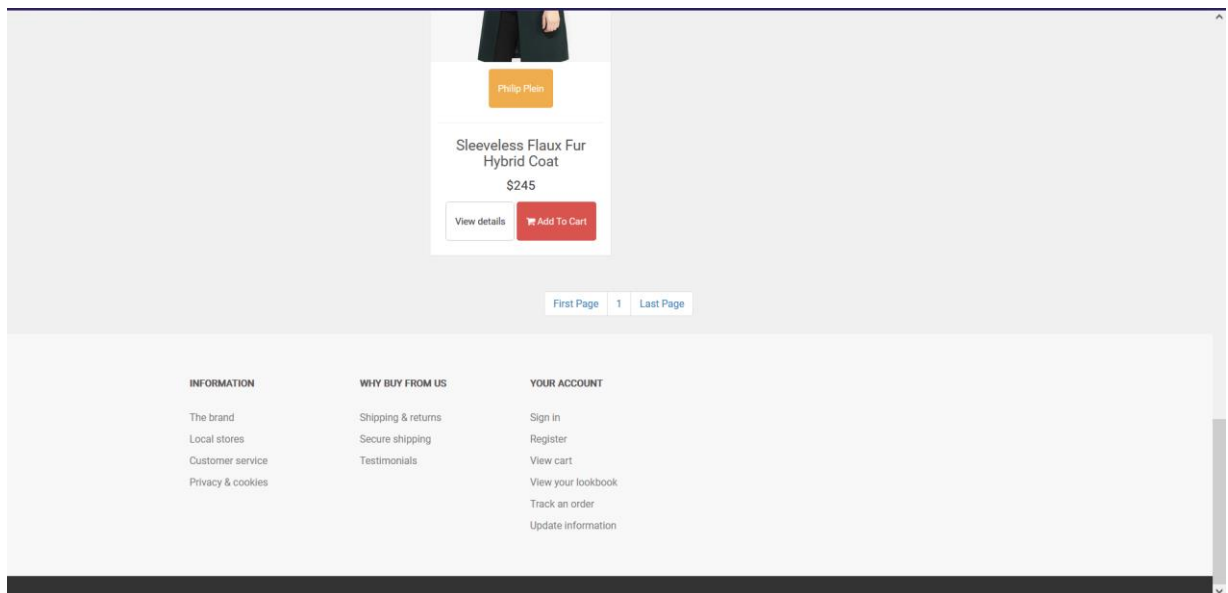


Рисунок 3.30 – Третя частина сторінки «shop.php»

Увесь функціонал для перегляду товару наданий також і неавторизованим користувачам. Після переходу на сторінку обраного товару, користувач отримує можливість переглянути інші фото товару, опис, та характеристики товару. Кожна сторінка товару має унікальну адресу, яка створюється при додаванні товару з боку адміністратора Також є можливість обрати розмір, та кількість одиниць товару для замовлення. На рисунку 3.31 зображено вигляд сторінки «cotton-polo-shirt».

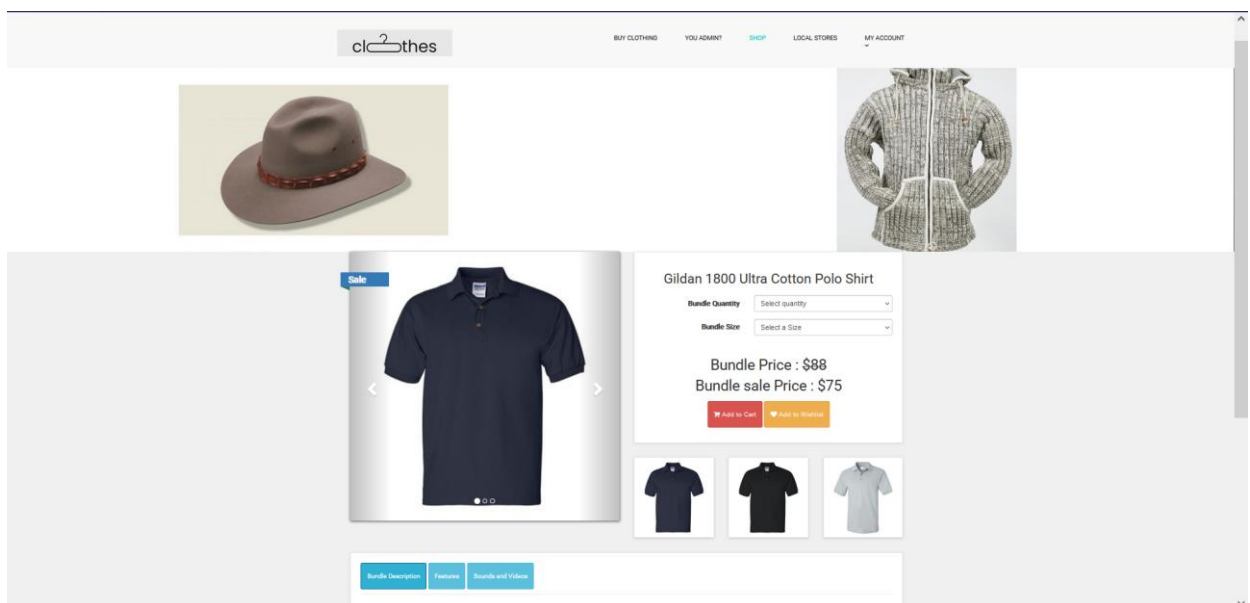


Рисунок 3.31 – Вигляд сторінки «cotton-polo-shirt»

Після того, як користувач додав товар до кошику, то виконується переадресація на сторінку з переглядом доданих товарів. Головна особливість кошику – відслідковування історії формування навіть не авторизованого користувача через IP-адресу пристрою. Також при замовленні користувач може оновити повністю кошик та використати купон, який надає знижку на замовлення. На рисунку 3.32 представлено вигляд сторінки «card.php».

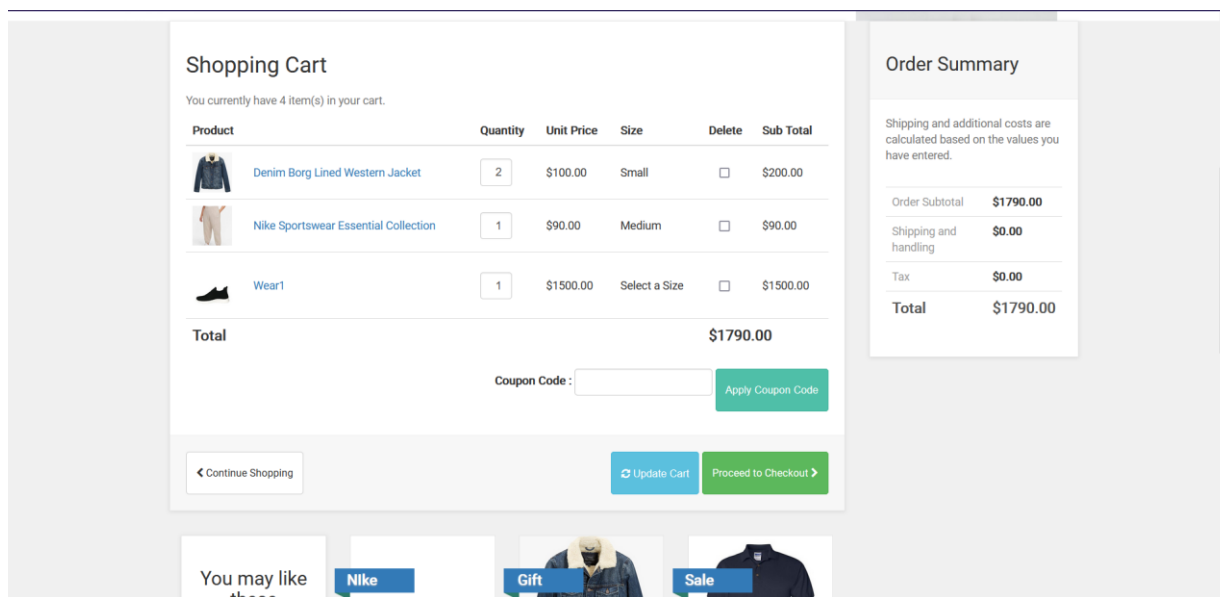


Рисунок 3.32 – Вигляд сторінки «card.php»

Коли користувач використовує купон, то система надсилає сповіщення про те, що операцію було виконано (рис.3.33).

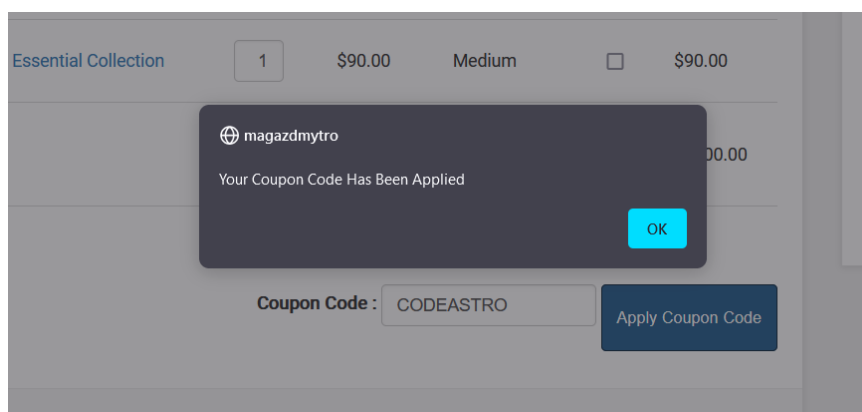


Рисунок 3.33 – Сповідження системи про успішне застосування купону

Якщо користувач незареєстрований в системі, то його буде автоматично переадресовано на сторінку авторизації. Також є опція вибору реєстрації

нового користувача. Система реєстрації передбачає перевірку даних, а також оцінку створеного пароля. Також користувач при реєстрації відразу зможе ввести основні дані, які необхідні для доставки. На рисунку 3.34 представлено вигляд сторінки реєстрації нового користувача.

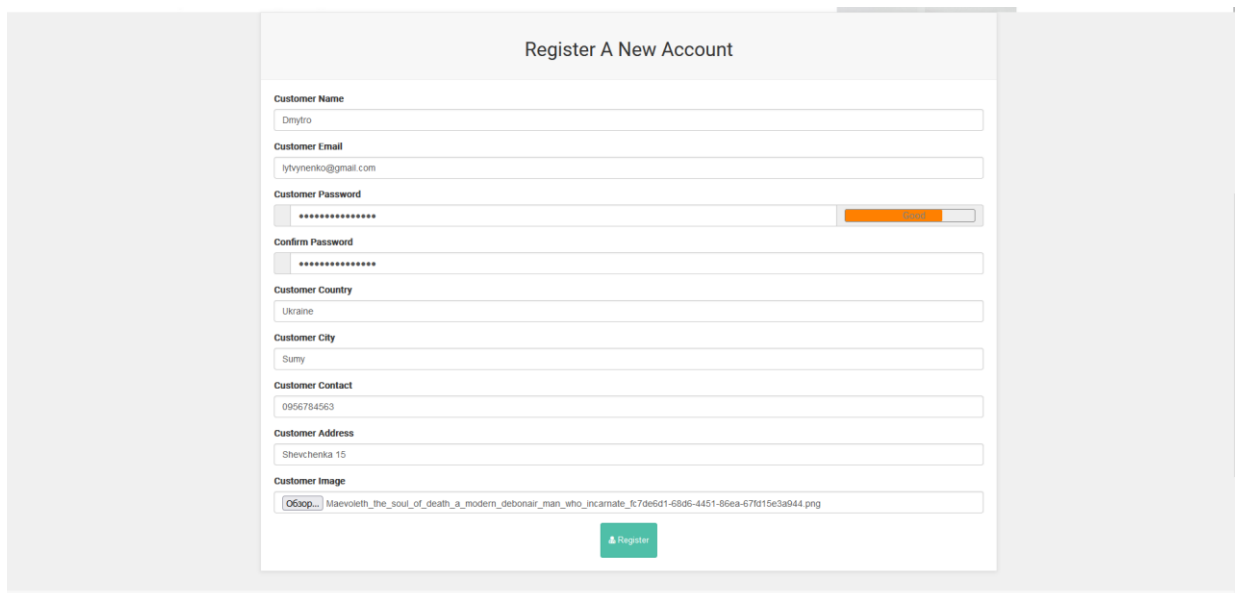


Рисунок 3.34 – Вигляд сторінки реєстрації користувача

Після успішної реєстрації в системі користувач отримає сповіщення (рис. 3.35)

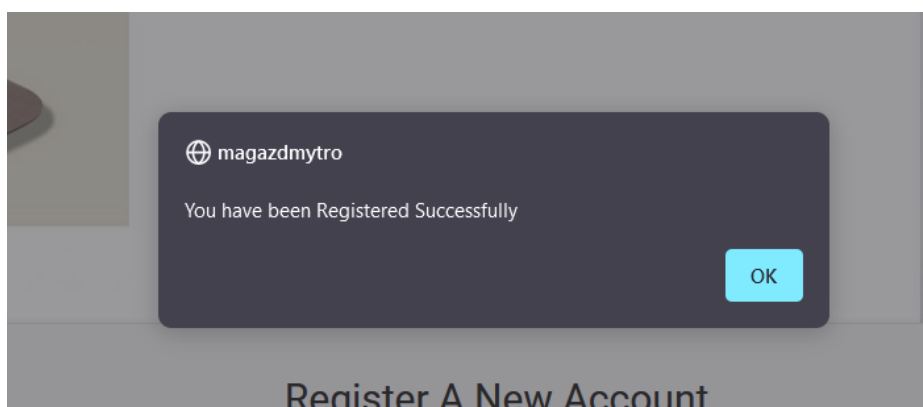


Рисунок 3.35 – Сповіщення користувача про успішну реєстрацію

Якщо користувач успішно зареєструвався то він зможе знову виконати операцію оформлення замовлення. Після того як замовлення було сформовано, система надає 2 вибори оплати – через систему PayPal та офлайн

оплату. При виборі будь-якого з способів, система сповістить користувача про успішне оформлення замовлення (рис. 3.36).

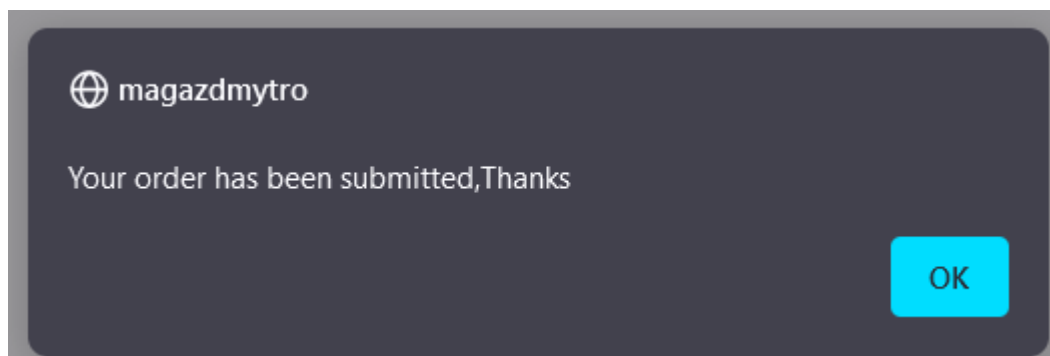


Рисунок 3.36 – Сповіднення користувача про успішне замовлення

Щоб переглянути всі замовлення, а також основну інформацію, користувач може перейти на власну сторінку яка знаходиться за посиланням в шапці профілю. На рисунку 3.37 представлено вигляд особистої сторінки користувача.

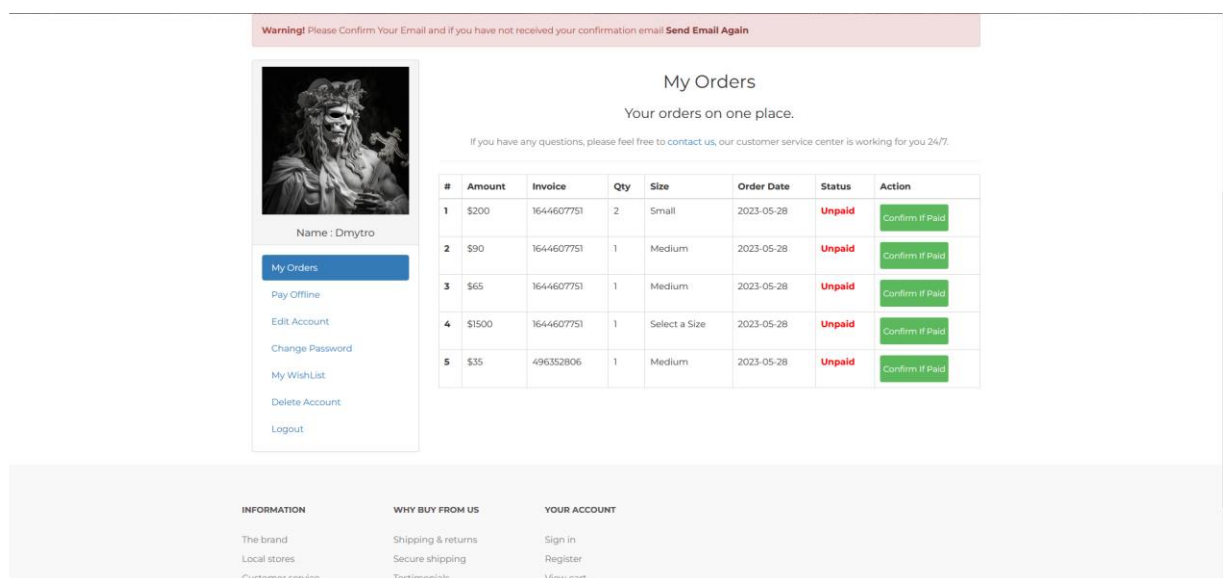


Рисунок 3.37 – Вигляд особистої сторінки користувача

На особистій сторінці користувач може оплатити вартість товару через оффлайн-оплату, змінити особисті дані з паролем, переглянути список товарів які сподобалися та видалити особисту сторінку. На рисунку 3.38 зображено вигляд сторінки для редагування особистої інформації.

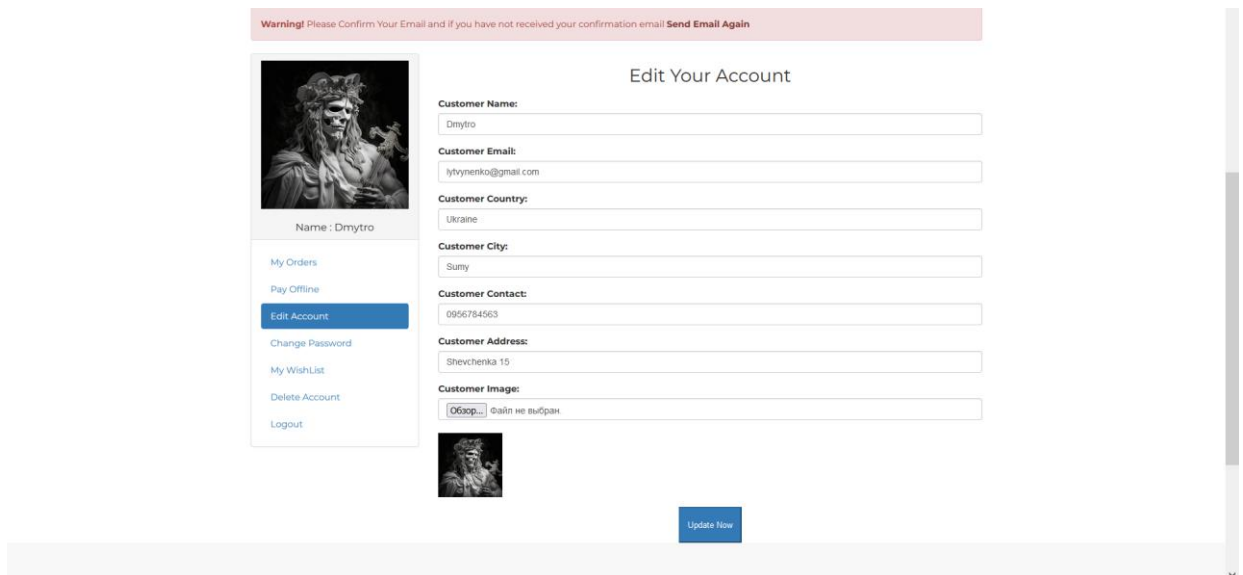


Рисунок 3.38 – Вигляд сторінки для редагування особистої інформації

Сторінка для оплати оффлайн містить реквізити банку, на які потрібно переказати кошти. Вигляд сторінки зображено на рисунку 3.39.

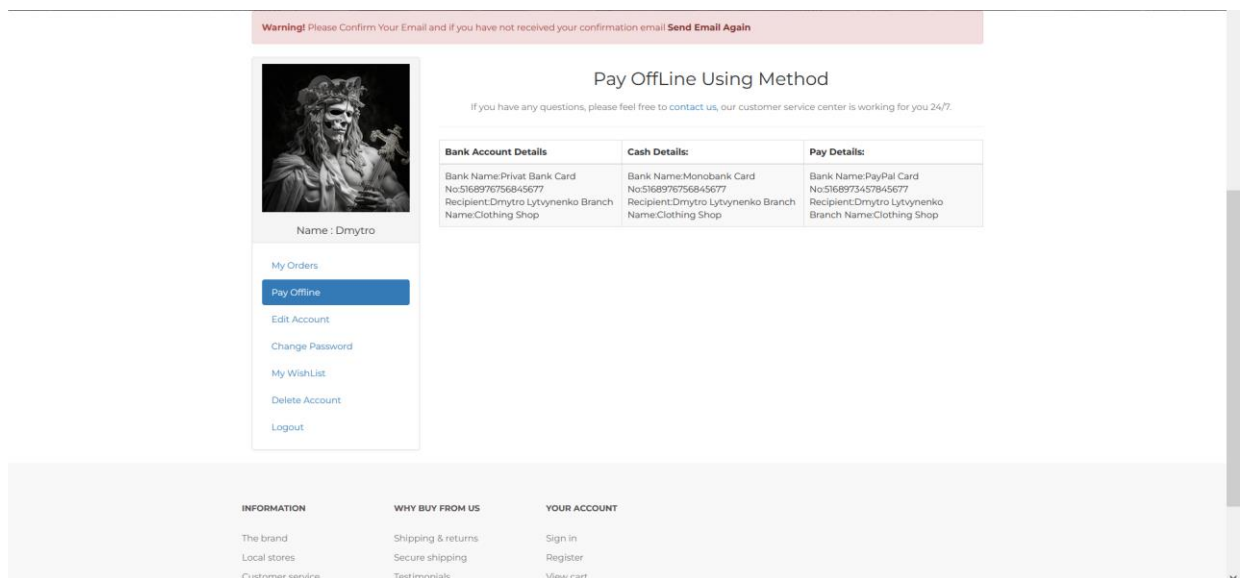


Рисунок 3.39 – Вигляд сторінки оффлайн оплати

Сторінка зміни паролю містить форму, яка потребує від користувача знання актуального паролю. Якщо користувач невірно введе дані, то система надасть відповідне повідомлення (рис. 3.40). Вигляд сторінки зміни паролю зображено на рисунку 3.41.

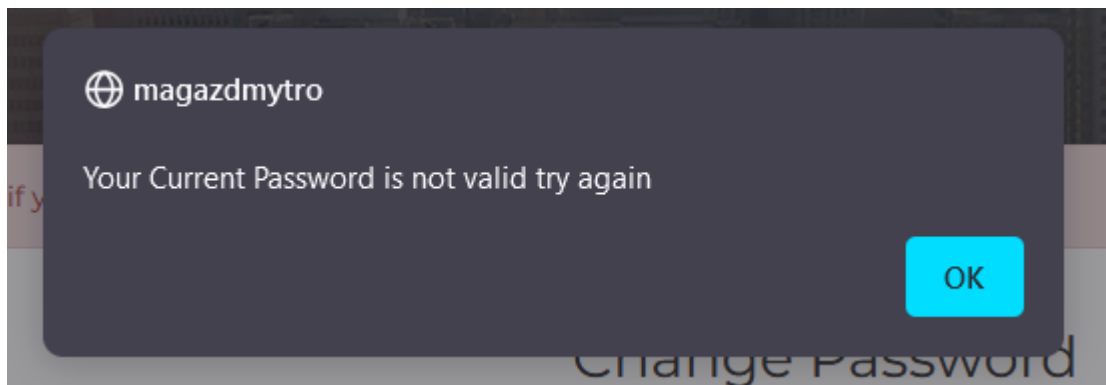


Рисунок 3.40 – Повідомлення про неправильні дані

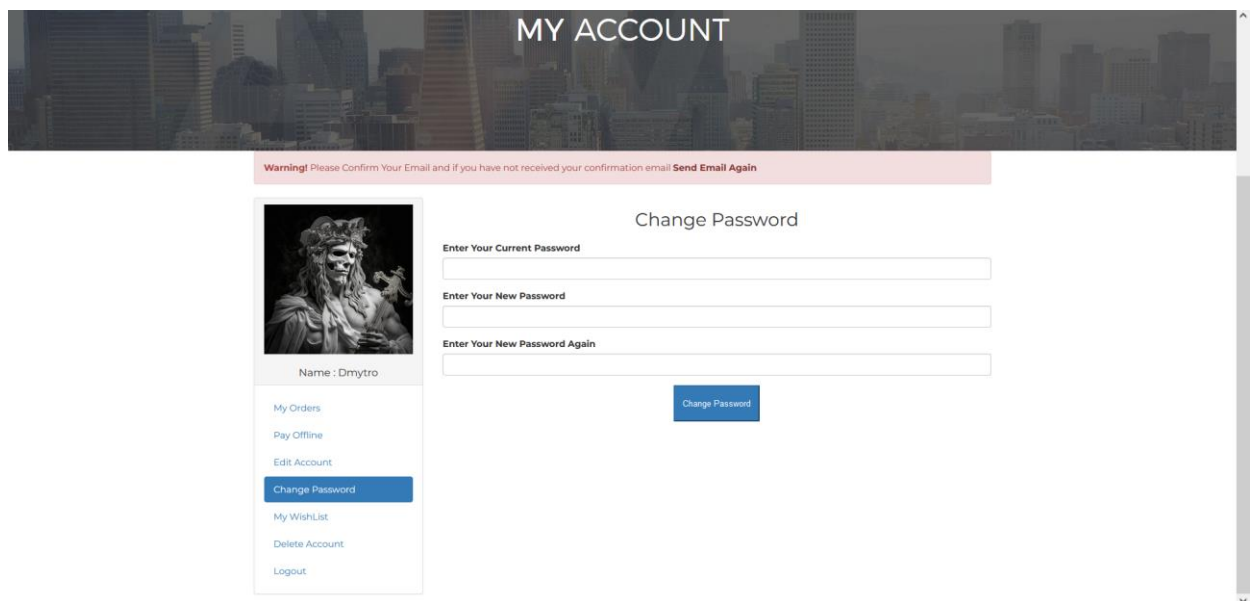


Рисунок 3.41 – Вигляд сторінки зміни паролю

Наступні сторінки веб-додатку містять інформаційний зміст для користувача. В них міститься інформація про доставку товару, або про діяльність магазину в цілому. На рисунку 3.42 представлено вигляд сторінки «About Us» на якій міститься інформація про діяльність магазину.

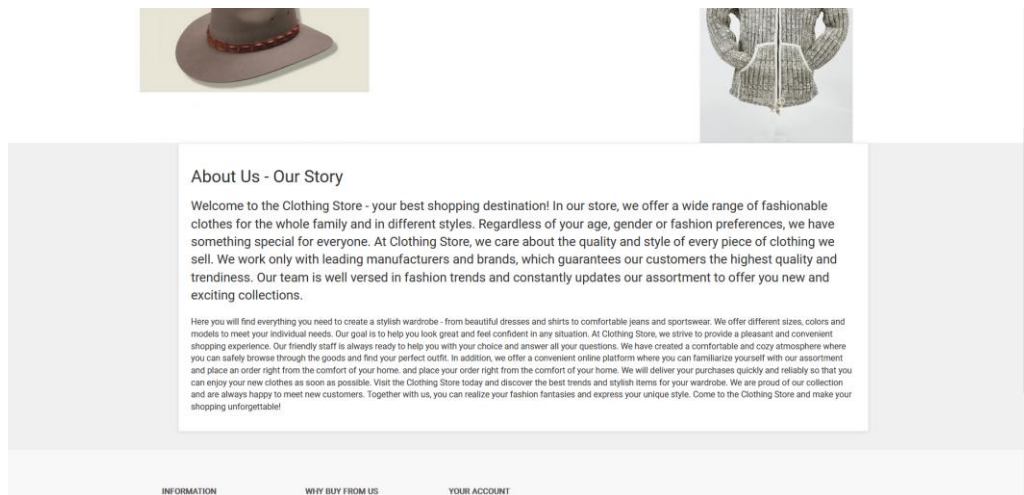


Рисунок 3.42 – Вигляд сторінки «About Us»

Сторінка «Rules» надає основну інформацію про правила використання сайту та описує дії, які необхідно виконати щоб уникнути проблем під час використання веб-додатку. На рисунку 3.43 представлено вигляд сторінки «Rules».

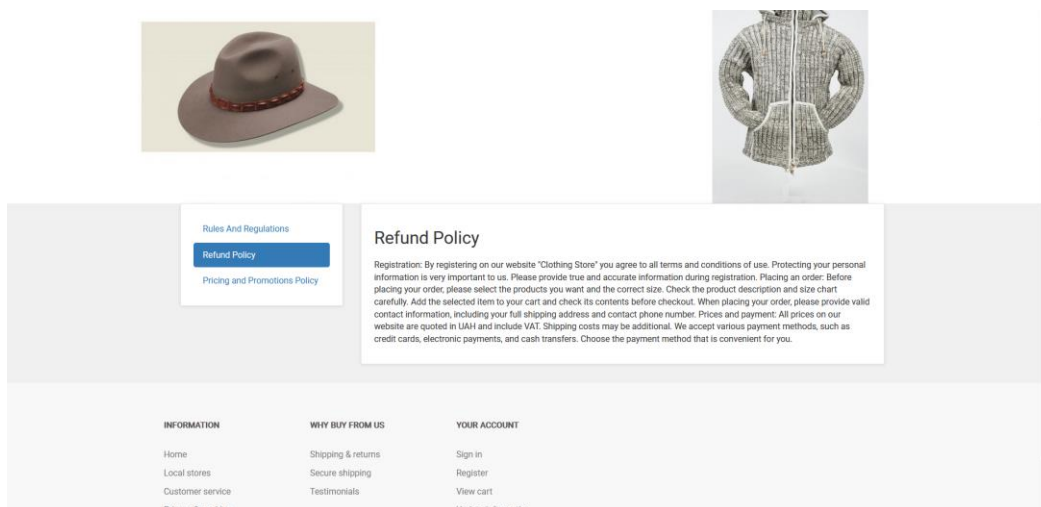


Рисунок 3.43 – Вигляд сторінки «Rules»

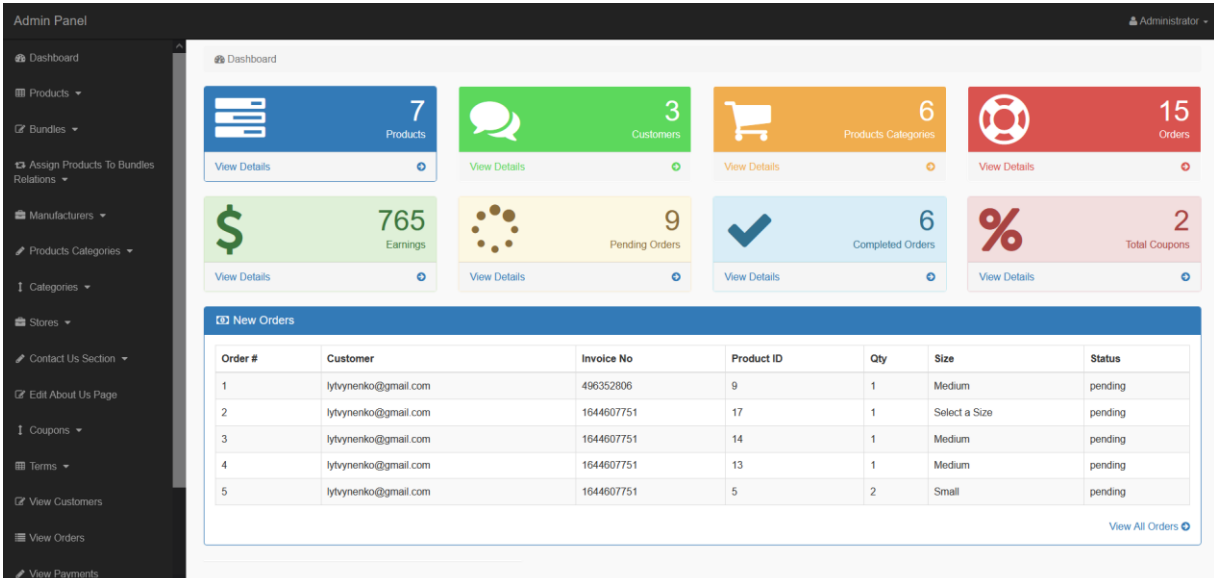
3.3 Розробка адміністративної частини веб-додатку

Адміністративна панель надає можливість редагувати дані з БД, використовуючи для цього лише веб-додаток. Адміністратори можуть

додавати, видаляти та редагувати товари що дозволяє сайту швидко змінювати товари на нові та які користуються популярністю серед покупців. Також адміністратор може переглядати та керувати замовленнями, які були надіслані клієнтами, а також змінювати статус замовлень.

Система моніторингу користувачів допоможе змінювати інформацію та видаляти неактивні аккаунти. Також система керування веб-додатком допоможе отримати узагальнену інформацію про відвідуваність сайту та аналітику замовлень.

Для початку роботи необхідно виконати авторизацію адміністратора. Для цього необхідно на URL адресу http://magazdmytro/admin_area/login.php та ввести правильні дані. Після того як дані були введені правильно, адміністратор отримає повідомлення про успішну авторизацію. На рисунку 3.44 представлено вигляд адміністративної панелі. Шаблон панелі було запозичено з сервісу платних шаблонів для адміністративної панелі [templatemonster](http://templatemonster.com). Під час процесу розробки веб-додатку шаблон було змінено, відповідно до власних вподобань.



The screenshot displays an administrative dashboard with a dark sidebar on the left containing navigation links such as 'Dashboard', 'Products', 'Bundles', 'Manufacturers', 'Products Categories', 'Categories', 'Stores', 'Contact Us Section', 'Coupons', 'Terms', 'View Customers', 'View Orders', and 'View Payments'. The main content area features a 'Dashboard' header and eight summary cards with icons and data: 7 Products, 3 Customers, 6 Products Categories, 15 Orders, 765 Earnings, 9 Pending Orders, 6 Completed Orders, and 2 Total Coupons. Below these cards is a section titled 'New Orders' containing a table with the following data:

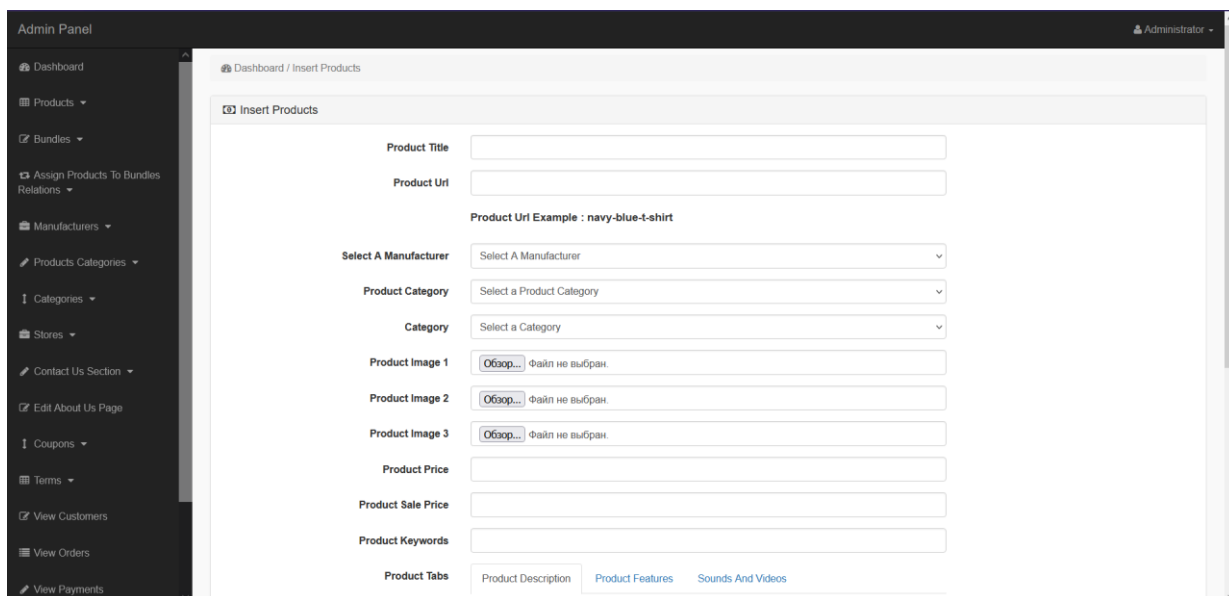
Order #	Customer	Invoice No	Product ID	Qty	Size	Status
1	lytynenko@gmail.com	496352806	9	1	Medium	pending
2	lytynenko@gmail.com	1644607751	17	1	Select a Size	pending
3	lytynenko@gmail.com	1644607751	14	1	Medium	pending
4	lytynenko@gmail.com	1644607751	13	1	Medium	pending
5	lytynenko@gmail.com	1644607751	5	2	Small	pending

Рисунок 3.44 – Вигляд адміністративної панелі

Головна дошка має систему навігації та надає основну інформацію про події, які виконувалися користувачами під час користування веб-додатком.

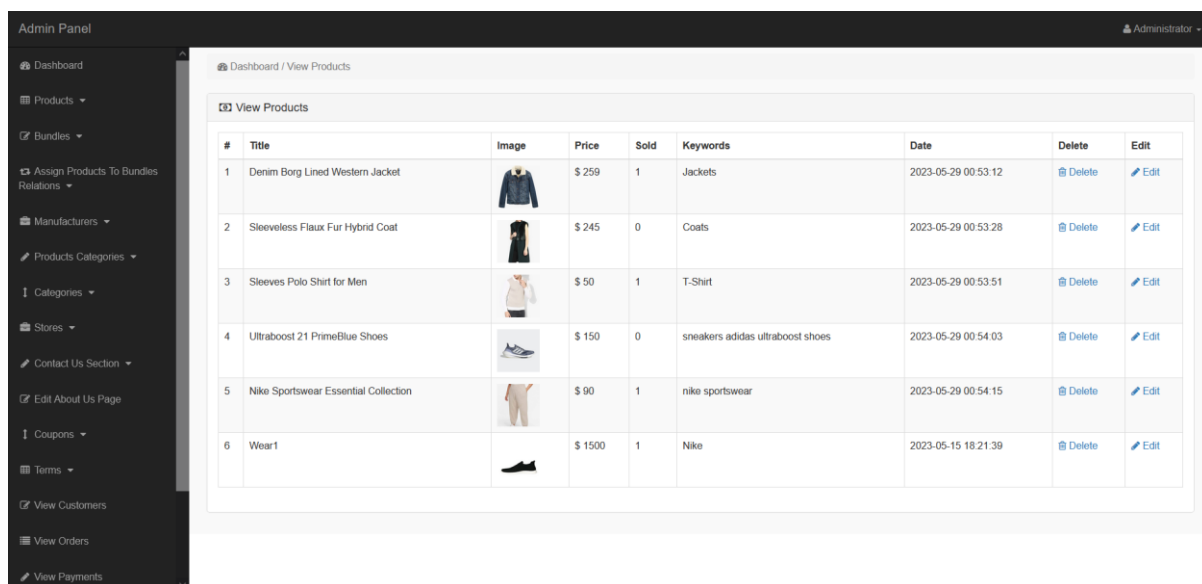
Також на головній дошці знаходиться інформація про нові замовлення від користувачів.

Найважливішим елементом адмін панелі є виконання операцій з товарами. Адміністратор може видаляти, додавати, та змінювати опис товарів. На рисунках 3.45 – 3.46 зображено вигляд сторінок для перегляду та додавання товарів.



The screenshot shows the 'Admin Panel' interface for adding a new product. The left sidebar contains navigation options like Dashboard, Products, Bundles, Assign Products To Bundles Relations, Manufacturers, Products Categories, Categories, Stores, Contact Us Section, Edit About Us Page, Coupons, Terms, View Customers, View Orders, and View Payments. The main content area is titled 'Dashboard / Insert Products' and contains the 'Insert Products' form. The form includes fields for Product Title, Product Url, Product Category, Product Price, and Product Image. There are also tabs for Product Description, Product Features, and Sounds And Videos.

Рисунок 3.45 – Сторінка додавання товару



The screenshot shows the 'Admin Panel' interface for viewing products. The left sidebar is the same as in the previous screenshot. The main content area is titled 'Dashboard / View Products' and contains a table of products. The table has columns for #, Title, Image, Price, Sold, Keywords, Date, Delete, and Edit.







#	Title	Image	Price	Sold	Keywords	Date	Delete	Edit
1	Denim Borg Lined Western Jacket		\$ 259	1	Jackets	2023-05-29 00:53:12	Delete	Edit
2	Sleeveless Flaux Fur Hybrid Coat		\$ 245	0	Coats	2023-05-29 00:53:28	Delete	Edit
3	Sleeves Polo Shirt for Men		\$ 50	1	T-Shirt	2023-05-29 00:53:51	Delete	Edit
4	Ultraboost 21 PrimeBlue Shoes		\$ 150	0	sneakers adidas ultraboost shoes	2023-05-29 00:54:03	Delete	Edit
5	Nike Sportswear Essential Collection		\$ 90	1	nike sportswear	2023-05-29 00:54:15	Delete	Edit
6	Wear1		\$ 1500	1	Nike	2023-05-15 18:21:39	Delete	Edit

Рисунок 3.46 – Сторінка перегляду товару

Якщо адміністратор обере операцію редагування товару то його буде перенаправлено на заповнену форму, в якій він зможе змінити дані. Сторінку редагування товару представлено на рисунку 3.47.

The screenshot shows the 'Edit Products' form in the Admin Panel. The form is titled 'Edit Products' and is located under the 'Dashboard / Edit Products' path. The form contains the following fields:

- Product Title:** Nike Sportswear Essential Collection
- Product Uri:** nike-sportswear-essen-col
- Product Uri Example:** navy-blue-t-shirt
- Select A Manufacturer:** Nike
- Product Category:** Trousers
- Category:** Feminine
- Product Image 1:** Обзор... файл не выбран. (Image placeholder)
- Product Image 2:** Обзор... файл не выбран. (Image placeholder)
- Product Image 3:** Обзор... файл не выбран. (Image placeholder)

Рисунок 3.47 – Сторінка редагування товару

Також продукти можна групувати в комплекти, або створювати зв'язки між ними. Це надає користувачу більш чіткого розуміння про те, який стиль одягу йому необхідно обрати. На рисунках 3.48 – 3.49 представлено форми для створення зв'язків та комплектів, та результат створення зв'язків.

The top screenshot shows the 'Insert Relation' form in the Admin Panel. The form is titled 'Insert Relation' and is located under the 'Dashboard / Insert Relation' path. The form contains the following fields:

- Relation Title:** Relation1
- Select Product:** Denim Borg Lined Western Jacket
- Select Bundle:** Gildan 1800 Ultra Cotton Polo Shirt

The bottom screenshot shows the 'View Relations' table in the Admin Panel. The table is titled 'View Relations' and is located under the 'Dashboard / View Relations' path. The table contains one relation:

#	Title	Product	Bundle	Delete	Edit
1	Relation1	Denim Borg Lined Western Jacket	Gildan 1800 Ultra Cotton Polo Shirt	Delete	Edit

Рисунок 3.48 – Приклад та результат створення зв'язків між товарами

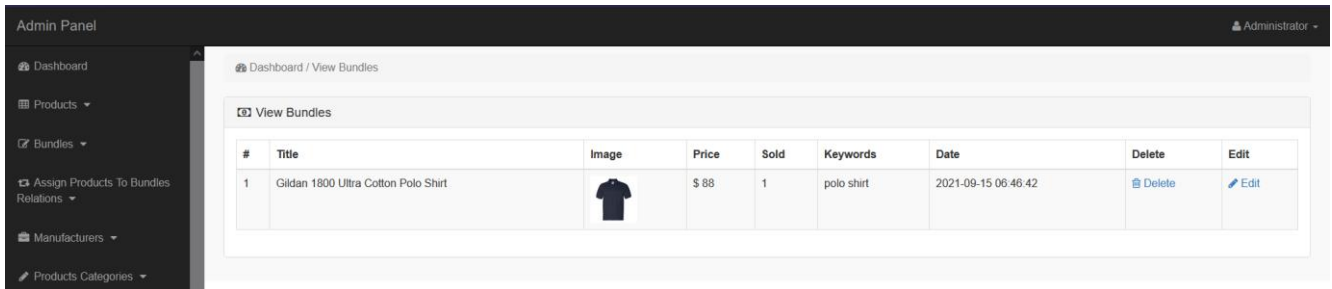
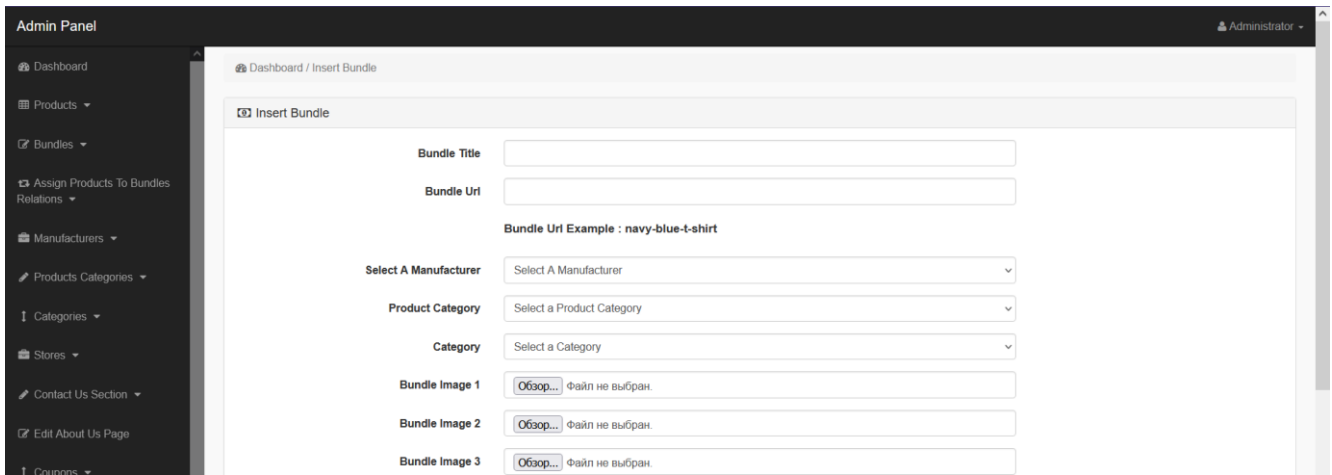


Рисунок 3.49 – Приклад та результат створення комплекту товарів

Для більш чіткого опису товару було створено опцію додавання виробника. Кожен виробник може бути доданий до опису товару. На рисунку 3.50 представлено вигляд форми для додавання виробника та перегляд результату додавання.

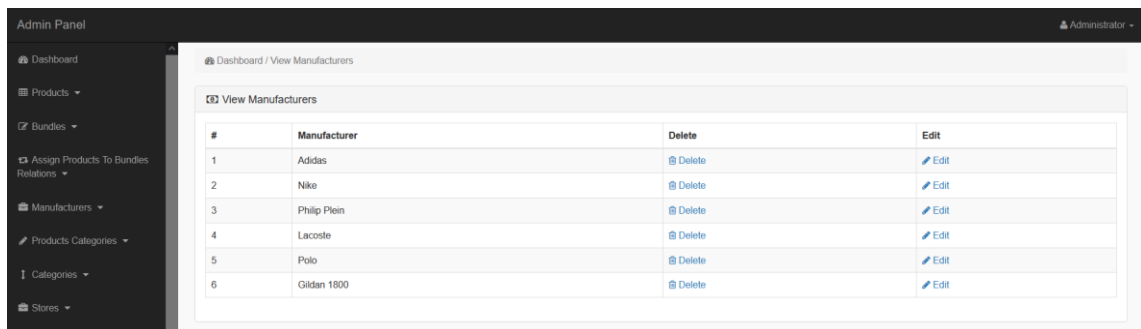
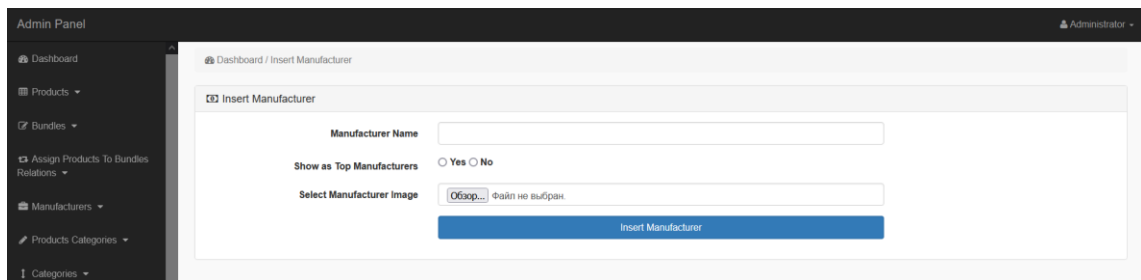


Рисунок 3.50 – Вигляд форми та результат додавання виробників

Також кожний товар має відповідну категорію. Адміністративна панель надає можливість додавати та видаляти категорії товарів. На рисунку 3.51 представлено результат додавання та перегляду категорій.

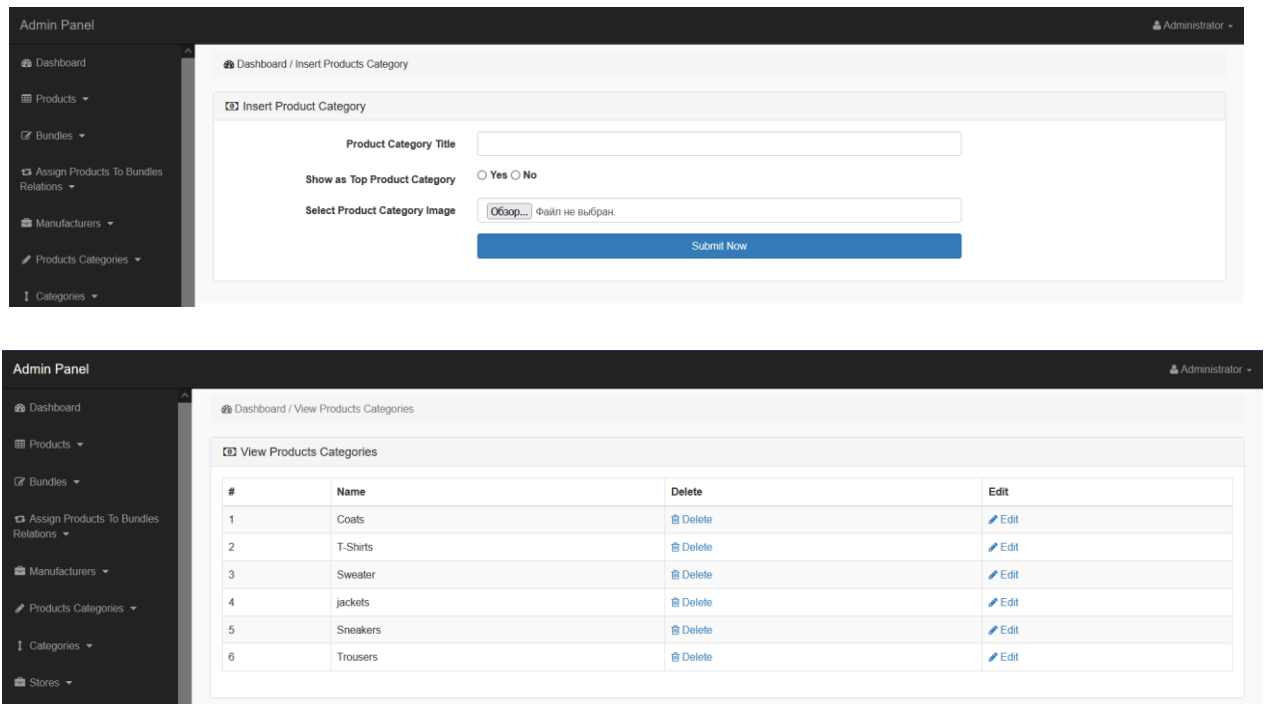


Рисунок 3.51 – Результат додавання та перегляду категорій

Для користувача важливо знати правила користування. Тому для цього було створено опцію додавання, редагування та перегляду правил. На рисунку 3.52 представлено приклад додавання та перегляду правил.

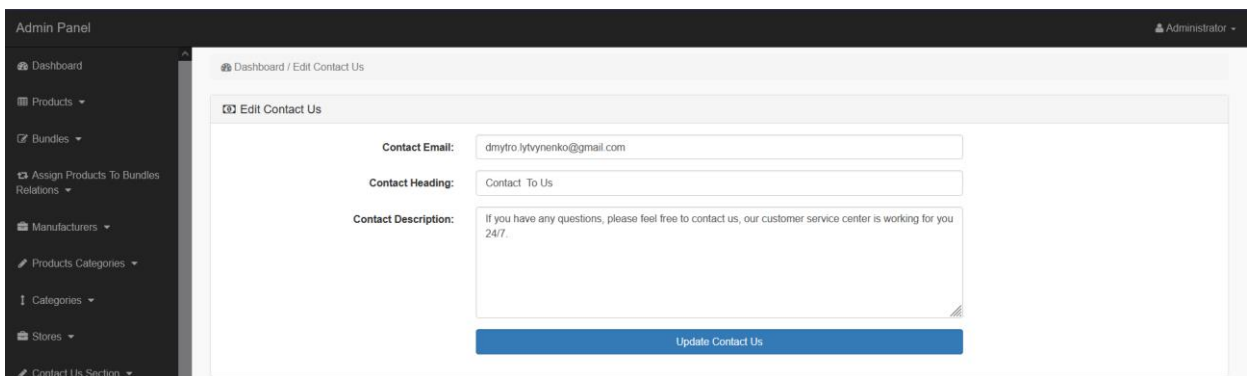




Рисунок 3.52 – Приклад додавання та перегляду правил

Для додаткових знижок було створено систему купонів. Адміністратор за допомогою форми зможе обрати товар, до якого буде застосований купон та надати відсоток знижки. На рисунку 3.53 представлено результат додавання та перегляду купонів.

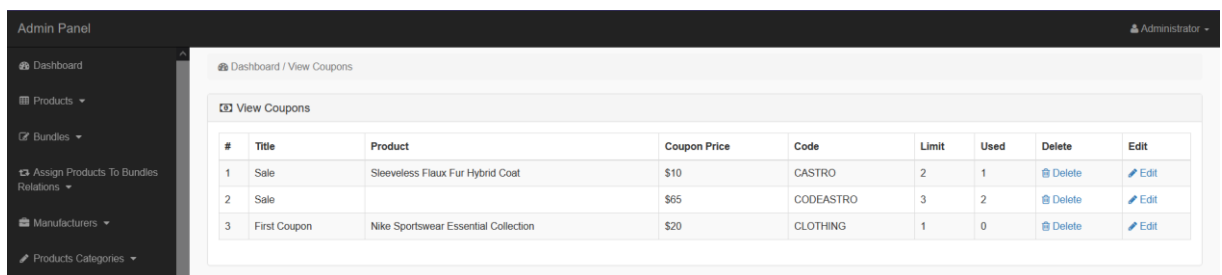
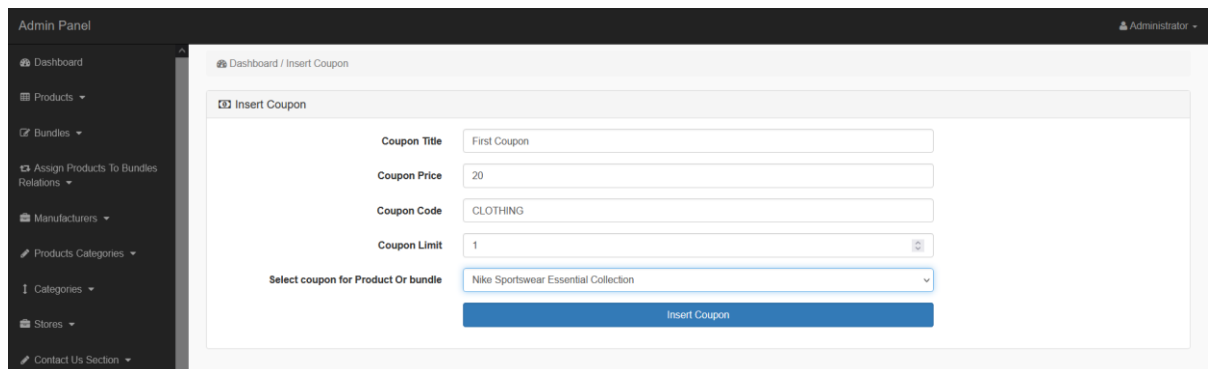


Рисунок 3.53 – Результат додавання та перегляду купонів

Також адміністративна панель надає можливість переглядати зареєстрованих користувачів та видаляти їх. На рисунку 3.54 зображено вигляд сторінки «View Customers».

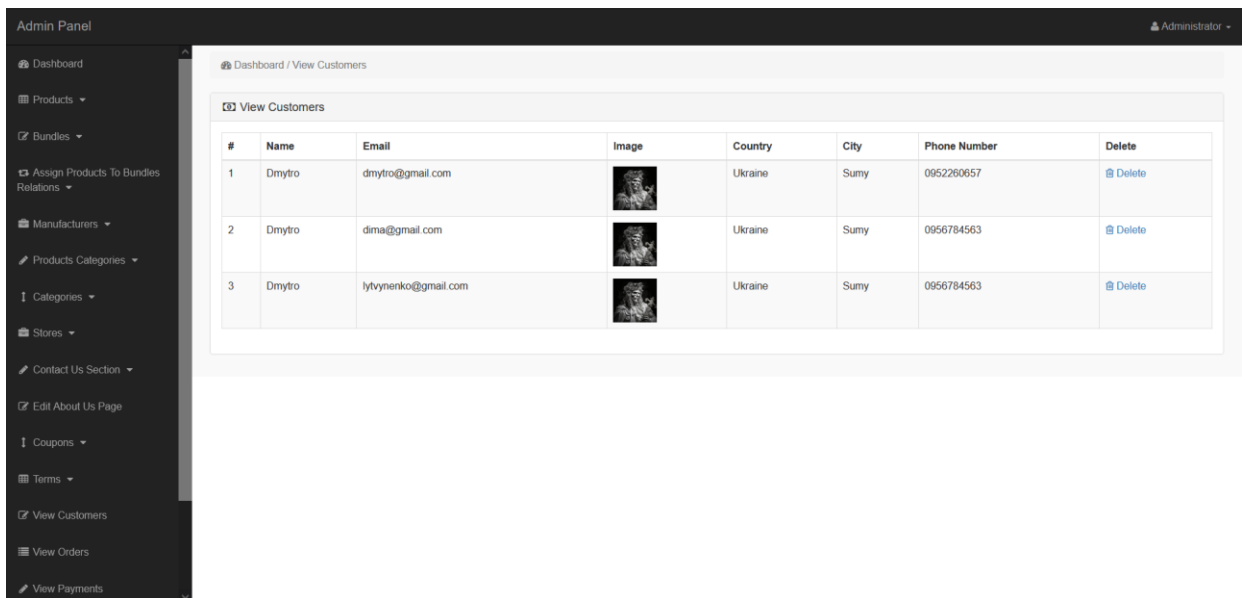


Рисунок 3.54 – Видгляд сторінки «View Customers»

Адміністратору надається можливість надавати права адміністратора іншим користувачам для полегшення власної роботи. Приклад додавання адміністратора зображено на рисунку 3.55.

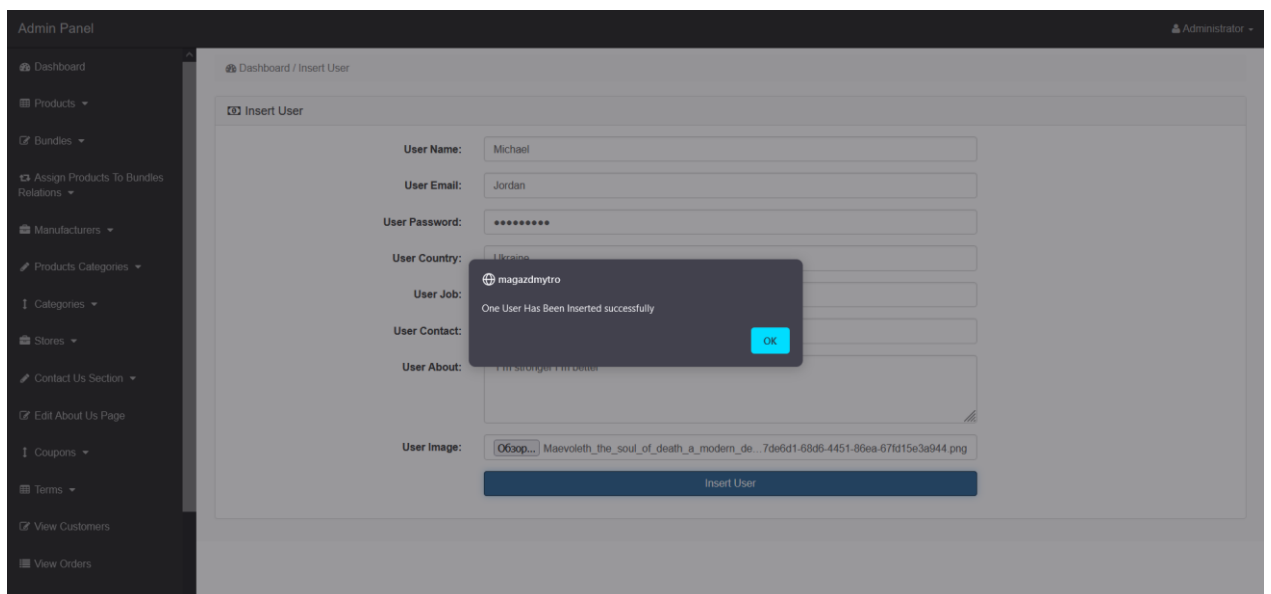


Рисунок 3.55 – Результат заповнення форми додавання нового адміністратора

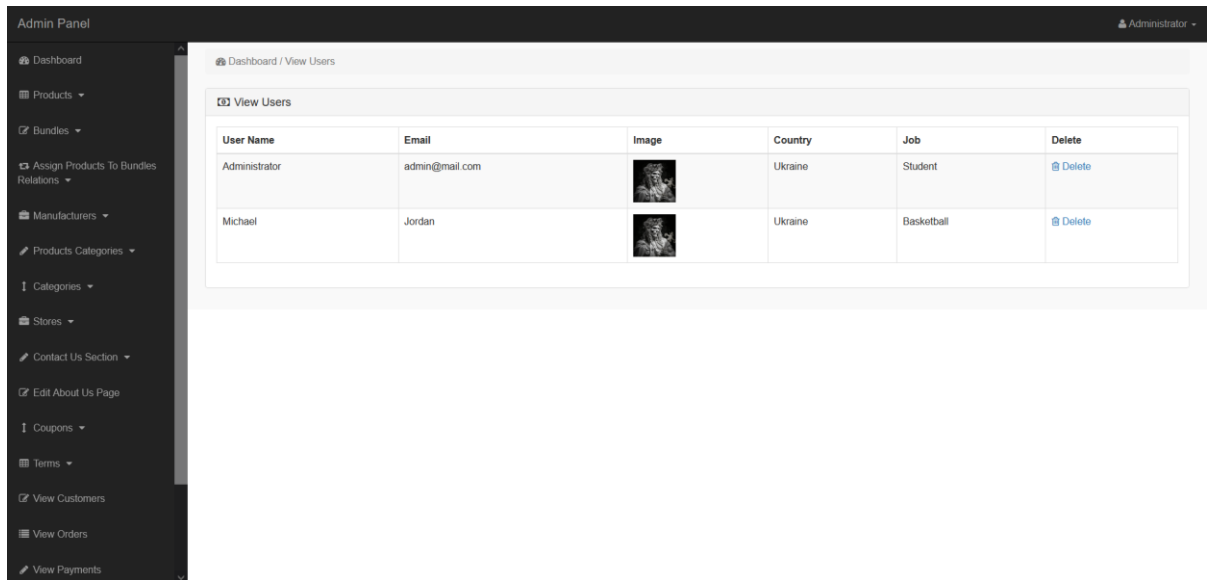


Рисунок 3.56 – Вигляд сторінки «Users»

ВИСНОВКИ

Під час виконання кваліфікаційної роботи бакалавра було досліджено предметну область, визначено актуальність та засоби для його реалізації.

Також було проаналізовано багато різних сайтів для продажу одягу, створено порівняльну таблицю та враховано їх недоліки.

У результаті було розроблено веб-додаток для підтримки діяльності магазину одягу. Головна мета створення веб-додатку полягає в тому, щоб залучити більше аудиторії, яка купує одяг та підвищити ефективність продаж за допомогою додатку.

Веб-додаток демонструє наскільки інформаційні технології здатні спростити сучасне життя. Він автоматизує рутинний процес замовлення одягу.

Останнім етапом дипломного проектування було успішне тестування розробленого веб-додатку у результаті якого суттєвих недоліків виявлено не було.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- 1 Ecommerce Statistics 2023 – Everything You Need To Know. URL: <https://www.searchlogistics.com/learn/statistics/ecommerce-statistics/> (дата звернення 01.05.2023).
- 2 Global Online Sales Projections. URL: <https://www.directlink.com/global-e-commerce-sales-projections/> (дата звернення 01.05.2023).
- 3 Магазин garne.com.ua URL: <https://garne.com.ua/> (дата звернення 02.05.2023).
- 4 Магазин cultboutique.com.ua URL: <https://cultboutique.com.ua/ua> (дата звернення 02.05.2023).
- 5 Магазин kasta.ua URL: <https://kasta.ua/uk/> (дата звернення 02.05.2023).
- 6 Which methodology is best for веб development? URL: <https://rootstack.com/en/blog/which-methodology-best-веб-development> (дата звернення 03.05.2023).
- 7 Create IDEF0 diagrams URL: <https://support.microsoft.com/en-us/office/create-idef0-diagrams-ea7a9289-96e0-4df8-bb26-a62ea86417fc> (дата звернення 03.05.2023).
- 8 What is a UML Diagram? URL: <https://www.smartdraw.com/uml-diagram/> (дата звернення 03.05.2023).
- 9 All You Need to Know About UML Diagrams: Types and 5+ Examples URL: <https://tallyfy.com/uml-diagram/> (дата звернення 04.05.2023).
- 10 Relational database. What is a relational database? URL: <https://www.techtarget.com/searchdatamanagement/definition/relational-database> (дата звернення 04.05.2023).
- 11 What is a High Level Design (HLD)? URL: <https://ipwithease.com/what-is-a-high-level-design-hld/> (дата звернення 04.05.2023).
- 12 7 Best IDE for Веб Development URL: <https://intellipaat.com/blog/best-ide-for-веб-development/?US> (дата звернення 05.05.2023).

13 Офіційна сторінка IDE Visual Studio Code URL: <https://code.visualstudio.com/> (дата звернення 05.05.2023).

14 Bootstrap: official site URL: <https://getbootstrap.com/> (дата звернення 06.05.2023).

15 Постановка цілей по SMART [Електронний ресурс] – Режим доступу до ресурсу: <https://worksection.com/ua/blog/kak-stavyt-zadachy.html> (дата звернення 03.05.2023).

16 Що таке WBS? Повне керівництво по структурі роботи [Електронний ресурс] – Режим доступу до ресурсу: <http://naukam.triada.in.ua/index.php/konferentsiji/45-p-yatnadtsyata-vseukrajinska-praktichno-piznavalna-internet-konferentsiya/306-pidkhodi-do-pobudovi-ierarkhichnoji-strukturi-robit-proektu-wbs-ta-obgruntuvannya-viboru-na-prikladi-proektu-stvorennya-sokovogo-baru-dnipro-the-juice> (дата звернення 04.05.2023).

17 Організаційна структура управління: типи і характерні особливості [Електронний ресурс] – Режим доступу до ресурсу: https://pidru4niki.com/15880315/menedzhment/organizatsiyana_struktura_upravlinnya_tipi_harakterni_osoblivosti (дата звернення 04.05.2023).

ДОДАТОК А

**Технічне завдання
на створення програмного продукту
«Веб-орієнтована система для забезпечення інформаційного
обслуговування діяльності магазину одягу»**

1. Призначення й мета створення веб-додатку

1.1 Призначення веб-додатку

Веб-додаток повинен забезпечити ефективне та автоматизоване інформаційне обслуговування магазину одягу, спростити процеси управління запасами, замовленнями, продажами та іншими аспектами діяльності магазину.

1.2 Мета створення веб-додатку

Основною метою створення веб-додатку є поліпшення ефективності та точності обробки даних, оптимізація процесів замовлення та продажу одягу, забезпечення зручного інтерфейсу для клієнтів та персоналу магазину, а також покращення загальної якості обслуговування.

1.3 Цільова аудиторія

До основної аудиторії веб-додатку можна віднести будь-кого, хто бажає придбати певний товар, який відноситься до категорії одягу. Також головною складовою цільової аудиторії є працівники, які підтримують діяльність веб-додатку, та інші особи, які зацікавлені в користуванні додатком.

2 Вимоги до веб-орієнтованої системи

2.1 Вимоги до веб-системи в цілому

2.1.1 Вимоги до структури й функціонування веб-орієнтованої системи

Інформаційна система має бути реалізована у вигляді вебсистеми, що буде складатися з декількох вебсторінок з окремим інформаційним та функціональним наповненням, а також з підключенням бази даних.

2.1.2 Вимоги до персоналу

Для успішного використання веб-додатку персонал магазину повинен мати базові навички роботи з комп'ютером та Інтернетом. Рекомендується,

щоб персонал був знайомий з основними принципами електронної комерції та роботи з веб-додатками.

2.1.3 Вимоги до збереження інформації

Система повинна забезпечувати безпечне зберігання та обробку всієї інформації, пов'язаної з діяльністю магазину, включаючи дані про клієнтів, товари, замовлення, оплати та інші деталі. Забезпечення конфіденційності та захисту персональних даних клієнтів є важливим пріоритетом, тому система повинна відповідати сучасним стандартам безпеки даних.

2.1.4 Вимоги до розмежування доступу

Система повинна забезпечувати різні рівні доступу для різних користувачів, включаючи адміністраторів та клієнтів магазину одягу. Кожному користувачеві повинні надаватися певні права доступу відповідно до його ролі та обов'язків. Необхідно врахувати забезпечення автентифікації та авторизації користувачів для забезпечення безпеки інформації та обмеження несанкціонованого доступу до системи.

2.2 Структура веб-додатку

2.2.1 Загальна інформація про структуру веб-додатку

Веб-додаток має складатися з наступних розділів:

Головна – статична (коротка інформація про сайт та таблиця з товарами, відгуками, та контактною інформацією) або динамічна (вивід нових матеріалів).

Про нас – на даній сторінці користувач може отримати інформацію про сайт.

Сторінка керування адміністративною панеллю - адміністратор може додавати, видаляти або редагувати користувачів системи, товари, на інформаційне наповнення додатку.

2.2.2 Навігація

Інтерфейс повинен забезпечувати зручне та інтуїтивно зрозуміле відображення структури інформації, швидкий та логічний перехід між розділами та сторінками. Навігаційні елементи повинні бути зрозумілими щодо свого змісту та способу взаємодії з ними. Усі елементи повинні мати чітку підписку та використовувати загальноприйняті символи. Посилання на сторінки повинні мати заголовки, а графічні елементи повинні мати альтернативний текстовий опис.

Система повинна забезпечувати навігацію по всім доступним ресурсам та коректне відображення інформації. Для навігації використовується контент-меню, яке представляє собою блок гіперпосилань у верхній або лівій частині веб-сторінки, відповідно до затвердженого дизайну.

2.2.3 Наповнення веб-додатку (контент)

Веб-додаток повинен містити головну сторінку, яка надає загальну інформацію про магазин одягу, його продукти та послуги. Також повинна бути можливість перегляду та пошуку товарів за різними критеріями, такими як категорія, бренд, розмір тощо та повинен містити можливість створення та керування обліковими записами користувачів, включаючи реєстрацію, вхід у систему та відновлення паролю. Веб-додаток повинен надавати можливість здійснення онлайн-замовлень, включаючи додавання товарів до кошика, введення інформації для доставки та здійснення оплати.

2.2.4 Дизайн та структура додатку

Необхідно розробити інтуїтивно зрозумілий та привабливий користувальницький інтерфейс, що відобразить основні функціональні можливості системи і забезпечуватиме зручний доступ до них для користувачів. Також необхідно визначити основних модулів та компонентів системи, такі як модуль каталогу товарів, модуль керування замовленнями, модуль розрахунків тощо. Забезпечити належний розподіл функціональності між модулями для ефективної роботи системи.

2.2.5 Функціональні вимоги

Вимоги до функціональних характеристик:

- авторизація користувачів;
- перехід на інші веб-сторінки за допомогою контент-меню;
- виконувати вхід на сайт як зареєстрований користувач для можливості оформлення замовлення та відслідковування замовлень;
- можливість додавати обраний товар до кошику;
- можливість оформлювати замовлення;
- можливість надіслати повідомлення адміністратору;
- можливість змінювати особисту інформацію в особистому кабінеті;

3 Порядок виконання робіт

- Вивчення бізнес-процесів магазину з продажу одягу, включаючи управління запасами та замовленнями.
- Збір вимог від замовника, уточнення деталей та встановлення основних функціональних та нефункціональних вимог до веб-додатку.
- Визначення технологій, які будуть використовуватися для реалізації веб-додатку.
- Створення структурної схеми веб-додатку, включаючи визначення потрібних модулів та їх взаємозв'язків.
- Розробка архітектури бази даних для зберігання інформації про товари, замовлення та інші дані, необхідні для роботи магазину.
- Реалізація серверної частини веб-додатку, включаючи логіку бізнес-процесів, обробку запитів та взаємодію з базою даних.
- Передача веб-додатку замовнику та надання необхідної документації для користування та підтримки системи.

3.1 Склад і зміст робіт зі створення веб-сайту

Докладний опис етапів створення веб-сайту наведено в таблиці 1

Таблиця А.1 – Етапи створення сайту

№	Склад і зміст роботи	Строк розробки (у робочих днях)
1	Розробка прототипу	4 днів
2	Розробка бази даних	3 днів
3	Розробка функціонала	3 днів
4	Зв'язування веб-сайту з БД	1 дня
5	Розробка сторінок	3 днів
6	Розробка дизайну шаблону	5 днів
7	Тестування та корекція помилок	2 днів
Загальна тривалість робіт і строк закінчення проєкту		21 днів

4 Вимоги до складу й змісту робіт із введення веб-додатку в експлуатацію

Розробка веб-додатку повинна бути здійснена відповідно до встановлених термінів та бюджету проєкту. Перед введенням веб-додатку в експлуатацію, необхідно здійснити тестування та усунення можливих помилок. Веб-додаток повинен бути документований, включаючи пояснення до функціональності, архітектури та інструкції з використання. Після введення веб-додатку в експлуатацію, необхідно забезпечити підтримку та обслуговування, включаючи виправлення помилок, розширення функціональності та забезпечення безпеки системи.

ДОДАТОК Б

Планування робіт

Для досягнення мети проєкту необхідно виконати наступні задачі:

- визначити актуальність роботи, дослідити предметну область та провести аналіз аналогів веб-додатків для підтримки роботи магазину з продажу одягу;
- виконати розробку веб-додатку;
- розробити та реалізувати компоненти для взаємодії додатку з клієнтом;

Деталізація мети проєкту методом SMART

SMART метод вже відома багатьом, але все ще надто рідко реалізується. Це видно з того, що деякі завдання та проєкти приречені на провал ще до того, як їх фактично почали. Причина звучить просто, але наслідки можуть бути серйозними. Цілі встановлені неправильно і не можуть бути досягнуті з самого початку. У цьому допомагає метод SMART. Необхідно правильно поставити цілі, щоб мати на увазі точну картину бажаного результату. Для того, щоб це працювало, цільова рецептура повинна відповідати ряду критеріїв [15].

Отже, сформульована мета проєкту за критеріями SMART наведена у таблиці 1.1.

Specific	Розробка веб-додатку для підтримки діяльності магазину одягу
Measurable	Результат використання передових технологій веб-додатку для продажу одягу сприятиме ефективному використанню ресурсів, таких як кадри та матеріали.

Achievable	Реалізація web-орієнтованої системи здійснюється за допомогою ручної верстки сайту з використанням HTML, CSS, JS, PHP, MySQL
Revelant	Усі необхідні технічні та програмні засоби наявні для успішної реалізації проекту, а команда розробників володіє достатнім рівнем кваліфікації для виконання поставлених завдань.
Time-bound	Проект реалізується з урахуванням зазначених обмежень в часі, які були встановлені на початковому етапі і становлять червень 2023 року.

Розробка веб-додатків вимагає систематичного та структурованого підходу, а для цього використовуються такі методи як Work Breakdown Structure (WBS) та Organizational Breakdown Structure (OBS). Ці дві структури мають важливе значення для успішної розробки web-додатків і допомагають забезпечити ефективне керівництво проектом та розподіл робіт між командами.

Work Breakdown Structure (WBS) - це ієрархічна структура, що деталізує проект на більш малі компоненти або робочі пакети. WBS дозволяє розбити складний проект на менші, керовані елементи, що сприяє кращому розумінню завдань, обсягу робіт та встановленню чітких меж для кожного елемента. У контексті розробки web-додатків, WBS дозволяє ідентифікувати та організувати різні функціональні блоки, такі як інтерфейс користувача, база даних, логіка додатку, безпека тощо. Це сприяє управлінню ресурсами, плануванню та визначенню пріоритетів робіт, а також полегшує контроль за прогресом проекту [16].

Organizational Breakdown Structure (OBS) - це структура, що відображає організаційну структуру команди розробки проекту. OBS визначає ролі та відповідальності учасників проекту, їхній рівень авторитету та ієрархію. У випадку розробки web-додатків, OBS дозволяє ясно визначити команду

розробників, аналітиків, дизайнерів, тестувальників та інших учасників проекту, а також встановити зв'язки та комунікацію між ними. Це сприяє розподілу робіт, визначенню ролей та відповідальностей, ефективному використанню ресурсів та забезпеченню координації між учасниками команди [17].

На рисунках Б.1 представлено WBS проекту щодо розробки сайту для підтримки роботи магазину одягу.

На рисунку Б.2 представлено організаційну структуру планування проекту. Список виконавців, що функціонують в проекті описано в таблиці Б.2.

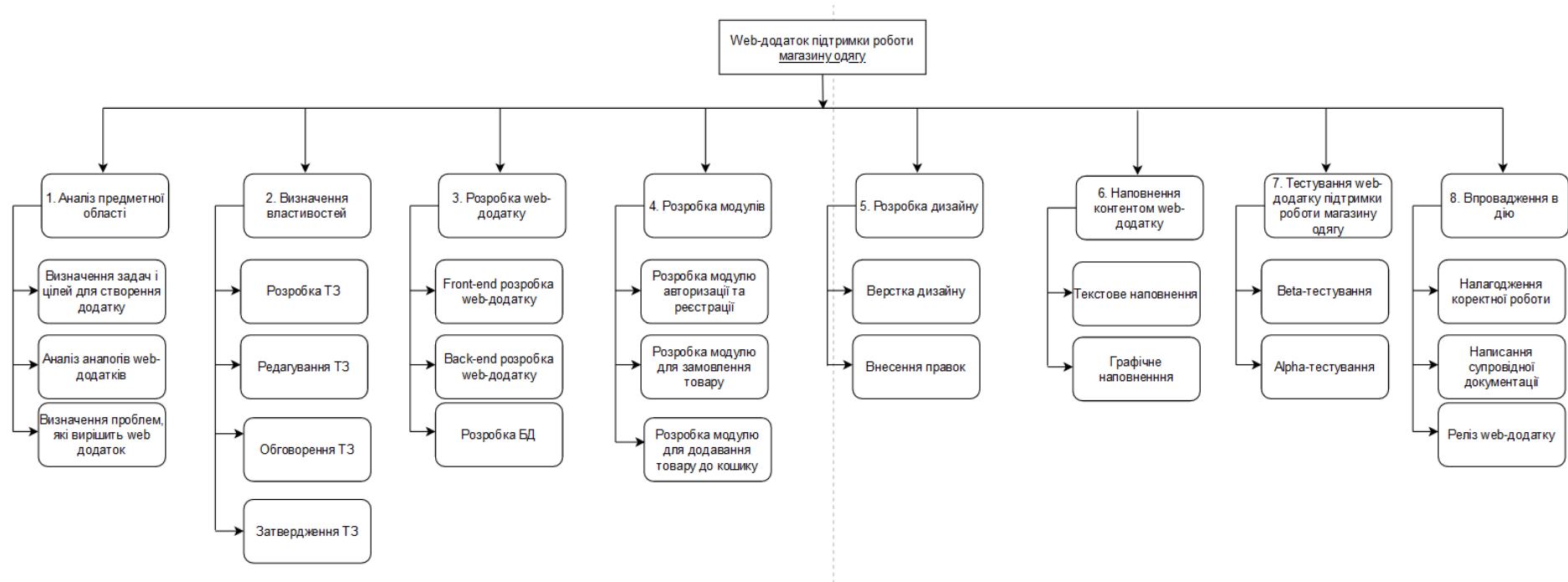


Рисунок Б.1 – WBS-структура робіт проєкту

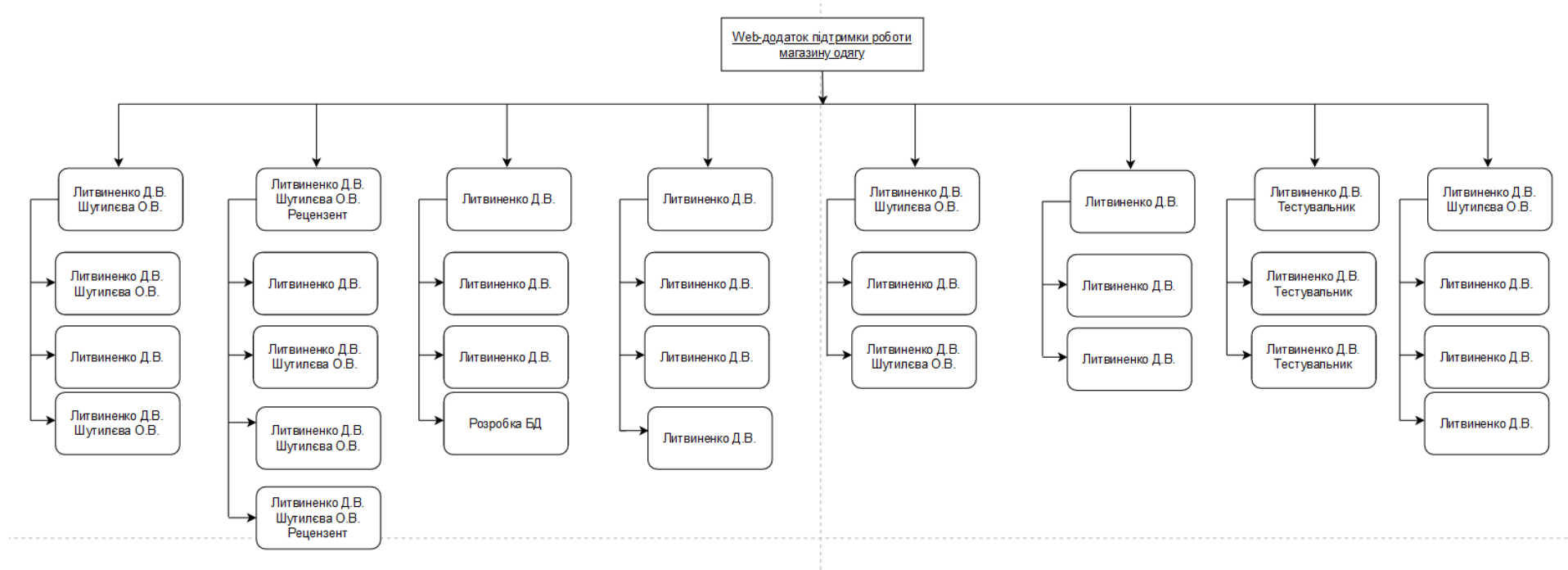


Рисунок Б.2 – OBS-структура робіт проект

Таблиця Б.2 – Виконавці проєкту

Роль	Ім'я	Проектна роль
Керівник проєкту	Шутилева О.В.	Ця особа відповідає за дотримання встановлених термінів, ефективний розподіл ресурсів та призначення завдань. Вона також займається збором та аналізом даних.
Дизайнер	Литвиненко Д.В.	Відповідає створення дизайну сайту.
Web розробник	Литвиненко Д.В.	Відповідає за front-end та back-end розробку.
Адміністратор БД	Литвиненко Д.В.	Виконує проєктування бази даних.
Локалізатор	Литвиненко Д.В.	Відповідає за локалізацію сайту
Тестувальник	Тестувальник	Знаходження помилок до релізу додатку.

Діаграма Ганта

Планування проєкту включає важливий етап - побудову календарного графіку або діаграми Ганта. Цей графік представляє собою розклад виконання робіт з врахуванням реального розподілу дат. Він дозволяє отримати достовірне уявлення про тривалість процесів, враховуючи обмеження у ресурсах, вихідні дні та святкові дні.

Побудова календарного графіку дозволяє зорієнтуватися в хронологічному виконанні завдань, визначити критичні шляхи та залежності між ними. Це допомагає забезпечити ефективне планування ресурсів, розподіл завдань між

командою та визначення термінів виконання. Календарний графік є інструментом, що сприяє управлінню проектом та забезпечує контроль над його ходом.

Календарний графік проекту представлено на рисунках Б.3-Б.4.

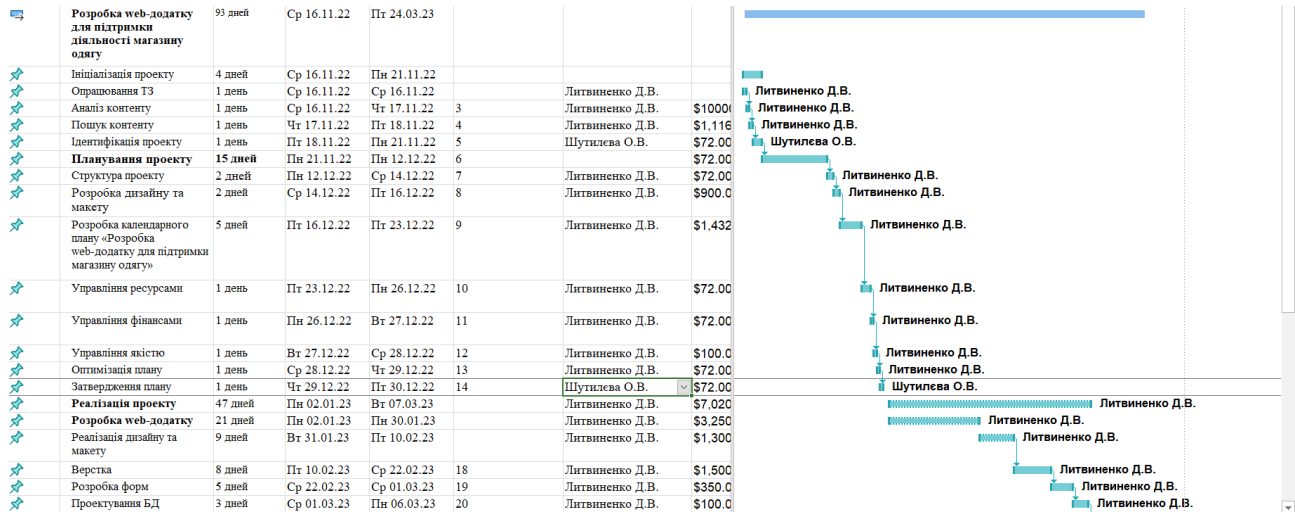


Рисунок Б.3 – Діаграма Ганта. Частина 1

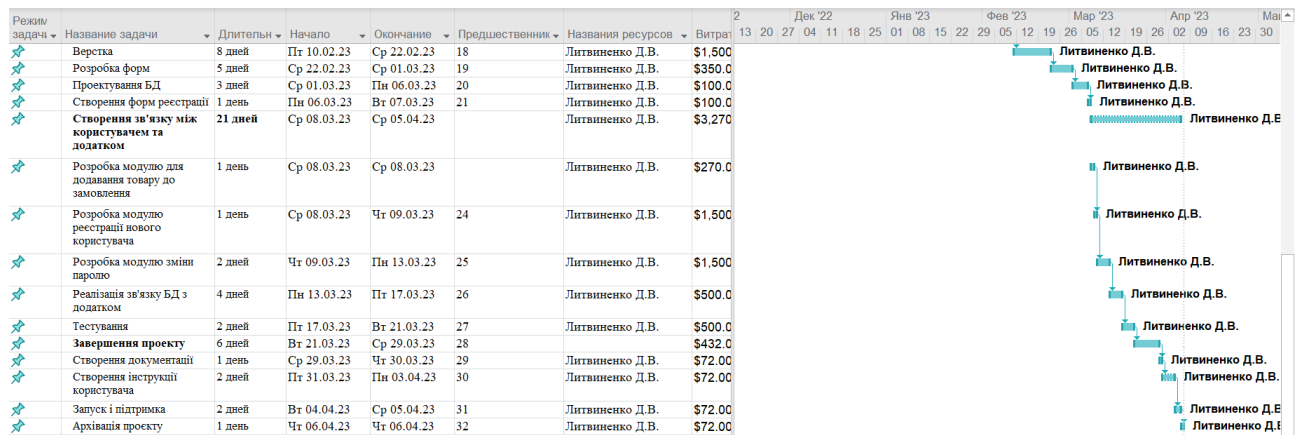


Рисунок Б.4 – Діаграма Ганта. Частина 2

Управління ризиками проекту

Під час проведення якісної оцінки ризиків, необхідно ідентифікувати ті ризики, які потребують негайного врегулювання. Реагування на ризики буде залежати від їх важливості. Далі слід провести кількісну оцінку ризиків. Кількісна та якісна оцінка можуть проводитися окремо або одночасно, в

залежності від особливостей проєкту. У таблиці Б.3 наведено шкалу для класифікації ризиків за їх впливом на проєкт та ймовірністю виникнення.

Під час виконання оцінки ризиків важливо враховувати, що ризики можуть мати різну суттєвість та можливість настання. Процес виявлення, аналізу та оцінки ризиків є важливою складовою ефективного управління проєктом. Це дозволяє забезпечити розуміння й управління потенційними проблемами, що можуть виникнути під час реалізації проєкту.

З метою зниження негативних наслідків ризиків для проєкту необхідно розробити план дій щодо їх управління. Цей план включає аналіз ефективності розробки та оцінку впливу ризиків на проєкт. Оцінка проводиться згідно з критеріями, описаними у таблиці Б.3. В результаті планування реагування була створена матриця ймовірності виникнення ризиків та їх впливу, яка зображена на рисунку Б.4. Зеленим кольором позначаються прийнятні ризики, жовтим – ті, які вимагають вжиття заходів, а червоним – неприйнятні ризики. Додатково, важливим аспектом є постійний моніторинг ризиків та вчасне реагування на них, що дозволяє запобігти можливим проблемам та забезпечити успішну реалізацію проєкту.

3	IMPACT	RS_2						RS_4 RS_5	
2		RS_13		RS_14 RS_9 RS_3					
1		RS_12		RS_8 RS_11 RS_15			RS_6 RS_7 RS_10 RS_1		
		Probability							
		1		2				3	

Рисунок Б.4. – Матриця ймовірності

Згідно з результатами аналізу, ризики були розподілені за рівнями у таблиці Б.4. Крім того, у таблиці Б.5 наведено детальний опис кожного ризику, а також визначено рекомендовані стратегії реагування для кожного випадку.

Важливим кроком у процесі управління ризиками є адекватне класифікування та аналіз ризиків, що дозволяє визначити належні заходи для їх управління та мінімізації впливу на проєкт.

Таблиця Б.4 – Шкала оцінювання за рівнем ризику.

№	Назва	Межі	Ризики, які входять
1	Прийнятні	$1 < R < 2$	8,11,12,13,15
2	Виправдані	$3 < R < 4$	1,2,3,6,7,9,10,14
3	Недопустимі	$6 < R < 9$	4,5

Таблиця Б.5 – Ризики та стратегії реагування

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS_1	Відкритий	Затримка в поставці необхідних компонентів або інфраструктури	Низька	Середній	3	Розробити альтернативний план поставок та підготувати запасні постачальники.	Попередження	Визначити можливості для прискорення процесу поставок.
RS_2	Відкритий	Технічні проблеми з сервером або хостингом	Низька	Середній	4	Забезпечити наявність резервних серверів або хостингу.	Прийняття	Виконати регулярні резервні копії даних та планувати періодичні перевірки стану сервера.

RS_3	Відкритий	Вразливість системи безпеки, що може призвести до витоку конфіденційної інформації клієнтів.	Середня	Середній	4	Застосувати найсучасніші заходи безпеки, такі як шифрування даних, двофакторна аутентифікація, регулярні аудити безпеки та підготовка персоналу до виявлення і реагування на можливі загрози.	Пом'якшення	Врахувати всі можливі способи враження системи
RS_4	Відкритий	Недостатня функціональність або незадовільний дизайн додатку, що може вплинути на задоволення користувачів.	Середня	Високий	6	Виконати тестування користувачами для збору фідбеку та виявлення слабких місць.	Попередження	Вдосконалювати функціональність та дизайн на основі відгуків користувачів.

RS_5	Відкритий	Недостатній обсяг ресурсів для забезпечення підвищеного навантаження або пікового попиту.	Висока	Високий	9	Масштабувати інфраструктуру засобами хмарних послуг або залучити додаткові ресурси для забезпечення високої доступності та швидкодії додатку.	Пом'якшення	Знайти додатки які зможуть забезпечити оптимізацію та швидкодію додатку
RS_7	Відкритий	Конфлікти між розробниками або командами проекту, що можуть призвести до затримок у роботі.	Середня	Середній	4	Забезпечити ефективну комунікацію та співпрацю між учасниками проекту.	Пом'якшення	Вирішувати конфлікти шляхом налагодження взаєморозуміння та пошуку компромісів.

RS_8	Відкритий	Зміни вимог з боку замовника або непродуктивна співпраця з ним.	Низька	Середній	2	Установити чіткий процес управління змінами вимог та регулярно звітувати перед замовником.	Пом'якшення	Встановити ефективні комунікаційні канали та створити механізми для раннього виявлення та вирішення проблем.
RS_9	Відкритий	Відсутність достатньої експертизи або недосвідченість команди розробників.	Середня	Середній	4	Забезпечити професійний розвиток команди через навчання, тренінги та співпрацю з експертами.	Пом'якшення	Використовувати код-рев'ю та інші методи контролю якості розробки для запобігання помилкам.

RS_10	Відкритий	Відсутність чіткого плану розвитку та керівництва проектом.	Низька	Середній	3	Розробити детальний план проекту з визначеними метою, завданнями, термінами та відповідальними особами.	Попередження	Застосовувати методики проектного управління та регулярно оцінювати прогрес виконання.
RS_11	Відкритий	Несумісність додатку з різними браузерами та платформами.	Низька	Середній	2	Виконати тестування та оптимізацію додатку для різних браузерів та платформ.	Попередження	Визначити мінімальні вимоги до середовища, у якому додаток повинен працювати.
RS_12	Відкритий	Зміни у технологічних стандартах або вимоги до безпеки даних.	Низька	Низький	1	Постійно відстежувати зміни у технологічних стандартах та вимогах до безпеки даних.	Використання	Забезпечити оновлення та використання найсучасніших засобів та практик для забезпечення безпеки та відповідності стандартам.

RS_13	Відкритий	Втрата або пошкодження даних клієнтів.	Середня	Низький	2	Регулярно виконувати резервне копіювання та забезпечити високу надійність системи зберігання даних.	Перенесення	Встановити механізми відновлення даних в разі їх втрати або пошкодження.
RS_14	Відкритий	Незадовільна підтримка та обслуговування після випуску додатку.	Середня	Середня	4	Забезпечити належну підтримку клієнтів після випуску, включаючи швидке реагування на запити та вирішення проблем.	Прийняття	Запровадити систему збору фідбеку від користувачів та постійно вдосконалювати додаток на основі отриманих даних.

RS_15	Відкритий	Відсутність відповідності законодавчим вимогам щодо захисту персональних даних.	Низька	Середній	3	Детально вивчити вимоги законодавства щодо захисту персональних даних та впровадити відповідні заходи.	Пом'якшення	Провести аудит безпеки для впевненості у відповідності додатку вимогам.
-------	-----------	---	--------	----------	---	--	-------------	---

ДОДАТОК В

Лістинг коду function.php

```
<?php

$db = mysqli_connect("localhost","root","","ecom_store");

function getRealUserIp(){
    switch(true){
        case (!empty($_SERVER['HTTP_X_REAL_IP'])) : return
$_SERVER['HTTP_X_REAL_IP'];
        case (!empty($_SERVER['HTTP_CLIENT_IP'])) : return
$_SERVER['HTTP_CLIENT_IP'];
        case (!empty($_SERVER['HTTP_X_FORWARDED_FOR'])) : return
$_SERVER['HTTP_X_FORWARDED_FOR'];
        default : return $_SERVER['REMOTE_ADDR'];
    }
}

function items(){

global $db;

$ip_add = getRealUserIp();

$get_items = "select * from cart where ip_add='$ip_add'";

$run_items = mysqli_query($db,$get_items);

$count_items = mysqli_num_rows($run_items);

echo $count_items;

}
```

```
function total_price(){

global $db;

$ip_add = getRealUserIp();

$total = 0;

$select_cart = "select * from cart where ip_add='$ip_add'";

$run_cart = mysqli_query($db,$select_cart);

while($record=mysqli_fetch_array($run_cart)){

$pro_id = $record['p_id'];

$pro_qty = $record['qty'];

$sub_total = $record['p_price']*$pro_qty;

$total += $sub_total;

}

echo "$" . $total;

}

function getPro(){

global $db;

$get_products = "select * from products order by 1 DESC LIMIT

0,8";
```

```
$run_products = mysqli_query($db,$get_products);

while($row_products=mysqli_fetch_array($run_products)){

$pro_id = $row_products['product_id'];

$pro_title = $row_products['product_title'];

$pro_price = $row_products['product_price'];

$pro_img1 = $row_products['product_img1'];

$pro_label = $row_products['product_label'];

$manufacturer_id = $row_products['manufacturer_id'];

$get_manufacturer = "select * from manufacturers where
manufacturer_id='$manufacturer_id'";

$run_manufacturer = mysqli_query($db,$get_manufacturer);

$row_manufacturer = mysqli_fetch_array($run_manufacturer);

$manufacturer_name = $row_manufacturer['manufacturer_title'];

$pro_psp_price = $row_products['product_psp_price'];

$pro_url = $row_products['product_url'];

if($pro_label == "Sale" or $pro_label == "Gift"){

$product_price = "<del> $$pro_price </del>";

$product_psp_price = "| $$pro_psp_price";
```

```
}
else{

$product_psp_price = "";

$product_price = "$$pro_price";

}
if($pro_label == ""){
}
else{

$product_label = "

<a class='label sale' href='#' style='color:black;'>

<div class='thelabel'>$pro_label</div>

<div class='label-background'> </div>

</a>

";

}
echo "

<div class='col-md-4 col-sm-6 single' >

<div class='product' >

<a href='$pro_url' >

<img src='admin_area/product_images/$pro_img1' class='img-
responsive' >
```

```
</a>
<div class='text' >
<center>
<p class='btn btn-warning'> $manufacturer_name </p>
</center>
<hr>
<h3><a href='$pro_url' >$pro_title</a></h3>

<p class='price' > $product_price $product_psp_price </p>

<p class='buttons' >

<a href='$pro_url' class='btn btn-default' >View Details</a>

<a href='$pro_url' class='btn btn-danger'>

<i class='fa fa-shopping-cart'></i> Add To Cart

</a>

</p>

</div>

$product_label

</div>

</div>

";

}
```



```
}
```

```
function getProducts(){
```

```
global $db;
```

```
$aWhere = array();
```

```
if(isset($_REQUEST['man'])&&is_array($_REQUEST['man'])) {
```

```
foreach($_REQUEST['man'] as $sKey=>$sVal){
```

```
if((int)$sVal!=0){
```

```
$aWhere[] = 'manufacturer_id='.(int)$sVal;
```

```
}
```

```
}
```

```
}
```

```
if(isset($_REQUEST['p_cat'])&&is_array($_REQUEST['p_cat'])) {
```

```
foreach($_REQUEST['p_cat'] as $sKey=>$sVal){
```

```
if((int)$sVal!=0){
```

```
$aWhere[] = 'p_cat_id='.(int)$sVal;
```

```
}
```

```
}
```

```
}
```

```
if(isset($_REQUEST['cat'])&&is_array($_REQUEST['cat'])) {

foreach($_REQUEST['cat'] as $sKey=>$sVal) {

if((int)$sVal!=0) {

$aWhere[] = 'cat_id='.(int)$sVal;

}

}

}

$per_page=6;

if(isset($_GET['page'])) {

$page = $_GET['page'];

} else {

$page=1;

}

$start_from = ($page-1) * $per_page ;

$sLimit = " order by 1 DESC LIMIT $start_from,$per_page";

$aWhere = (count($aWhere)>0?' WHERE '.implode(' or ', $aWhere):'').$sLimit;

$get_products = "select * from products ".$aWhere;

$run_products = mysqli_query($db, $get_products);
```

```
while($row_products=mysqli_fetch_array($run_products)){

$pro_id = $row_products['product_id'];

$pro_title = $row_products['product_title'];

$pro_price = $row_products['product_price'];

$pro_img1 = $row_products['product_img1'];

$pro_label = $row_products['product_label'];

$manufacturer_id = $row_products['manufacturer_id'];

$get_manufacturer = "select * from manufacturers where
manufacturer_id='$manufacturer_id'";

$run_manufacturer = mysqli_query($db,$get_manufacturer);

$row_manufacturer = mysqli_fetch_array($run_manufacturer);

$manufacturer_name = $row_manufacturer['manufacturer_title'];

$pro_psp_price = $row_products['product_psp_price'];

$pro_url = $row_products['product_url'];

if($pro_label == "Sale" or $pro_label == "Gift"){

$product_price = "<del> $$pro_price </del>";

$product_psp_price = "| $$pro_psp_price";

}

else{
```

```
$product_psp_price = "";

$product_price = "$$pro_price";

}
if($pro_label == ""){
}
else{

$product_label = "

<a class='label sale' href='#' style='color:black;'>

<div class='thelabel'>$pro_label</div>

<div class='label-background'> </div>

</a>

";

}
echo "

<div class='col-md-4 col-sm-6 center-responsive' >

<div class='product' >

<a href='$pro_url' >

<img src='admin_area/product_images/$pro_img1' class='img-
responsive' >

</a>
```

```
<div class='text' >

<center>

<p class='btn btn-warning'> $manufacturer_name </p>

</center>

<hr>

<h3><a href='$pro_url' >$pro_title</a></h3>

<p class='price' > $product_price $product_psp_price </p>

<p class='buttons' >

<a href='$pro_url' class='btn btn-default' >View details</a>

<a href='$pro_url' class='btn btn-danger'>

<i class='fa fa-shopping-cart' data-price=$pro_price></i> Add To
Cart

</a>
</p>

</div>

$product_label

</div>

</div>

";
```

```
}
```

```
function getPaginator(){
```

```
$per_page = 6;
```

```
global $db;
```

```
$aWhere = array();
```

```
$aPath = '';
```

```
if(isset($_REQUEST['man'])&&is_array($_REQUEST['man'])) {
```

```
foreach($_REQUEST['man'] as $sKey=>$sVal){
```

```
if((int)$sVal!=0){
```

```
$aWhere[] = 'manufacturer_id='.(int)$sVal;
```

```
$aPath .= 'man[]='.(int)$sVal.'&';
```

```
}
```

```
}
```

```
}
```

```
if(isset($_REQUEST['p_cat'])&&is_array($_REQUEST['p_cat'])) {

foreach($_REQUEST['p_cat'] as $sKey=>$sVal) {

if((int)$sVal!=0) {

$aWhere[] = 'p_cat_id='.(int)$sVal;

$aPath .= 'p_cat[]='.(int)$sVal.'&';

}

}

}

if(isset($_REQUEST['cat'])&&is_array($_REQUEST['cat'])) {

foreach($_REQUEST['cat'] as $sKey=>$sVal) {

if((int)$sVal!=0) {

$aWhere[] = 'cat_id='.(int)$sVal;

$aPath .= 'cat[]='.(int)$sVal.'&';

}

}

}

}
```

```
$sWhere = (count($aWhere)>0?' WHERE '.implode(' or
', $aWhere): '');

$query = "select * from products ".$sWhere;

$result = mysqli_query($db, $query);

$total_records = mysqli_num_rows($result);

$total_pages = ceil($total_records / $per_page);

echo "<li><a href='shop.php?page=1'";

if(!empty($aPath)){ echo "&".$aPath; }

echo "' >". 'First Page'."</a></li>";

for ($i=1; $i<=$total_pages; $i++){

echo "<li><a
href='shop.php?page=".$i.(!empty($aPath)?'&'. $aPath: '')."
>".$i."</a></li>";

};

echo "<li><a href='shop.php?page=$total_pages'";

if(!empty($aPath)){ echo "&".$aPath; }

echo "' >". 'Last Page'."</a></li>";

}
```



```
?>
```

Лістинг коду shop.php

```
<?php
```

```
session_start();
```

```
include("includes/db.php");
```

```
include("includes/header.php");
```

```
include("functions/functions.php");
```

```
include("includes/main.php");
```

```
?>
```

```
<!-- MAIN -->
```

```
<main>
```

```
<!-- HERO -->
```

```
<div class="nero">
```

```
<div class="nero__heading">
```

```
<span class="nero__bold">Wear</span>Shop
```

```
</div>
```

```
<p class="nero__text">
```

```
</p>
```

```
</div>
```

```
</main>
```

```
<div id="content" >
```

```
<div class="container" >
```

```
<div class="col-md-12" >
```

</div>

<div class="col-md-3">

<?php include("includes/sidebar.php"); ?>

</div>

<div class="col-md-9" >

<?php getProducts(); ?>

</div>

<center>

<ul class="pagination" >

<?php getPaginator(); ?>

</center>

</div>

</div>

</div>

</div>

```
<?php

include("includes/footer.php");

?>

<script src="js/jquery.min.js"> </script>

<script src="js/bootstrap.min.js"></script>

<script>

$(document).ready(function() {

$('.nav-toggle').click(function() {

$(".panel-collapse, .collapse-data").slideToggle(700, function() {

if ($(this).css('display') == 'none') {

$(".hide-show").html('Show');

}

else {

$(".hide-show").html('Hide');

}

});

});

});
```

```
$(function(){

$.fn.extend({

filterTable: function(){

return this.each(function(){

$(this).on('keyup', function(){

var $this = $(this),

search = $this.val().toLowerCase(),

target = $this.attr('data-filters'),

handle = $(target),

rows = handle.find('li a');

if(search == '') {

rows.show();

} else {

rows.each(function(){

var $this = $(this);

$this.text().toLowerCase().indexOf(search) === -1 ? $this.hide()

: $this.show();

});

});

});
```

```
}
```

```
});
```

```
});
```

```
}
```

```
});
```

```
$('#[data-action="filter"][id="dev-table-  
filter"]').filterTable();
```

```
});
```

```
});
```

```
</script>
```

```
<script>
```

```
$(document).ready(function() {
```

```
function getProducts() {
```

```
var sPath = '';

var aInputs = $('li').find('.get_manufacturer');

var aKeys = Array();

var aValues = Array();

iKey = 0;

$.each(aInputs, function(key, oInput) {

    if(oInput.checked) {

        aKeys[iKey] = oInput.value

    };

    iKey++;

});

if(aKeys.length>0) {

    var sPath = '';

    for(var i = 0; i < aKeys.length; i++){

        sPath = sPath + 'man[]=' + aKeys[i]+'&';

    }

}
```

```
var aInputs = Array();

var aInputs = $('li').find('.get_p_cat');

var aKeys = Array();

var aValues = Array();

iKey = 0;

$.each(aInputs, function(key, oInput) {

    if(oInput.checked) {

        aKeys[iKey] = oInput.value

    };

    iKey++;

});

if(aKeys.length>0) {

    for(var i = 0; i < aKeys.length; i++){

        sPath = sPath + 'p_cat[]=' + aKeys[i]+'&';

    }

}

var aInputs = Array();

var aInputs = $('li').find('.get_cat');
```

```
var aKeys = Array();

var aValues = Array();

iKey = 0;

$.each(aInputs, function(key, oInput) {

    if(oInput.checked) {

        aKeys[iKey] = oInput.value

    };

    iKey++;

});

if(aKeys.length>0) {

    for(var i = 0; i < aKeys.length; i++){

        sPath = sPath + 'cat[]=' + aKeys[i]+'&';

    }

}

$('#wait').html('');

$.ajax({
```



```
url:"load.php",

method:"POST",

data: sPath+'sAction=getProducts',

success:function(data) {

    $('#Products').html('');

    $('#Products').html(data);

    $("#wait").empty();

}

});

$.ajax({

url:"load.php",

method:"POST",

data: sPath+'sAction=getPaginator',

success:function(data) {

$('#.pagination').html('');

$('#.pagination').html(data);

}

});

}

$('#.get_manufacturer').click(function() {
```

```
getProducts();

});

$('.get_p_cat').click(function(){

getProducts();

});

$('.get_cat').click(function(){

getProducts();

});

});

</script>

</body>

</html>
```

Лістинг коду style.css

```
body {
font-family: "Roboto", Helvetica, Arial, sans-serif;
font-size: 14px;
line-height: 1.42857143;
color: #333333;
background-color: #f0f0f0;
```

```
overflow-x: hidden;
```

```
}
```

```
#top {
```

```
background: #555555;
```

```
padding: 10px 0;
```

```
}
```

```
#top .offer {
```

```
color: #fff;
```

```
}
```

```
#top .offer .btn {
```

```
text-transform: uppercase;
```

```
}
```

```
@media (max-width: 991px) {
```

```
#top .offer {
```

```
margin-bottom: 10px;
```

```
}
```

```
}
```

```
@media (max-width: 991px) {
```

```
#top {
```

```
font-size: 12px;
```

```
text-align: center;
```

```
}
```

```
}
```

```
#top a {
```

```
color: #fff;
```

```
}
```

```
#top ul.menu {
```

```
padding-top: 5px;
```

```
margin: 0;
```

```
text-align: right;
```

```
font-size: 12px;
```

```
list-style: none;
```

```
}
```

```
@media (max-width: 991px) {
```

```
#top ul.menu {
```

```
text-align: center;
```

```
}
```

```
}
```

```
#top ul.menu > li {
```

```
display: inline-block;
```

```
}
```

```
#top ul.menu > li a {
```

```
color: #eeeeee;
```

```
}
```

```
#top ul.menu > li + li:before {
```

```
content: "\00a0";
```

```
padding: 0 5px;
```

```
color: #f7f7f7;
```

```
}
```

```
.navbar {
```

```
background:white;
```

```
}
```

```
.navbar-collapse .right {
```

```
float:right;
```

```
}
```

```
.navbar-brand {
```

```
float:left;
```

```
padding: 10px 15px;
```

```
font-size:18px;
```

```
line-height:20px;
```

```
height:70px;
```

```
}
```

```
.navbar-brand:hover,
```

```
.navbar-brand:focus{
```

```
text-decoration:none;
```

```
}
```

```
.navbar ul.nav > li > a {  
text-transform:uppercase;  
font-weight:bold;  
font-size:14px;  
  
}
```

```
.padding-nav{  
padding-top:10px;  
}
```

```
.navbar ul.nav > li > a:hover{  
background:#e7e7e7;  
}
```

```
.btn-primary{  
color:rgb(255, 255, 255);  
background-color:rgb(79, 191, 168);  
border-color:rgb(65, 179, 156);  
}
```

```
#search .navbar-form{  
float:right;  
}
```

```
#search {  
  
clear: both;  
border-top: solid 1px #9adacd;  
text-align:right;  
  
}
```

```
#search .navbar-form .input-group {  
display:table;
```

```
}

#search .navbar-form .input-group .form-control {
width:100%;

}

#slider {

margin-bottom:40px;

}

.box {
background: #fff;
margin: 0 0 30px;
border: solid 1px #e6e6e6;
box-sizing: border-box;
padding:20px;
box-shadow: 0 1px 5px rgba(0, 0, 0, 0.1);

}

#advantages {
text-align:center;
}

#advantages .box .icon {
position: absolute;
font-size: 120px;
width:100%;
```

```
text-align:center;
top: -20px;
left:0;
height:100%;
float:left;
color:#eeeeee;
transition: all 0.2s ease-out;
z-index:1;
box-sizing: border-box;

}
```

```
#advantages .box h3 {
position:relative;
margin: 0 0 20px;
font-weight: 300;
text-transform: uppercase;
z-index:2;

}
```

```
#advantages .box h3 a:hover {
text-decoration:none;
}
```

```
#advantages .box p{
position:relative;
color:#555555;
z-index:2;

}
```

```
#hot h2 {
```



```
text-transform: uppercase;
font-size: 36px;
color: #4993e4;
font-weight: 100;
text-align: center;

}

#content {
padding-left:25px;

}

.single {
width:290px;
}

@media (max-width: 768px) {

.single {
width:60%;
margin:0 auto;
}

}

#content .product {
background:#fff;
border: solid 1px #e6e6e6;
box-sizing: border-box;
margin-bottom: 30px;
}

#content .product .text p.price {
font-size:18px;
text-align:center;
```

```
font-weight:300;
```

```
}
```

```
#content .product .text .buttons {
```

```
clear:both;
```

```
text-align:center;
```

```
}
```

```
#content .product .text h3 {
```

```
text-align:center;
```

```
font-size:20px;
```

```
}
```

```
#content .product .text h3 a {
```

```
color:rgb(85, 85, 85);
```

```
}
```

```
#content .product .text {
```

```
padding:10px 10px 0px;
```

```
}
```

```
#content .product .text .buttons .btn {
```

```
margin-bottom:10px;
```

```
}
```

```
.breadcrumb {
```

```
padding: 8px 15px;
```

```
margin-bottom:20px;
```

```
background-color: #ffffff;
```

```
border-radius: 0;
```

```
box-shadow: 0 1px 5px rgba(0, 0, 0, 0.5);
```

```
}
```

```
.breadcrumb > li + li:before {  
content: ">\00a0";  
color:#cccccc;
```

```
}
```

```
@media (max-width: 991px) {
```

```
.breadcrumb {  
padding: 8px 0;  
text-align:center;  
}
```

```
}
```

```
#content .panel.sidebar-menu {
```

```
box-sizing: border-box;  
box-shadow: 0 1px 5px rgba(0, 0, 0, 0.1);
```

```
}
```

```
#content .panel.sidebar-menu ul.nav.category-menu {  
margin-bottom:20px;
```

```
}
```

```
#content .panel.sidebar-menu ul.nav.category-menu li a {  
text-transform: uppercase;  
font-weight: bold;
```

```
}
```

```
#content .panel.sidebar-menu h3 {  
padding: 5px 0;  
margin: 0;  
}
```

```
@media (max-width: 768px) {  
  
.center-responsive {  
width:70%;  
margin:0 auto;  
}  
  
}
```

```
@media (max-width: 550px) {  
  
.center-responsive {  
width:95%;  
margin:0 auto;  
  
}  
  
}
```

```
#content #productMain {  
  
margin-bottom: 30px;
```

```
}
```

```
#content #productMain .price {  
font-size: 30px;  
font-weight: 300;  
text-align: center;  
margin-top: 40px;
```

```
}
```

```
#content #mainImage {  
box-shadow: 0 1px 5px rgba(0, 0, 0, 0.7);  
}
```

```
#content #thumbs a {  
display: block;  
box-shadow: 0 1px 5px rgba(0, 0, 0, 0.2);  
border : solid 2px transparent;  
}
```

```
#content .headline {  
height: 350px;  
}
```

```
#content #cart .table tbody tr td img {  
width: 50px;  
}
```

```
#content #cart .table tbody tr td input {  
width: 40px;  
text-align: right;  
  
}
```

```
#content #cart .table tbody tr td{
vertical-align: middle;
```

```
}
```

```
#content #cart .table tfoot {
font-size:18px;
```

```
}
```

```
.box .box-footer {
background: #f7f7f7;
margin: 30px -20px -20px;
padding:20px;
border-top: solid 1px #eeeeee;
```

```
}
```

```
.box .box-footer:before,
.box .box-footer:after {
content:" ";
display: table;
}
```

```
.box .box-footer:after {
clear:both;
}
```

```
.box .box-header {
background:#f7f7f7;
margin:-20px -20px 20px;
padding:20px;
border-bottom: solid 1px #eeeeee;
```

```
}
```

```
#content #order-summary table {
```

```
margin-top: 20px;
```

```
}
```

```
#content #order-summary table td {
```

```
color: #999999;
```

```
}
```

```
#content #order-summary table tr.total td,
```

```
#content #order-summary table tr.total th{
```

```
font-size: 18px;
```

```
color:#555555;
```

```
font-weight: 700;
```

```
}
```

```
#footer {
```

```
background: #e0e0e0;
```

```
padding: 20px 0;
```

```
}
```

```
#footer ul {
```

```
padding-left: 0;
```

```
list-style: none;
```

```
}
```

```
#footer ul a {
```

```
color:#999999;
```

```
padding: 20px 0;
```

```
}
```

```
#footer .social {
```

```
text-align: left;
}
```

```
#footer .social a {
margin: 0 10px 0 0;
color: #fff;
display: inline-block;
width: 30px;
height: 30px;
border-radius: 15px;
line-height: 30px;
font-size: 15px;
text-align: center;
transition: all 0.2s ease-out;
vertical-align: bottom;
background-color:#555555;
}
```

```
#copyright {

background:#333;
color:#ccc;
padding:20px 0;
font-size:12px;

}
```

```
#copyright p {
margin: 0;
}
```

```
.nav.category-menu{
margin-top:-21px;
}
```



```
.scroll-menu{
height:260px;
overflow-y:scroll;
}
```

```
.checkbox span{
display:inline-block;
vertical-align:middle;
position:relative;
padding-left:5px;
}
```

```
.checkbox span::before{
content: "";
display: inline-block;
position:absolute;
width:17px;
height:17px;
left: 0;
margin-left: -20px;
border: 2px solid #cccccc;
border-radius:3px;
background-color:#fff;
transition:border 0.15s ease-in-out, color 0.15s ease-in-out;
}
```

```
.checkbox span::after{
display:inline-block;
position:absolute;
width:16px;
height:16px;
left:0;
top:0;
margin-left: -20px;
```

```
padding-left:3px;
padding-top:1px;
font-size:11px;
color:#555555;
```

```
}
```

```
.checkbox input[type="checkbox"]{
opacity:0;
z-index:1;
cursor:pointer;
```

```
}
```

```
.checkbox input[type="checkbox"]:focus + span::before{
```

```
outline: thin dotted;
outline:5px auto;
outline-offset: -2px;
```

```
}
```

```
.checkbox input[type="checkbox"]:checked + span::after{
```

```
font-family: "FontAwesome";
```

```
content: "\f00c";
```

```
}
```

```
.checkbox-primary input[type="checkbox"]:checked + span::before{
```

```
background-color: #337ab7;
```

```
border-color: #337ab7;
```

```
}
```

```
.checkbox-primary input[type="checkbox"]:checked + span::after{
```

```
color: #fff;
```

```
}
```

```
/* Label Styles */
```

```
.label {
```

```
text-decoration : none;
```

```
position: absolute;
```

```
top: 50px;
```

```
padding-left:51px;
```

```
z-index: 20;
```

```
}
```

```
.label .label-background {
```

```
position: absolute;
```

```
top:0;
```

```
right:0;
```

```
}
```

```
.label .thelabel {  
  
position: relative;  
width: 100px;  
padding: 6px 50px 6px 15px;  
margin: 40px 50px 10px -71px;  
color: #fff;  
font-size:18px;  
font-weight:bold;  
background-color: #337ab7;  
text-shadow: 0px 1px 2px #bbb ;  
  
}
```

```
.label .thelabel:before,  
.label .thelabel:after {  
content: '';  
position: absolute;  
width:0;  
height:0;  
}
```

```
.label .thelabel:after {  
left: 0px;  
top: 100%;  
border-width : 5px 10px;  
border-style: solid;  
border-color: #2d7b6b #2d7b6b transparent transparent;  
}
```

```
.label.sale {  
top: 0;  
  
}
```

```
.tick1{
font-size:18px !important;
color:red;
}
```

```
.cross1{
font-size:18px !important;
color:red;
}
```

```
.tick2{
font-size:18px !important;
color:red;
}
```

```
.cross2{
font-size:18px !important;
color:red;
}
```

```
/* Password Strength Checker Styles */
```

```
#meter_wrapper{
border:1px solid grey;
width:202px;
height:20px;
margin:0;
border-radius:3px;
}
```

```
#meter{
width:0px;
```

```
height:18px;
border-radius:2px;

}
```

```
#pass_type{
font-size:15px;
margin-top:10px;
position:absolute;
top:0;
right:90px;
margin-bottom:10%;
color:grey;

}
```

```
.services .col-md-4 {

margin-left:3%;

width:29%;

}
```

```
@media(max-width: 1000px){

.services .col-md-4 {

margin-left:2%;
```

```
width:46%;
```

```
}
```

```
}
```

```
@media(max-width: 800px){
```

```
.services .col-md-4 {
```

```
width:90%;
```

```
}
```

```
}
```