

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ВЕБСИСТЕМИ МАГАЗИНУ ТОВАРІВ ДЛЯ ВАГІТНИХ.
DEVELOPMENT OF A WEB SYSTEM FOR A STORE OF GOODS FOR
PREGNANT WOMEN

Виконав(ла): студент(ка) 4 курсу, групи КНТ-139

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

САЙНІДІ Д.С.

(ПРИЗВИЩЕ та ініціали)

Керівник КУЛИКОВСЬКА Н.А.

(ПРИЗВИЩЕ та ініціали)

Рецензент ТЕРЕЩЕНКО Е.В.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Факультет ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
 “ ” 2023 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

САЙНІДІ Дар’ї Сергіївни
(ПРИЗВИЩЕ, ім’я, по батькові)

1. Тема проєкту (роботи) Розробка вебсистеми магазину товарів для вагітних.
Development of a Web System for a Store of Goods for Pregnant Women

керівник проєкту (роботи) КУЛИКОВСЬКА Наталія Анатоліївна,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім’я, по батькові)

затверджені наказом закладу вищої освіти від “04” квітня 2022 року № 87

2. Строк подання студентом проєкту (роботи) 01 червня 2022 року
 3. Вихідні дані до проєкту (роботи) рекомендована література, технічне
завдання, опис товарів для вагітних, методи для проєктування та розробки
вебсайтів

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Огляд вебсистем і аналіз технологій.
2. Проєктування вебсистеми магазину. 3. Розробка основних програмних частин
вебсистеми магазину товарів для вагітних. 4. SEO-оптимізація та тестування
вебсистеми.

5. Перелік графічного матеріалу (з з точним зазначенням обов’язкових креслень, кількість слайдів, плакатів)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	КУЛИКОВСЬКА Н.А., ст.викл.		
Нормоконтроль	ЛИПОВЕЦЬ М.В., асистент		

7. Дата видачі завдання « 04 » квітня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз сечасних вебсистем та методів розробки	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури бази даних	2 тиждень	Розділ 3
5	Розробка дизайну вебсайту. Розробка серверної та клієнтської частини.	3-4 тижні	Розділ 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Дар'я САЙНІДІ
(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Наталія КУЛИКОВСЬКА
(Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 75 с., 40 рис., 2 дод., 18 джерел.

БАЗА ДАНИХ, ВЕБСАЙТ, ВАГІТНІСТЬ, КОРИСТУВАЧ, ІНТЕРНЕТ-МАГАЗИН, ІНТЕРФЕЙС, ПРОЄКТУВАННЯ, СЕРВЕР, BOOTSTRAP, JAVASCRIPT, JQUERY, HTML, MYSQL, PHP, UML-ДІАГРАМА, SEO.

Об'єкт розробки – програмні засоби проєктування та розробки вебсистеми.

Предмет розробки – вебсайт магазину товарів для вагітних.

Мета роботи – розробити інформаційну вебсистему для продажу товарів для вагітних через Інтернет мережу.

Матеріали, методи та технічні засоби: структурне та об'єктно-орієнтоване програмування, технології створення вебсайтів HTML, CSS, JavaScript, AJAX, PHP, персональний комп'ютер з процесором Intel Core 4 Duo під управлінням операційної системи Microsoft Windows 10.

Результати. Створено вебсистеми магазину товарів для вагітних, де кожен користувач може здійснити замовлення товарів через Інтернет.

Висновки. Розроблена система автоматизує роботу магазину та забезпечить зручність та ефективності покупок для вагітних жінок.

Галузь використання – інтернет магазини.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 75 pages, 40 figures, 2 appendixes, 18 sources.

DATABASE, WEBSITE, PREGNANCY, USER, ONLINE STORE, INTERFACE, DESIGN, SERVER, BOOTSTRAP, JAVASCRIPT, JQUERY, HTML, MYSQL, PHP, UML DIAGRAM, SEO.

Object of study is a software for designing and developing a web system.

Subject of study are the website of a store of goods for pregnant women.

The purpose of the work is to develop an informational web system for the sale of goods for pregnant women via the Internet.

Materials, methods, and tools: structural and object-oriented programming, technologies for creating HTML, CSS, JavaScript, AJAX, PHP websites, a personal computer with an Intel Core 4 Duo processor running the Microsoft Windows 10 operating system.

Results. The web system of the store of goods for pregnant women has been created, where every user can order goods via the Internet.

Conclusions The developed system will automate the operation of the store and ensure the convenience and efficiency of shopping for pregnant women.

The field of use is a online stores.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	7
Вступ.....	8
1 Огляд вебсистем і аналіз технологій.....	10
1.1 Аналіз сучасних вебсистем інтернет-магазинів.....	10
1.2 Аналіз технологій розробки вебсистем	13
1.3 Постановка завдання розробки вебсистеми магазину товарів для вагітних	16
1.4 Висновки за розділом 1	19
2 Проєктування вебсистеми магазину	20
2.1 Загальна структура вебсистеми	20
2.2 Проєктування бази даних вебсистеми	21
2.3 Дизайн інтерфейсу вебсистеми	23
2.4 Висновки за розділом 2	31
3 Розробка основних програмних частин вебсистеми магазину товарів для вагітних	33
3.1 Розробка серверної частини вебсистеми	33
3.2 Розробка клієнтської частини вебсистеми	35
3.3 Розробка панелі адміністратора вебсистеми	46
3.4 Висновки за розділом 3	48
4 SEO-оптимізація та тестування вебсистеми.....	50
4.1 SEO-просування сайту.....	50
4.2 Тестування кросбраузерності сайту	55
4.3 Висновки за розділом 4	58
Висновки	59
Перелік джерел посилань	61
Додаток А Програмний код форм	63
Додаток Б Слайди презентації	69

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

- API – Application Programming Interface;
- CRM – Customer Relationship Mmanagement;
- HTML – HyperText Markup Language;
- JS – JavaScript;
- MVC – Model-View-Controller;
- SEO – Search Engine Optimization;
- IC – інформаційна система.

ВСТУП

У сучасному світі, де інформаційні технології займають центральне місце, Інтернет став необхідною складовою частиною розвитку бізнесу. Для магазинів це не просто нова платформа для просування своїх товарів і послуг, але й потужний інструмент для досягнення успіху та привернення уваги клієнтів. Завдяки Інтернету, магазини отримують можливість оперативно представляти свою продукцію широкій аудиторії з різних куточків світу [1].

Сьогодні існують незліченні сайти, які надають платформу для продажу товарів та послуг. Найпопулярнішими серед них є електронні майданчики, аукціонні сайти, торгові платформи та онлайн-магазини. Ці сайти пропонують споживачам широкий спектр товарів та послуг, від одягу та електроніки до подорожей та послуг краси. Комфорт та зручність онлайн-шопінгу залучають все більше клієнтів, і це робить Інтернет одним з найбільш важливих каналів продажу [2].

Розробка вебсистеми для магазину товарів для вагітних має свою власну актуальність. За останні роки споживачі все більше орієнтуються на онлайн-покупки, зокрема, коли йдеться про спеціалізовані магазини для вагітних жінок. Це пов'язано зі зростанням свідомості щодо здоров'я та благополуччя під час вагітності, а також зі збільшенням кількості зручних та високоякісних товарів, які пропонуються спеціально для цієї категорії клієнтів [3].

Розробка вебсистеми для магазину товарів для вагітних є нагальною задачею, оскільки надає можливість забезпечити спеціалізований асортимент товарів та послуг, які відповідають унікальним потребам та вимогам вагітних жінок. Вона сприяє зручному та доступному шопінгу для майбутніх матерів, дозволяючи їм знайти необхідні товари без виходу з дому. Крім того, вебсистема забезпечує можливість швидко оновлювати асортимент, надавати детальну інформацію про товари, а також забезпечує зручність оплати та доставки [4].

Отже, розробка вебсистеми для магазину товарів для вагітних є важливим та актуальним завданням, спрямованим на задоволення потреб спеціальної аудиторії та забезпечення зручності та ефективності покупок для вагітних жінок.

Метою дипломної роботи є розробка вебсистеми магазину товарів для вагітних, який надає услуги опису товарів, продажу, оформлення замовлення та доставки товарів.

Задачі, які треба виконати в ході роботи:

- аналіз сучасних вебсистем;
- розробка бази даних;
- розробка дизайну сторінок;
- розробка вебсайту;
- SEO- оптимізація.

1 ОГЛЯД ВЕБСИСТЕМ І АНАЛІЗ ТЕХНОЛОГІЙ

1.1 Аналіз сучасних вебсистем інтернет-магазинів

Сучасні технології розробки вебсайтів відкривають широкі можливості для створення різних типів і складності вебресурсів зі спеціальною спрямованістю. З кожним роком вебтехнології продовжують удосконалюватися, піднімаючись на новий рівень. На сьогоднішній день існує безліч різноманітних рішень, які дозволяють розробити вебсайт будь-якої складності [1].

Технології розробки вебсайтів можна розділити на дві категорії: серверні та клієнтські. До серверних технологій відносяться PHP, Perl, ASP. Вони виконуються на сервері і після обробки надсилають клієнту HTML-код за допомогою клієнтських технологій. Клієнтські технології включають HTML/XML, CSS, JavaScript/AJAX, Java/JavaFX, Flex/Flash, Silverlight, ActiveX, VBScript, DHTML та інші [2].

Програми, що розробляються з використанням клієнтських мов, виконуються на боку користувача за допомогою браузера. Оскільки існує багато різних браузерів на різних платформах, це може призводити до неоднозначного відображення програми. Проблема полягає в тому, що один і той же скрипт може відображатися по-різному в різних браузерах та їх версіях. Завдяки виконанню клієнтських програм на боці користувача, а не на сервері, зменшується навантаження на сервер, оскільки програма не завжди звертається до нього [2]-[3].

При розробці вебсайту необхідно використовувати ліцензійне програмне забезпечення, яке має вбудовані можливості забезпечення безпеки та надійного зберігання даних. Вхід в систему користувача повинен пройти через реєстрацію та введення пароля.

При розробці вебсайту потрібно враховувати такі вимоги. Можливість адаптації, здатність сайту гнучко налаштовуватися і адаптуватися до різних пристроїв. Відкритість взаємодії вебсистеми з іншими додатками шляхом

обміну даними і керуючих впливів. Аналітичність надання якісної аналітичної інформації на різних рівнях управління [4].

Ергономічність інтерфейсу стандартизованих засобів інтерфейсу користувача, що відповідають операціям, виконуваним користувачами. Розв'язання проблеми розподіленого бізнесу роботи з розподіленими базами даних, доступом через Інтернет, консолідацією звітності для віддалених філій тощо.

Сучасні вебсистеми для інтернет-магазинів зробили революцію в тому, як люди здійснюють покупки, запропонувавши зручність, доступність і широкий асортимент товарів одним натисканням кнопки. Ці системи мають ряд переваг і недоліків, які важливо враховувати [5]:

Переваги:

- вебсистеми дозволяють онлайн-магазинам охоплювати глобальну аудиторію, долаючи географічні бар'єри та розширюючи свою клієнтську базу;

- на відміну від фізичних магазинів з обмеженим графіком роботи, вебсистеми дозволяють онлайн-магазинам бути доступними цілодобово, пропонуючи клієнтам гнучкість робити покупки, коли їм зручно;

- вебсистеми надають онлайн-магазинам можливість демонструвати та продавати широкий асортимент продуктів без обмежень фізичного простору на полицях. Це дозволяє клієнтам отримати доступ до більшої кількості товарів;

- вебсистеми можуть використовувати дані та вподобання клієнтів, щоб пропонувати персоналізовані рекомендації та індивідуальний досвід покупок, підвищуючи задоволеність клієнтів і залученість;

- керування онлайн-магазином через вебсистему може бути економічно ефективнішим порівняно зі звичайним магазином. Це усуває витрати, пов'язані з фізичною інфраструктурою, персоналом та обслуговуванням.

Недоліки сучасних вебсистем:

- онлайн-покупки через вебсистеми не мають тактильного та сенсорного досвіду традиційних покупок, яким деякі клієнти можуть віддати перевагу;

- вебсистемам може бракувати персоналізованої допомоги та особистої взаємодії, які клієнти можуть відчуті у звичайних магазинах;

- вебсистеми значною мірою залежать від технологій, що робить їх уразливими до технічних проблем, таких як збої вебсайтів, проблеми з підключенням і загрози кібербезпеці;

Вебсистеми для інтернет-магазинів використовують різні ІТ-технології, щоб забезпечити безперешкодний досвід покупок. Інтернет-магазини побудовані на платформах електронної комерції, які пропонують такі функції, як керування каталогом продуктів, функціональність кошика для покупок і безпечні платіжні шлюзи. Системи керування вмістом (CMS) дозволяють легко керувати вмістом вебсайту, дозволяючи онлайн-магазинам оновлювати інформацію про продукт, зображення та рекламні акції. Системи CRM допомагають відстежувати взаємодію з клієнтами, керувати даними клієнтів і надавати персоналізований досвід. Інтернет-магазини використовують інструменти аналітики для відстеження продуктивності вебсайту, поведінки користувачів і даних про продажі. Ця інформація допомагає приймати рішення на основі даних і оптимізувати покупки [6].

Інтернет-магазини, які обслуговують саме вагітних жінок, набули популярності завдяки унікальним потребам і перевагам цієї цільової аудиторії [7]. Деякі ключові аспекти, які слід враховувати під час аналізу цих магазинів, включають:

- асортимент продукції. Інтернет-магазини для вагітних повинні пропонувати різноманітний асортимент товарів, адаптованих до потреб майбутніх матерів, включаючи одяг для вагітних, вітаміни для вагітних, дитячі товари першої необхідності та аксесуари, пов'язані з вагітністю;

- зручний інтерфейс. Вебсистема повинна мати інтуїтивно зрозумілий і зручний інтерфейс, який дозволяє вагітним жінкам легко переміщатися по сайту, знаходити товари та робити покупки;

- посібники з розмірів і підгонки. Інтернет-магазини повинні надавати точні довідники з розмірами та інформацію про підгонку, щоб допомогти вагітним жінкам у виборі одягу та аксесуарів, які відповідають їхнім змінним фігурам;

- довіра та безпека. Враховуючи делікатний характер товарів, пов'язаних із вагітністю, ці магазини повинні надавати пріоритет довірі та безпеці, впроваджуючи безпечні платіжні шлюзи, прозору політику повернення та зберігаючи конфіденційність даних клієнтів.

- поради та підтримка експертів. Інтернет-магазини можуть покращити досвід покупок, надаючи доступ до порад експертів, таких як блоги, форуми та підтримка клієнтів, щоб вирішити конкретні проблеми та запитання вагітних жінок.

Підсумовуючи, сучасні вебсистеми для інтернет-магазинів пропонують численні переваги, включаючи глобальне охоплення, доступність 24/7, розширений асортимент товарів, персоналізацію та економічну ефективність. Однак вони також мають обмеження, пов'язані з відсутністю фізичного досвіду та особистої взаємодії. Ці системи покладаються на такі ІТ-технології, як платформи електронної комерції, CMS, CRM та інструменти аналітики. Аналізуючи інтернет-магазини для вагітних жінок, слід враховувати такі фактори, як асортимент продукції, зручний інтерфейс, інструкції щодо розміру та посадки, довіра та безпека, а також поради експертів, щоб задовольнити унікальні потреби майбутніх матерів.

1.2 Аналіз технологій розробки вебсистем

Технології розробки вебсистем швидко розвивалися, пропонуючи розробникам широкий спектр можливостей для створення надійних і

ефективних вебсистем. Розглянемо сучасні популярні технології, останні розробки, їх переваги та недоліки, а також використання ІТ-технологій як на клієнтській, так і на серверній стороні вебсистем.

Сучасні популярні технології:

- інтерфейсні технології: HTML5, CSS3 і JavaScript широко використовуються для створення інтерактивних і чуйних інтерфейсів користувача. Такі фреймворки, як React, Angular і Vue.js, набули популярності завдяки своїй ефективній розробці на основі компонентів і покращенню взаємодії з користувачем [8];

- бекенд-технології: популярні мови програмування для бекенд-розробки включають Python, Ruby, Java та Node.js. Такі фреймворки, як Django, Ruby on Rails, Spring і Express.js, надають розробникам інструменти та бібліотеки для створення масштабованих і безпечних вебдодатків;

- бази даних: бази даних на основі SQL, такі як MySQL, PostgreSQL і Oracle, а також бази даних NoSQL, такі як MongoDB і Firebase, зазвичай використовуються для зберігання та пошуку даних у вебсистемах [9];

- API: вебсистеми часто використовують API (інтерфейси прикладного програмування) для інтеграції із зовнішніми службами, уможливорюючи такі функції, як обробка платежів, інтеграція соціальних медіа та служби геолокації.

Прогресивні вебпрограми (PWA) поєднують у собі найкращі функції вебпрограм і мобільних програм, пропонуючи користувачам працювати, схожу на програму, безпосередньо через веббраузер. Вони забезпечують автономну роботу, push-повідомлення та покращену продуктивність [10].

Односторінкові програми (SPA) завантажують одну сторінку HTML і динамічно оновлюють вміст під час взаємодії користувачів із програмою. Цей підхід покращує взаємодію з користувачем, мінімізуючи перезавантаження сторінок і покращуючи швидкість реагування.

Архітектура мікросервісів замість створення монолітних додатків вебсистеми все частіше застосовують архітектуру мікросервісів, де додатки

поділяються на менші, слабо пов'язані сервіси, які можна розробляти, розгортати та масштабувати незалежно [11].

До переваг сучасних технологій розробки вебсистем можна віднести:

- ефективність. Сучасні технології пропонують інструменти, фреймворки та бібліотеки, які спрощують процеси розробки, забезпечуючи швидшу та ефективнішу розробку вебсистем;
- масштабованість. Багато технологій підтримують горизонтальне масштабування, що дозволяє вебсистемам справлятися зі збільшеним трафіком і вимогами користувачів без втрати продуктивності;
- гнучкість. Розробники можуть вибирати з широкого спектру технологій на основі вимог проєкту, забезпечуючи найкращу відповідність конкретним потребам вебсистеми.

Недоліки та проблеми, пов'язані з технологіями розробки вебсистем, включають:

- крива навчання. Деякі технології потребують кривої навчання, щоб розробники стали досвідченими, особливо під час впровадження нових фреймворків або мов;
- сумісність. Забезпечення сумісності між браузерами та пристроями може бути складним завданням, оскільки різні технології можуть мати різний рівень підтримки на різних платформах;
- безпека. Вебсистеми вразливі до вразливостей безпеки, і розробники повинні впроваджувати належні заходи безпеки для захисту від таких загроз, як витік даних і атаки ін'єкцій [9].

Вебсистеми використовують різні ІТ-технології як на стороні клієнта, так і на стороні сервера. Технології на стороні клієнта [4]:

- HTML, CSS і JavaScript: ці технології забезпечують представлення, стиль та інтерактивність вебсторінок у браузері користувача;
- фронтальні фреймворки такі, як React, Angular і Vue.js, спрощують розробку динамічних і чуйних інтерфейсів користувача;

- API вебпереглядача надають доступ до таких функцій, як геолокація, локальне сховище та push-повідомлення, покращуючи взаємодію з користувачем.

Серверні технології [6]:

- мови програмування такі мови, як Python, Ruby, Java та Node.js, керують серверною логікою та обробкою даних;
- внутрішні фреймворки такі, як Django, Ruby on Rails, Spring і Express.js, надають інструменти та бібліотеки для розробки на стороні сервера, маршрутизації та інтеграції бази даних [10];
- технології баз даних. Бази даних зберігають і керують даними вебсистеми з базами даних на основі SQL, такими як MySQL, для структурованих даних і базами даних NoSQL, такими як MongoDB, для неструктурованих даних [12].

Підсумовуючи, сучасні технології розробки вебсистем пропонують такі переваги, як ефективність, масштабованість і гнучкість. Останні розробки включають PWA, SPA та архітектуру мікросервісів. Однак є труднощі, які слід враховувати, зокрема крива навчання, сумісність і безпека. ІТ-технології використовуються як на стороні клієнта, так і на стороні сервера, охоплюючи HTML, CSS, JavaScript, зовнішні фреймворки, мови програмування, бек-енд фреймворки та бази даних.

1.3 Постановка завдання розробки вебсистеми магазину товарів для вагітних

Основна мета вебсистеми – забезпечити інтуїтивно зрозумілу та зручну платформу для вагітних жінок, щоб переглядати та купувати продукти, адаптовані до їхніх потреб. Для цього вебсистема складається з кількох основних блоків, кожен з яких виконує певну функцію.

Внутрішній інтерфейс - це блок зосереджений на візуальному представленні вебсайту. Він містить елементи дизайну, макет і компоненти

інтерфейсу користувача, які забезпечують привабливість і залучення відвідувачів. Внутрішній інтерфейс має бути чутливим і оптимізованим як для настільних, так і для мобільних пристроїв [13].

Каталог товарів. Блок каталогу продуктів служить центральним сховищем для всіх продуктів, доступних у магазині. Він включає різні категорії та підкатегорії, щоб допомогти користувачам легко орієнтуватися та знаходити потрібні елементи. У кожному списку продукту мають міститися відповідні деталі, як-от зображення продукту, опис, ціни та наявність.

Кошик для покупок. Блок кошика для покупок дозволяє користувачам вибирати та збирати продукти, які вони бажають придбати. Це дозволяє користувачам переглядати вибрані товари, змінювати кількість і переходити до процесу оформлення замовлення. Кошик для покупок має забезпечувати безперебійний і безпечний досвід, гарантуючи, що користувачі можуть без зусиль додавати, видаляти або оновлювати вибрані товари [6].

Реєстрація та авторизація користувачів - це блок присвячений управлінню користувачами та автентифікації. Це дозволяє новим користувачам створювати облікові записи, надаючи необхідну інформацію, таку як ім'я, адреса електронної пошти та пароль. Потім зареєстровані користувачі можуть безпечно входити в систему, використовуючи свої облікові дані. Реєстрація та авторизація користувачів життєво важливі для персоналізації процесів покупок, керування замовленнями та використання таких функцій, як списки бажань і збережені адреси.

Оформлення замовлення обробляє весь життєвий цикл замовлення. Він включає такі функції, як розміщення замовлень, інтеграція платежів, відстеження замовлень та історія замовлень. Це забезпечує плавний та ефективний процес від моменту підтвердження користувачем своєї покупки до успішної доставки продуктів [7].

Блок керування вмістом дозволяє адміністраторам динамічно керувати та оновлювати вміст вебсайту. Він включає такі функції, як додавання нових списків продуктів, редагування наявного вмісту, керування рекламними

акціями та публікація статей у блогах або інформаційного вмісту, пов'язаного з вагітністю.

У процесі розробки вкрай важливо віддавати перевагу зручності використання, продуктивності та безпеці. Реалізація принципів адаптивного дизайну, оптимізація часу завантаження сторінок і включення безпечних платіжних шлюзів є важливими міркуваннями для успіху вебсистеми [14].

Ретельно інтегруючи ці блоки та забезпечуючи їх безперебійну функціональність, вебсистема для магазину товарів для вагітних прагне забезпечити зручні та приємні покупки, задовольняючи специфічні потреби своєї цільової аудиторії [15] – [16].

На основі перлічених блоків структури вебсистеми розроблена діаграм варіантів використання UML (рис.1.1).

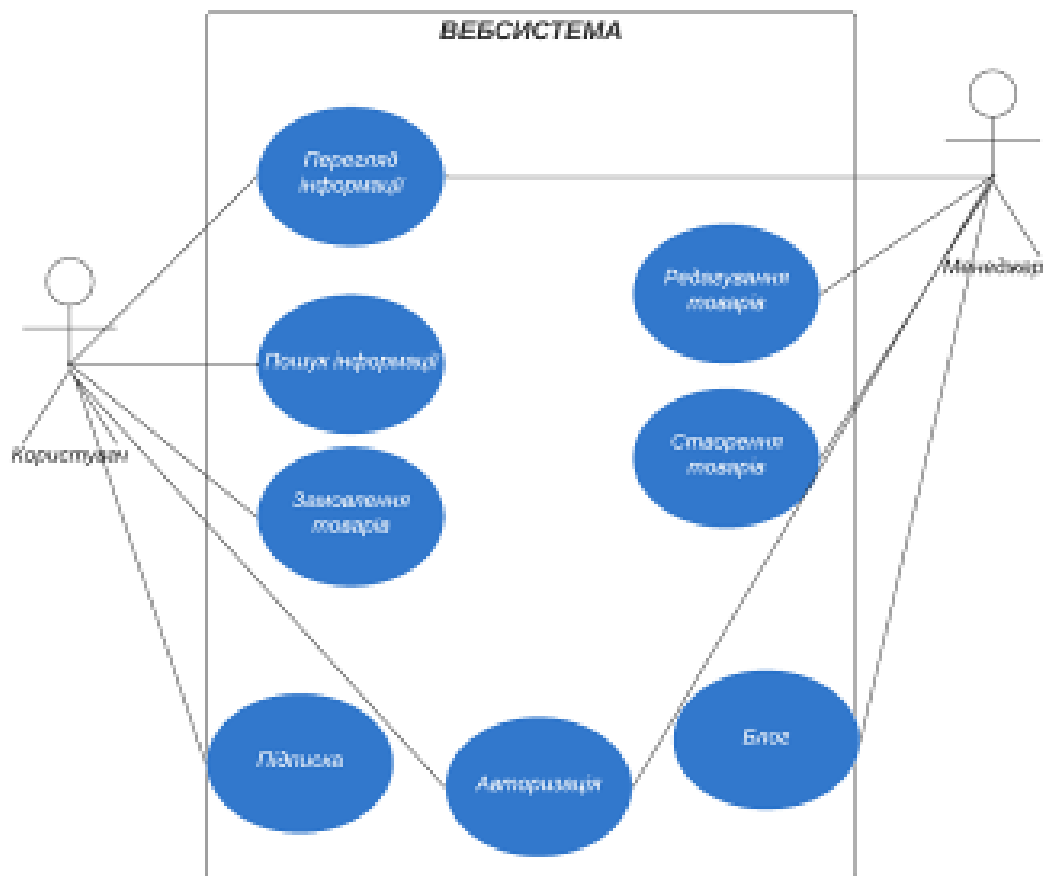


Рисунок 1.1- UML-Діаграма вебсистеми

1.4 Висновки за розділом 1

Перший розділ присвячений комплексному розгляду сучасних вебсистем. Він охоплював оцінку клієнтських фреймворків і серверних технологій, проливаючи світло на їхню складну роботу та можливості. Розділ також забезпечив теоретичну основу, окресливши фундаментальні принципи розробки вебсистем. Крім того, він встановив цілі та завдання, які спонукали до розвитку конкретної системи, що розглядається. На основі аналізу функціональних вимог було створено ретельну діаграму варіантів використання, яка візуально представляє різні способи використання вебсистеми.

2 ПРОЄКТУВАННЯ ВЕБСИСТЕМИ МАГАЗИНУ

2.1 Загальна структура вебсистеми

Розробка вебсистеми для магазину товарів для вагітних передбачає планування та продуманість структури сайту для забезпечення безперебійної взаємодії з користувачем. Структурна схема сайту не лише окреслює сторінки та їх посилання, але й дає можливість майбутнього розширення шляхом розміщення додаткових розділів і посилань.

Діаграма також визначає методи навігації, які будуть використовуватися на вебсайті. Ці методи навігації включають переміщення за посиланнями на сайті, використання форми пошуку на сайті та доступ до вмісту через портал. Хоча всі три методи необов'язкові, найчастіше використовується підхід за допомогою посилань на сайті.

На рисунку 2.1 зображено структурну схему сайту магазину одягу для вагітних. Розглянемо елементи сайту першого рівня докладніше.

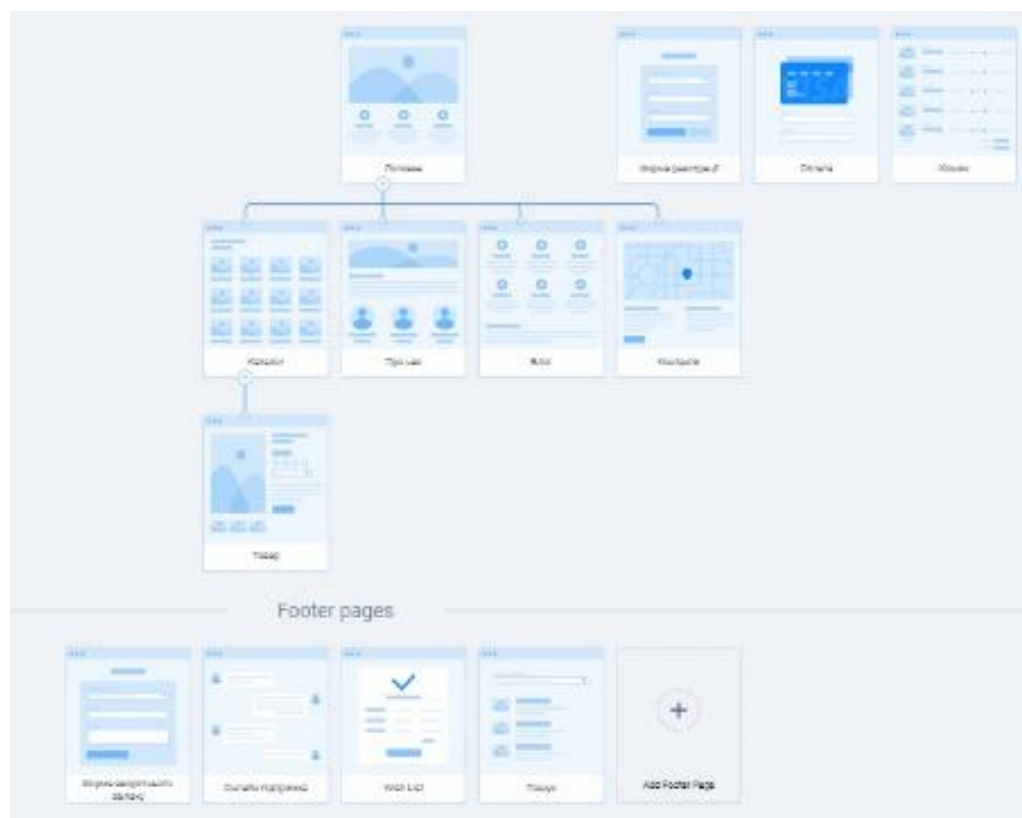


Рисунок 2.1 – Структура вебсистеми магазину товарів для вагітних

Головна сторінка служить точкою входу на вебсайт, привертаючи увагу цільової аудиторії та перетворюючи користувачів на клієнтів. На головній сторінці нашої системи буде меню, контактна інформація та блоки новин.

Меню є основною системою навігації, присутньою на більшості вебсайтів. Він складається з коротких текстових посилань, які спрямовують користувачів на певні сторінки сайту. Щоб оптимізувати взаємодію з користувачем, бажано розміщувати найважливіші посилання в меню, оскільки це найбільш активна область сайту. У нашому випадку в меню будуть такі розділи, як «Головна», «Каталог», «Блог» і «Про нас».

У багатьох випадках меню рубрик має пріоритет і стає де-факто головним меню, тоді як основне меню відіграє другорядну роль. Приклад цього можна побачити на нашому сайті, де меню заголовків у нижньому колонтитулі забезпечує основну навігацію вмістом сайту. Заголовки можуть містити такі категорії, як «Профіль», «Замовлення» та «Контакти».

2.2 Проєктування бази даних вебсистеми

Завдання розробки бази даних для магазину одягу для вагітних передбачає розробку та реалізацію системи керування базами даних за допомогою PHP і MySQL. Мета полягає в тому, щоб створити ефективну та організовану систему для зберігання та керування інформацією, пов'язаною з продуктами, користувачами, замовленнями та блогом.

Сутності в базі даних:

- «Товари»: цей об'єкт представляє одяг для вагітних, доступний у магазині. Він містить такі атрибути, як ідентифікатор товару, назва, опис, ціна, розмір, колір і кількість на складі (рис. 2.2, 2.3);

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	user_id	int(11)			No	None		AUTO_INCREMENT	
☐	2	name	varchar(255)	utf8_general_ci		No	None			
☐	3	email	varchar(255)	utf8_general_ci		No	None			
☐	4	phone	varchar(255)	utf8_general_ci		Yes	NULL			
☐	5	password	varchar(255)	utf8_general_ci		No	None			
☐	6	salt	varchar(255)	utf8_general_ci		Yes	NULL			
☐	7	role	varchar(255)	utf8_general_ci		Yes	NULL			

Рисунок 2.2 – Таблица «Товари»

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	category_id	int(11)			No	None		AUTO_INCREMENT	
☐	2	name	varchar(255)	utf8_general_ci		No	None			
☐	3	sort_order	int(11)			No	0			
☐	4	status	int(11)			No	1			

Рисунок 2.3 – Таблица «Категорії»

- «Користувачі»: Ця організація зберігає інформацію про користувачів інтернет-магазину. Він містить такі атрибути, як ідентифікатор користувача, ім'я, адреса електронної пошти, пароль, адреса доставки та контактні дані (рис. 2.4);

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	user_id	int(11)			No	None		AUTO_INCREMENT	
☐	2	name	varchar(255)	utf8_general_ci		No	None			
☐	3	email	varchar(255)	utf8_general_ci		No	None			
☐	4	phone	varchar(255)	utf8_general_ci		Yes	NULL			
☐	5	password	varchar(255)	utf8_general_ci		No	None			
☐	6	salt	varchar(255)	utf8_general_ci		Yes	NULL			
☐	7	role	varchar(255)	utf8_general_ci		Yes	NULL			

Рисунок 2.4 – Таблица «Користувачі»

- «Замовлення»: ця організація відстежує деталі замовлень клієнтів. Він містить такі атрибути, як ідентифікатор замовлення, ідентифікатор користувача, дата замовлення, загальна сума, статус платежу та статус доставки (рис. 2.5);

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	
2	user_comment	text	utf8_general_ci		No	None			
3	user_id	int(11)			Yes	NULL			
4	date	timestamp			No	CURRENT_TIMESTAMP		DEFAULT_GENERATED	
5	status	int(11)			No	1			
6	product_id	int(11)			Yes	NULL			
7	quantity	int(11)			Yes	NULL			

Рисунок 2.5 – Таблица «Замовлення»

На рис. 2.6 наведена розроблена база даних в phpMyAdmin.

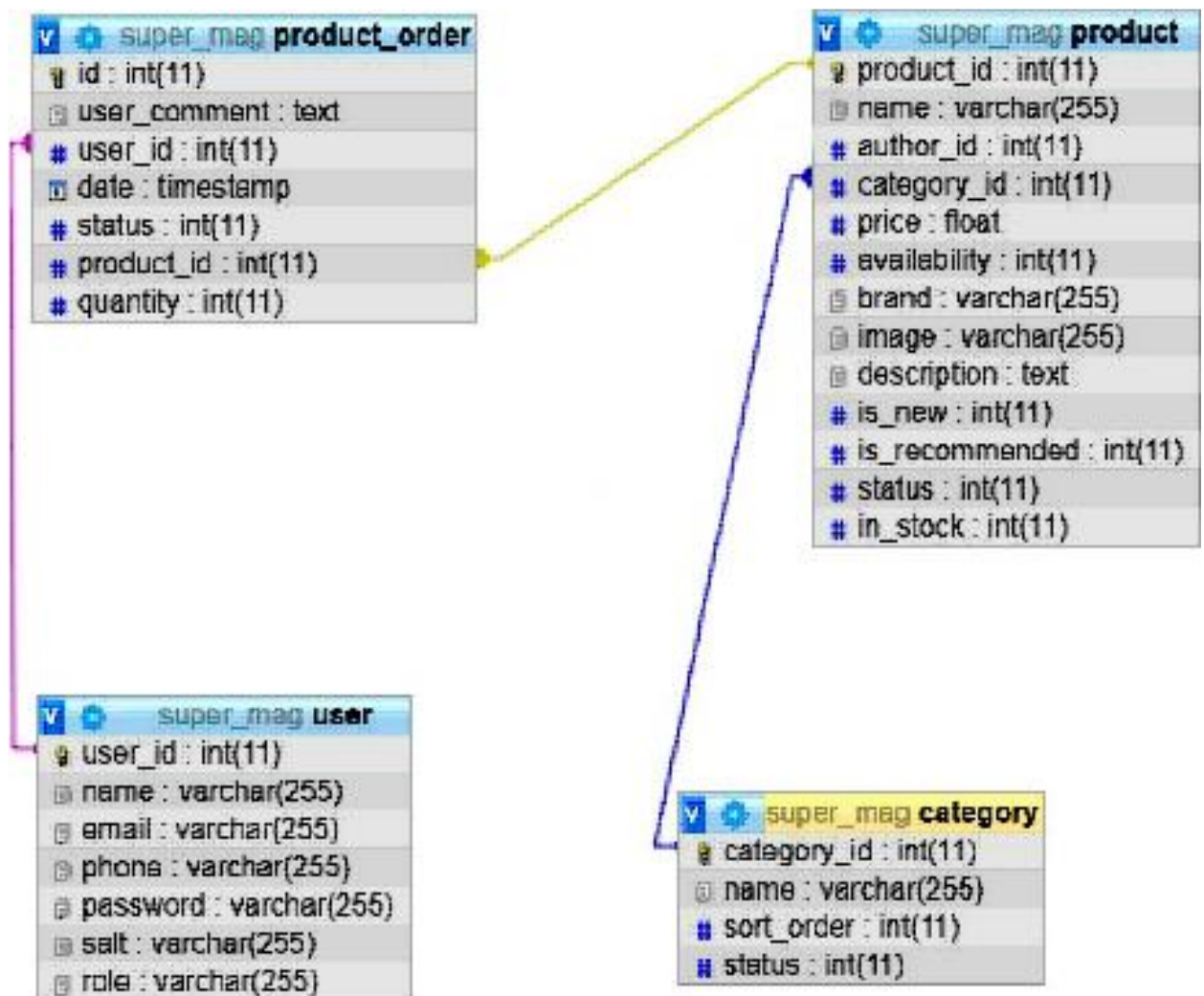


Рисунок 2.6 – Розроблена база даних

2.3 Дизайн інтерфейсу вебсистеми

Перед тим, як заглибитися у візуальний дизайн, створюється каркас, щоб окреслити основну структуру та макет вебсайту. Це включає ескіз

розміщення ключових елементів і блоків вмісту без зосередження уваги на конкретних деталях дизайну.

Після завершення створення каркасу починається етап візуального дизайну. Це включає в себе вибір колірних схем, типографіки та графічних елементів, які відповідають ідентичності бренду та бажаному досвіду користувача.

Макет сторінки: макет сторінки визначає, як різні елементи розташовані на вебсторінці. Це передбачає врахування ієрархії вмісту, визначення розміщення верхніх і нижніх колонтитулів, навігаційних меню та бічних панелей, а також забезпечення збалансованого розподілу візуальних елементів.

Чуйний дизайн: у зв'язку зі збільшенням використання мобільних пристроїв дуже важливо створювати адаптивний дизайн, який адаптується до різних розмірів і орієнтацій екрана. Чуйний дизайн забезпечує оптимальну взаємодію з користувачами на різних пристроях, підвищуючи доступність і зручність використання.

Розміщення основні елементів вебсторінки та їх переваги.

Заголовок: зазвичай містить логотип вебсайту, навігаційне меню та ключову контактну інформацію. Це допомагає користувачам легко орієнтуватися на вебсайті та забезпечує послідовність брендингу.

Блоки вмісту: Блоки вмісту використовуються для представлення різних розділів інформації, наприклад рекомендованих продуктів, послуг або публікацій у блогах. Добре організовані та візуально привабливі блоки вмісту покращують читабельність і залучають користувачів.

Кнопки із закликом до дії (СТА): ці кнопки спонукають користувачів виконувати певні дії, наприклад підписуватися на інформаційний бюлетень, робити покупки або запитувати додаткову інформацію. СТА мають бути візуально чіткими, лаконічними та стратегічно розміщеними, щоб керувати поведінкою користувачів.

Зображення та візуальні матеріали: високоякісні зображення та візуальні матеріали покращують візуальну привабливість вебсайту, привертають увагу та ефективно передають інформацію. Вони допомагають створити емоційний зв'язок із користувачами та підтримують загальне повідомлення бренду.

Меню навігації: чітке та інтуїтивно зрозуміле меню навігації дозволяє користувачам легко переміщатися вебсайтом. Він повинен бути на видному місці, мати логічну організацію та реагувати на різні пристрої.

Нижній колонтитул: нижній колонтитул розташований у нижній частині сторінки та часто містить додаткові навігаційні посилання, контактну інформацію, інформацію про авторські права та посилання на важливі сторінки. Це створює відчуття завершення та допомагає користувачам знайти додаткову інформацію.

Білий простір: порожній простір між елементами та розділами на вебсторінці. Це покращує читабельність, зменшує візуальний безлад і дозволяє виділити важливі елементи.

Типографіка: вибір шрифтів і типографіки відіграє вирішальну роль у зручності читання та візуальній привабливості вебсайту. Він має бути розбірливим, відповідати стилю бренду та реагувати на різні пристрої.

Ретельно проєктуючи та впроваджуючи ці елементи, розробники вебсайтів можуть створювати візуально привабливі, зручні та ефективні вебсторінки, які відповідають потребам як власника вебсайту, так і його відвідувачів.

Головна сторінка магазину складається з наступних елементів. Лого-банер - помітний і візуально яскравий банер у верхній частині сторінки. Містить високоякісні зображення майбутніх матерів у модному одязі для вагітних разом із захоплюючим заголовком або рекламним повідомленням.

Навігаційне меню: чітке та інтуїтивно зрозуміле навігаційне меню у верхній частині сторінки, щоб допомогти відвідувачам легко досліджувати

різні розділи вебсайту. Містить категорії, як «Головна», «Товари», «Блог» і «Про нас», щоб забезпечити швидкий доступ до відповідної інформації.

Рекомендовані товари: на першій сторінці вибір рекомендованих товарів або колекцій одягу для вагітних. Використовуйте візуально привабливі зображення, короткі описи та помітні кнопки із закликом до дії, щоб спонукати відвідувачів досліджувати та робити покупки.

Рекламні пропозиції (тренди): демонструють будь-які поточні акції, знижки чи спеціальні пропозиції на першій сторінці. Це може бути у формі банерів, слайдерів або спеціальних розділів, щоб привернути увагу та спонукати відвідувачів скористатися вигідними пропозиціями.

Відгуки. Позитивні відгуки можуть переконати потенційних покупців у якості та вартості одягу для вагітних.

Підписка на розсилку: форма для підписки або заклик до дії, що запрошує відвідувачів підписатися на магазин. Це дозволяє магазину залучати потенційних клієнтів для майбутніх маркетингових заходів і інформувати клієнтів про нові продукти, акції та оновлення.

Інтеграція соціальних медіа: піктограми або канали соціальних мереж, щоб зв'язати відвідувачів із профілями магазину в соціальних мережах. Це заохочує взаємодію, дозволяє клієнтам ділитися своїм досвідом і допомагає створити спільноту навколо бренду.

Футер: добре структурований нижній колонтитул, який містить такі важливі посилання, як контактна інформація, правила доставки та повернення, поширені запитання, посібники з розмірами та додаткові параметри навігації. Він надає вичерпне резюме та навігаційну допомогу для відвідувачів.

Макет головної сторінки показан на рисунку 2.7.

Сторінка «Каталог» складається з наступних елементів. Розташування сітки: продукти в чистому та структурованому макеті сітки. Кожен список продукту має містити чітке мініатюрне зображення, назву продукту, ціну та короткий опис або основні характеристики. Це дозволяє відвідувачам швидко сканувати та порівнювати різні варіанти.

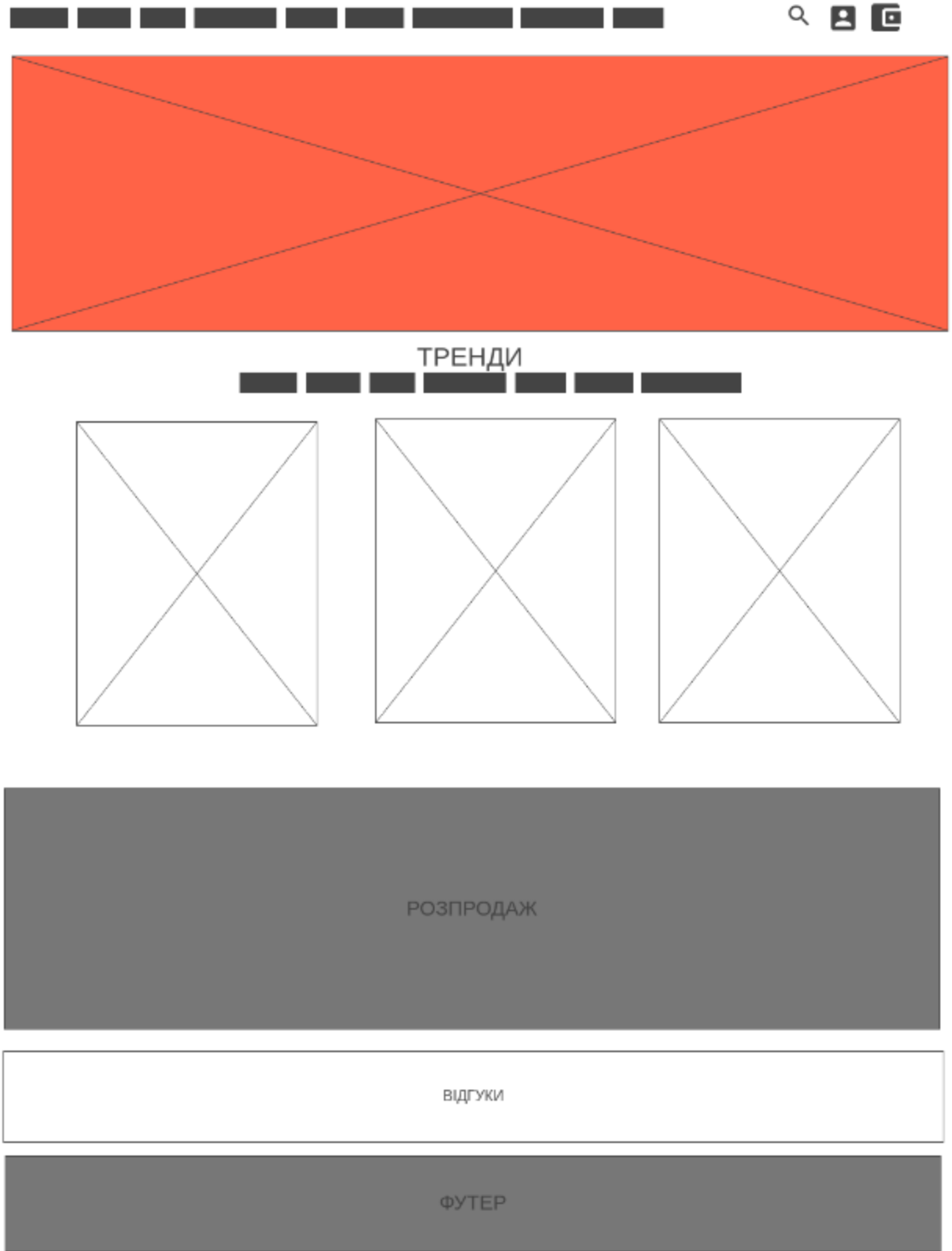


Рисунок 2.7 – Макет сторінки «Головна»

Параметри сортування та фільтрації: функції сортування та фільтрації, щоб допомогти користувачам уточнити пошук і легко знайти потрібні

продукти (сортування за ціною, розміром, кольором, стилем і наявністю). Це покращує взаємодію з користувачем і економить час для покупців.

Категорії та підкатегорії продуктів: розділіть продукти на відповідні категорії та підкатегорії, щоб полегшити навігацію. Посилання на сторінку з детальним описом товару.

Макет сторінки каталогу товарів показан на рисунку 2.8.

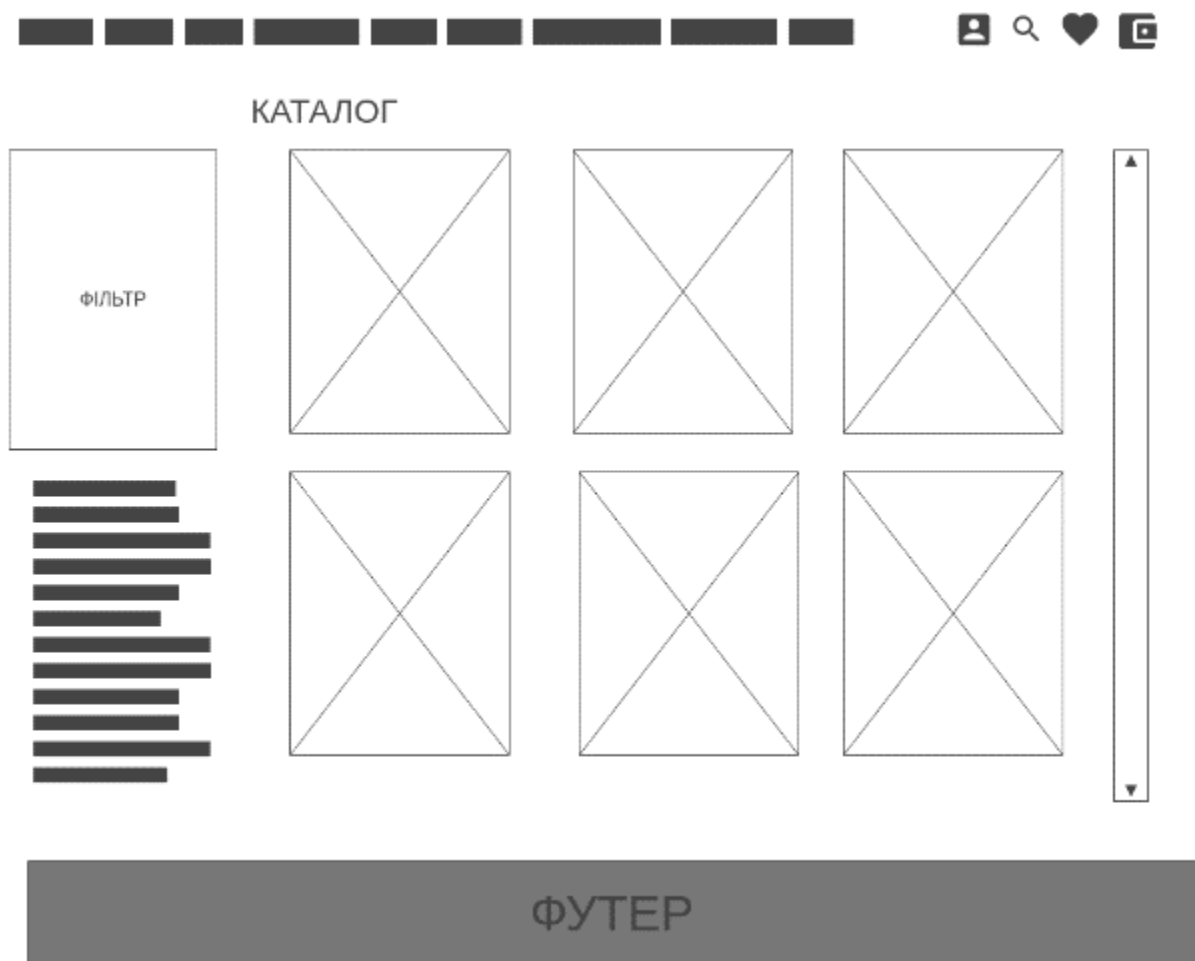


Рисунок 2.8 – Макет сторінки «Каталог»

Сторінка «Про нас» містить такі блоки (рис. 2.9). Логотип і фірмова символіка магазину. Вступ і опис магазину: короткий вступ до магазину одягу для вагітних, підкресливши його місію, цінності та унікальні особливості. Показуйте позитивні відгуки від задоволених клієнтів, щоб спонукати відвідувачів зробити покупку.

Інформація про магазин і контактні дані: фізичну адресу магазину, номер телефону, адресу електронної пошти та дані соціальних мереж. Додайте контактну форму або пряме посилання на спеціальну сторінку контактів, щоб полегшити спілкування з клієнтами.

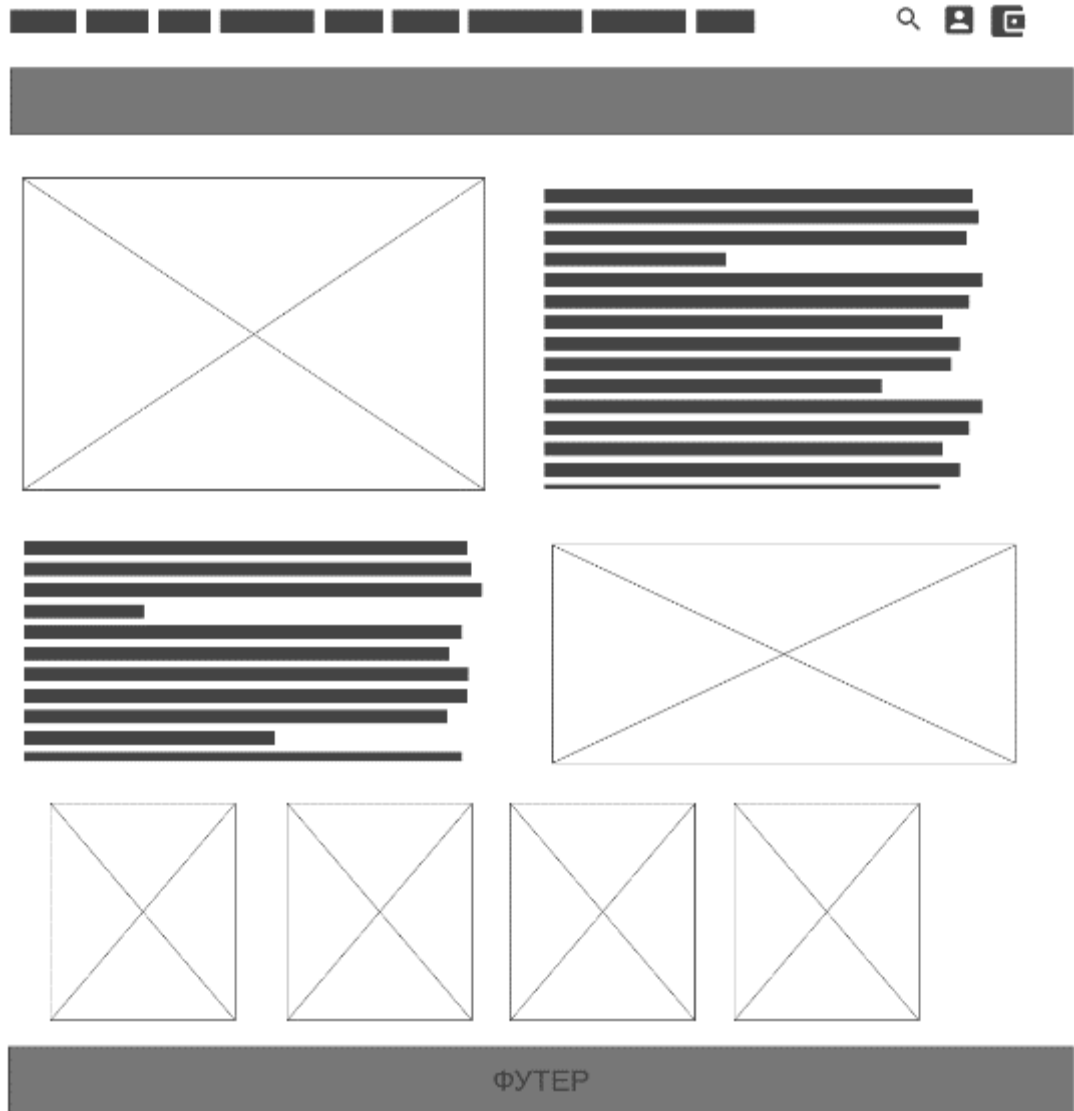


Рисунок 2.9 – Макет сторінки «Про нас»

Розроблені форми реєстрації, авторизації, кошика для покупок, підписки та реєстрації замовлень, щоб забезпечити бездоганну та зручну роботу для відвідувачів. Ось основні поля, включені в кожну з цих форм:

Форма реєстрації складається з:

- повне ім'я: щоб отримати повне ім'я користувача для персоналізації та спілкування;
- адреса електронної пошти: для створення унікального ідентифікатора та забезпечення зв'язку з користувачем;
- пароль: щоб забезпечити безпеку облікового запису та дозволити користувачам отримати доступ до свого облікового запису в майбутньому;
- підтвердити пароль: поле для повторного введення вибраного пароля для підтвердження;
- додаткова інформація: необов'язкові поля для збору додаткових відомостей, таких як номер телефону, адреса або демографічна інформація.

Форма авторизації/входу:

- адреса електронної пошти: зареєстрована адреса електронної пошти користувача для ідентифікації;
- пароль, пов'язаний з обліковим записом користувача;
- запам'ятати мене, опція прапорця для ввімкнення автоматичного входу для зручності під час майбутніх відвідувань.

Кошик:

- зображення продукту: візуальне представлення товару, доданого до кошика;
- назва або назва вибраного продукту;
- ціна продукту;
- кількість. Поле для вказівки бажаної кількості вибраного товару;
- проміжний підсумок: автоматично розраховане поле, що відображає загальну вартість кожного елемента;
- видалити: можливість видалити товар із кошика для покупок.

Форма підписки:

- адреса електронної пошти;
- кнопка підписки: кнопка із закликом до дії для підтвердження запиту на підписку.

Форма оформлення замовлення:

- адреса доставки: поле для введення користувачем адреси, куди має бути відправлено замовлення;
- платіжна адреса: поле для введення користувачем платіжної адреси, якщо вона відрізняється від адреси доставки;
- спосіб оплати: спадне меню або поле вибору для вибору бажаного способу оплати (кредитна картка, PayPal тощо);
- інформація про кредитну картку: поля для введення даних кредитної картки, як-от номер картки, термін дії та CVV;
- розмістити замовлення: кнопка із закликом до дії, щоб завершити замовлення та продовжити оплату.

Ці форми були розроблені з підходом, орієнтованим на користувача, зберігаючи їх простими, інтуїтивно зрозумілими та безпечними. Належну перевірку та обробку помилок реалізовано, щоб забезпечити точність даних і безперебійну роботу користувача.

2.4 Висновки за розділом 2

Підсумовуючи, розробка дизайну вебсайту передбачає повне розуміння цільової аудиторії, цілей вебсайту та ефективне використання елементів дизайну. Структура сайту, як правило, включає заголовок, область основного вмісту та нижній колонтитул, тоді як форми задовольняють потреби реєстрації користувачів, авторизації, покупки та підписки.

Розробка дизайну вебсайту передбачає створення візуально привабливих макетів, вибір відповідних кольорів і типографіки, а також забезпечення зручного інтерфейсу.

Ключові елементи дизайну вебсайту включають заголовок, який привертає увагу відвідувачів і швидко передає суть сайту, і нижній колонтитул, який надає важливу інформацію та підвищує зручність для користувачів. Крім того, сторінка каталогу відіграє вирішальну роль у

ефективній демонстрації товарів, тоді як сторінка з інформацією про магазин містить відповідну інформацію про магазин одягу для вагітних.

Форми реєстрації та авторизації фіксують інформацію користувача для створення облікових записів і доступу до них, а форми реєстрації кошика для покупок і замовлення полегшують процес покупки. Форми підписки дозволяють користувачам отримувати оновлення та інформаційні бюлетені від магазину. Ці форми містять такі поля, як ім'я, адреса електронної пошти, пароль, адреси доставки та виставлення рахунку, способи оплати тощо, залежно від їх конкретного призначення.

3 РОЗРОБКА ОСНОВНИХ ПРОГРАМНИХ ЧАСТИН ВЕБСИСТЕМИ МАГАЗИНУ ТОВАРІВ ДЛЯ ВАГІТНИХ

3.1 Розробка серверної частини вебсистеми

Щоб забезпечити чітку та організовану структуру, проєкт було розділено на окремі каталоги. Компоненти, пов'язані з сервером, перемістили до папки «node_modules», а стилі CSS знайшли своє місце у визначеній папці з відповідною назвою. Графічні елементи та зображення були консолідовані в папці «img». Папка «vendor» містить відповідні сторінки для бездоганної інтеграції. Папка «js» для сценаріїв JavaScript (рис. 3.1).

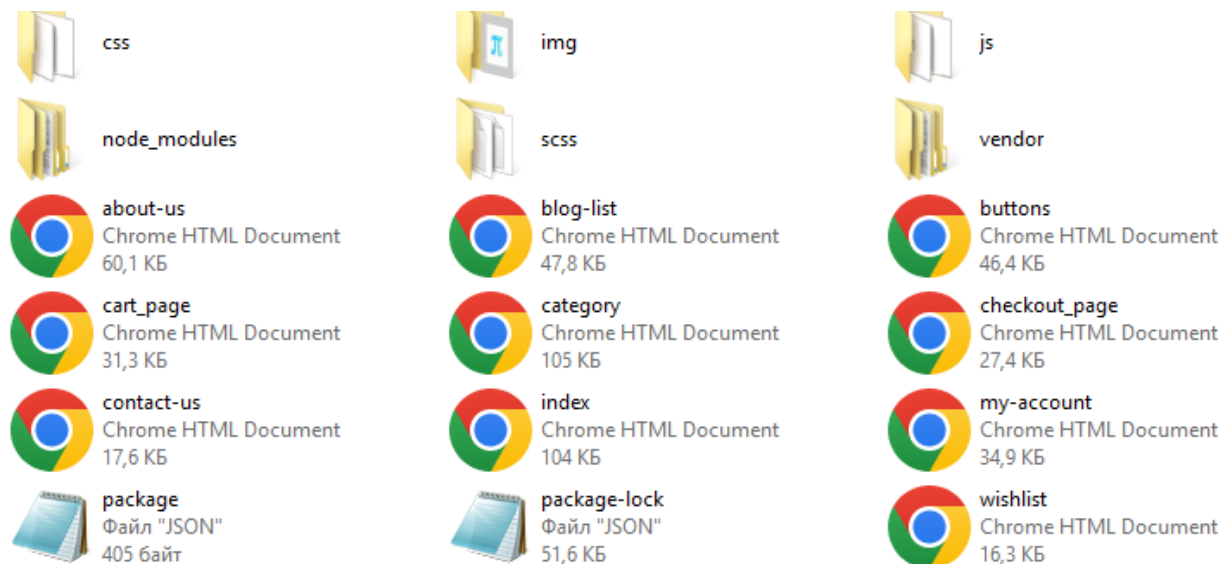


Рисунок 3.1 – Каталог вебсистеми

Що стосується макета сторінки сайту, використовувався тристоронній поділ, що складається з шапки сайту, блоку основного вмісту та нижнього розділу. Відповідні блоки, такі як «заголовок», «стаття» та «нижній колонтитул», були використані відповідно. Визнаючи їх універсальне використання на багатьох сторінках, розділи верхнього та нижнього колонтитулів було ізольовано та розміщено в окремих файлах.

Щоб надати сторінкам естетично привабливий дизайн, стилі CSS були ретельно створені та згодом експортовані у спеціальний файл. Частковий

уришок, що демонструє реалізацію дизайну нижнього колонтитула, можна побачити на лістингу 3.1.

Лістинг 3.1 - Стили футеру

```
body::before {
  font-family: "Source Code Pro", "SF Mono", Monaco, Inconsolata, "Fira Mono",
    "Droid Sans Mono", monospace, monospace;
  white-space: pre;
  display: block;
  padding: 1em;
  margin-bottom: 1em;
  border-bottom: 2px solid black;
  content: "Error: Undefined variable.\a \2577 \a 6 \2502          font:400 18px/24px $primary-font;\d\a \2502
}
```

Стили сайта для версии браузера компьютера имеют открий блок (лістинг 3.2).

Лістинг 3.2 – Стили media

```
@charset "UTF-8";
#homestah .tabs {
  float: left;
  width: 100%;
  margin: 0 0 30px;
}
#homestah .tabs .nav-tabs {
  border: none;
}
#homestah .tabs .nav-tabs .nav-item {
  position: relative;
}
@media (max-width: 543px) {
  #homestah .tabs .nav-tabs .nav-item {
    width: 70%;
    text-align: center;
    margin: 0 0 5px;
  }
}
```

На рисунку 3.2 показана структура всіх розроблених стилів та додатко залучених бібліотек.

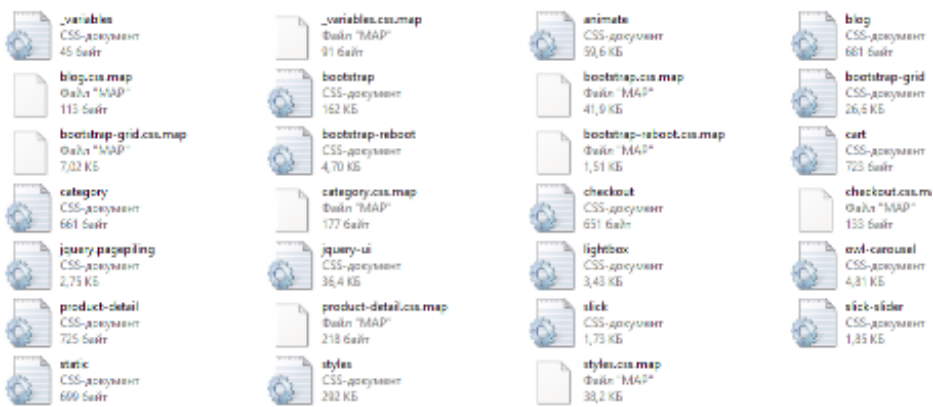


Рисунок 3.2 – Таблиці стилів вебсайту

3.2 Розробка клієнтської частини вебсистеми

Коли йдеться про сприйняття користувачами, заголовок вебсайту відіграє вирішальну роль, оскільки він служить початковою точкою взаємодії. Всього за 3-4 секунди заголовок повинен привернути увагу відвідувача і ефективно передати суть сайту. Якщо цього не зробити, це може призвести до втраченої можливості справити позитивне враження, особливо в умовах високої конкуренції в Інтернеті. Отже, шапка сайту є важливим і не обговорюваним елементом, який повинен бути присутнім на кожній сторінці.

На головній сторінці використовується комплексний дизайн заголовка, що включає різні елементи. Зазвичай вони включають фірмовий стиль, який охоплює логотип, назву або слоган, корпоративні кольори та загальний корпоративний стиль. Ще одним важливим компонентом є розділ контактної інформації. Цей блок важливий не тільки для відвідувачів сайту, але і для пошукових систем. Роботи пошукових систем сканують контактні дані, вказані в заголовку, щоб зібрати інформацію про місцезнаходження компанії, згодом посилаючись на неї на сторінці «Контакти». Зазвичай заголовок містить важливі контактні дані, такі як номери телефонів і адреси електронної пошти.

Невід'ємною особливістю шапки є горизонтальне верхнє меню. Служить багатьом цілям, він додає стилю сторінці, привертає увагу та пропонує користувачам зручну навігацію сайтом.

На додаток до основних елементів, згаданих вище, існують вторинні елементи, які можуть покращити функціональність заголовка. До них входять посилання на популярні соціальні мережі, які забезпечують зв'язок між вебсайтом і профілями компанії в соціальних мережах. Іншим корисним компонентом є кнопка зворотнього дзвінка, яка дозволяє відвідувачам легко ініціювати контакт з компанією. Меню гамбургерів є ще одним помітним елементом, який в першу чергу стосується мобільної версії заголовка сайту, але також може бути використаний на екранах настільних комп'ютерів.

Метою гамбургерного меню є збереження місця в заголовку для більш важливих пунктів. Цей компактний дизайн меню добре поєднується з мінімалістичною естетикою вебсайту та забезпечує зручне розміщення заголовка. Скріншот програмного коду на рисунку 3.3. Програмний код наведено на лістингу 3.3.

Лістинг 3.3 – Хедер сайту

```
<body class="index layout">
  <header class="header-area header-sticky text-center header-default">
    <div class="header-main-sticky">
      <div class="header-main">
        <div class="container">
          <div class="row-left float-left"><div class="theader-service">Служба клієнтів</div>
          </div>
          <div class="row-right float-right d-flex">
            <div class="theader-mail"><a href="mailto:info@colos.shop">info@colos.shop</a></div>
          </div>
        </div>
      </div>
      <div class="header-main-head">
        <div class="header-main">
          <div class="container">
            <div class="header-left float-left d-flex d-lg-flex d-md-block d-sm-block">
              <div class="language-wrapper toggle">
                <button type="button" class="btn text-capitalize dropdown-toggle"><img alt="ukr" height="12" width="12"/><span>Українська</span></button>
                <div id="language-dropdown" class="dropdown">
                  <ul>
                    <li><img alt="ukr" height="12" width="12"/><span>Українська</span></li>
                    <li><img alt="eng" height="12" width="12"/><span>English</span></li>
                  </ul>
                </div>
              </div>
              <div class="currency-wrapper toggle">
                <button type="button" class="btn text-capitalize dropdown-toggle"><span>UAH</span></button>
                <div id="currency-dropdown" class="dropdown">
                  <ul>
                    <li><span>UAH</span></li>
                    <li><span>USD Dollar</span></li>
                  </ul>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
```



Рисунок 3.3- Скріншот хедера

Розділ нижнього колонтитула вебсайту зазвичай містить інформацію, яка має бути легко доступною з будь-якої сторінки (лістинг 3.4). У результаті може виникнути часткове дублювання вмісту заголовка. Конкретний вміст, включений у нижній колонтитул, залежить від різних факторів, таких як тип сайту, тематика, особливості дизайну та вміст, присутній в інших блоках.

Одним з елементів, який може значно підвищити зручність для користувачів, є включення інтерактивної карти в нижній колонтитул. Це особливо цінно, коли точне місцезнаходження компанії є важливим для користувачів, наприклад, коли вони шукають щось неподалік від свого місця

проживання. Надаючи інтерактивну карту, вебсайт не тільки допомагає користувачам знайти компанію, але й зміцнює довіру, підтверджуючи фізичне існування компанії поза межами онлайн-сфери. Це може допомогти встановити довіру та сприяти почуттю надійності серед відвідувачів.

Лістинг 3.4 – Футер сайту

```
<!-- Footer -->
<div class="block-newsletter">
  <div class="parallax" data-source-url="img/banner/parallax.jpg" style="background-image: url(img/banner/parallax.jpg); background-position: 50% 50%;">
    <div class="container">
      <div class="fb-newsletter col-sm-7">
        <div class="text-uppercase">Підпишіться на наші новини</div>
      </div>
      <div class="block-content col-sm-5">
        <form method="post" action="contact-us.html">
          <div class="input-group">
            <input type="email" name="email" value="" placeholder="Email address..." required="" class="form-control">
            <div class="input-group-btn">
              <button class="btn btn-theme text-uppercase btn-primary" type="submit">Підписатися</button>
            </div>
          </div>
        </form>
      </div>
    </div>
  </div>
</div>

<!-- Footer -->
<div class="page-footer font-small footer-default">
  <div class="container text-center text-md-left">
    <div class="row">
      <div class="col-md-2 footer-cms footer-column">
        <div class="text-center">
          <div class="img"><img alt="Footer logo" width="50" height="50"></div>
          <div class="text">ColoShop</div>
        </div>
      </div>
      <div class="col-md-3 footer-column">
        <div class="text">
          <a href="#">Головна</a>
          <a href="#">Товари</a>
          <a href="#">Блог</a>
          <a href="#">Про нас</a>
        </div>
      </div>
      <div class="col-md-3 footer-column">
        <div class="text">
          <a href="#">Підписка на акції</a>
          <a href="#">Контакти</a>
        </div>
      </div>
    </div>
  </div>
</div>
```

Вигляд футера надан на рисунку 3.4. Блоки футери відображають поле для підписки. Також частково дублюється головне меню. Додаються ще специфічні посилання для особистого профілю (рис. 3.5).

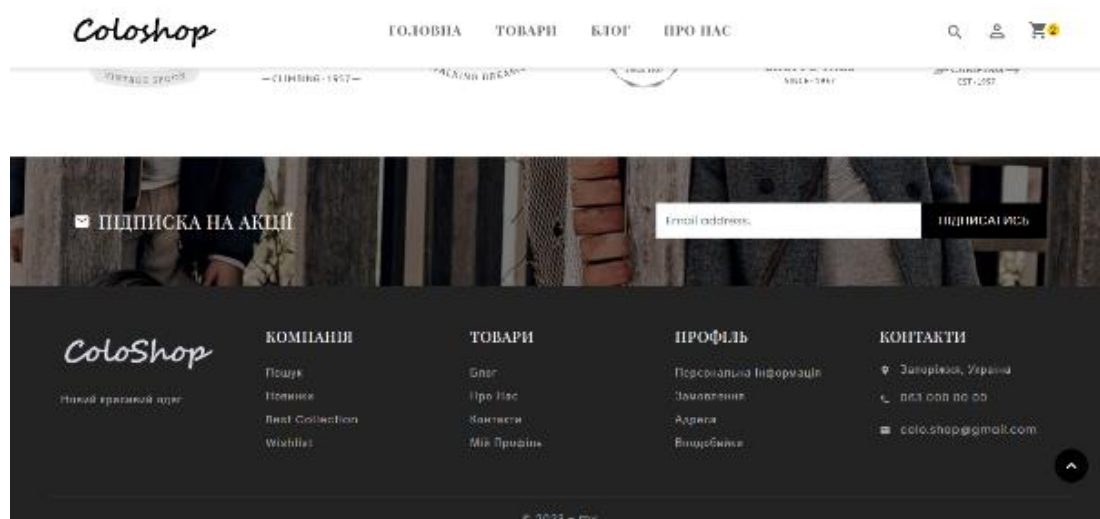


Рисунок 3.4 – Скріншот футера

Головна сторінка магазину складається з наступних елементів. Лого-банер - помітний і візуально яскравий банер у верхній частині сторінки. Навігаційне меню: чітке та інтуїтивно зрозуміле навігаційне меню у верхній частині сторінки, щоб допомогти відвідувачам легко досліджувати різні розділи вебсайту. Містить категорії, як «Головна», «Товари», «Блог» і «Про нас», щоб забезпечити швидкий доступ до відповідної інформації. екомендовані товари: на першій сторінці вибір рекомендованих товарів або колекцій одягу для вагітних. Використовуйте візуально привабливі зображення, короткі описи та помітні кнопки із закликом до дії, щоб спонукати відвідувачів досліджувати та робити покупки. Рекламні пропозиції (тренди): демонструють будь-які поточні акції, знижки чи спеціальні пропозиції на першій сторінці. Це може бути у формі банерів, слайдерів або спеціальних розділів, щоб привернути увагу та спонукати відвідувачів скористатися вигідними пропозиціями. Відгуки. Позитивні відгуки можуть переконати потенційних покупців у якості та вартості одягу для вагітних.

Результат розробки головної сторінки показан на наступних рисунках:

– слайдер (рис. 3.5);



Рисунок 3.5 – Слайдер

– відгуки (рис. 3.8).

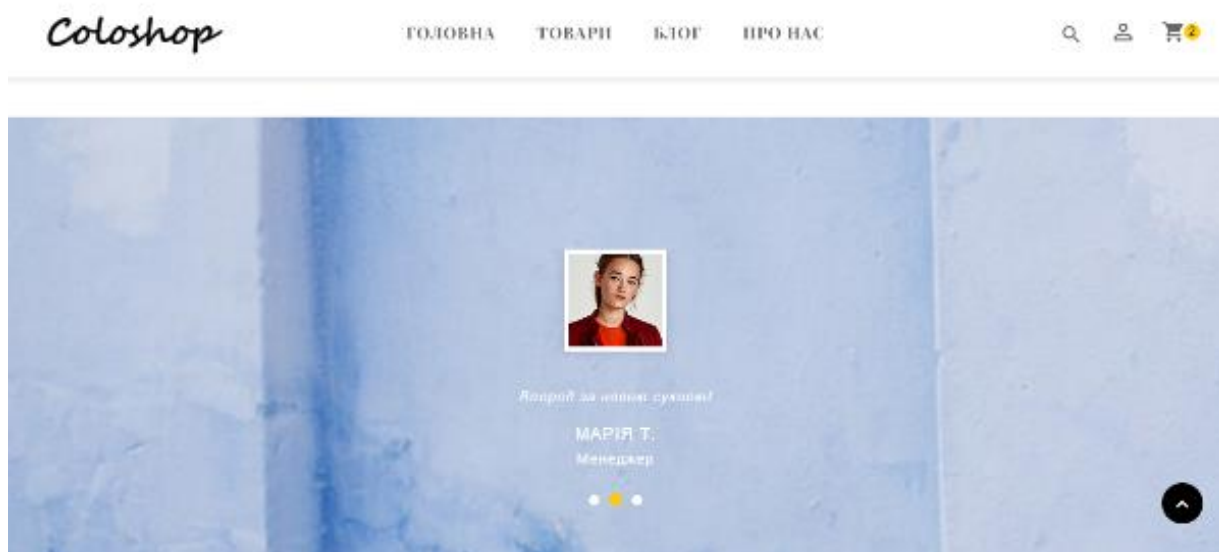


Рисунок 3.8 – Відгуки

Сторінка «Каталог» складається з наступних елементів (рис. 3.9). Розташування сітки: продукти в чистому та структурованому макеті сітки.

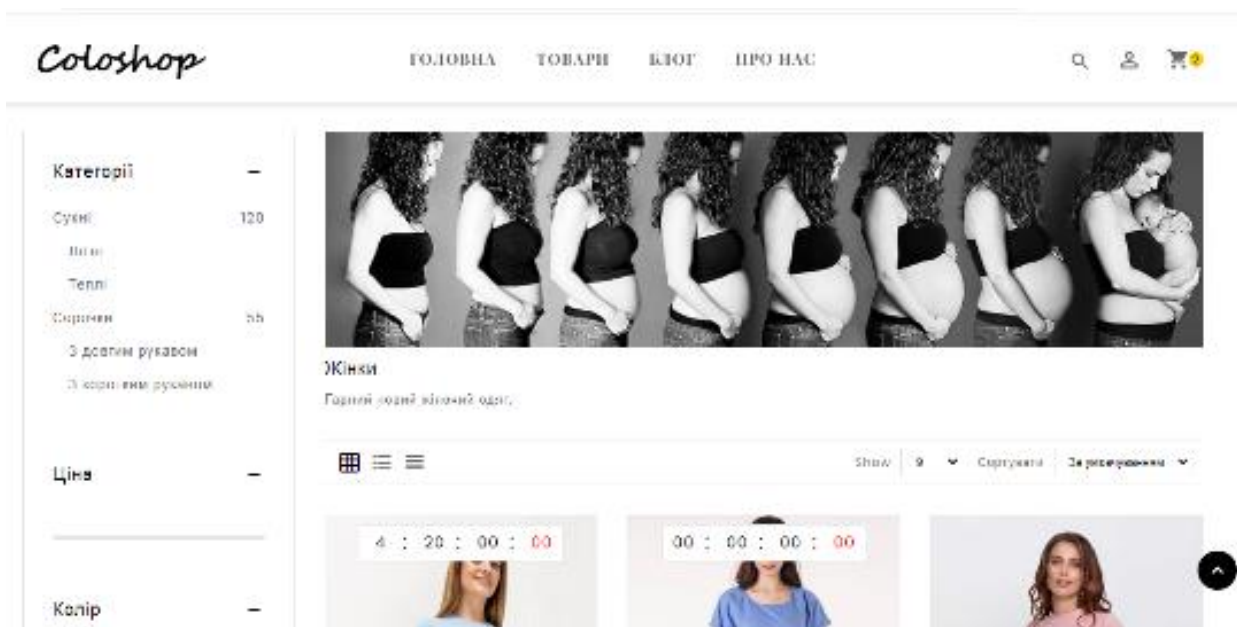


Рисунок 3.9 - Сторінка «Товари»

Кожен список продукту має містити чітке мініатюрне зображення, назву продукту, ціну та короткий опис або основні характеристики (рис. 3.10).



Рисунок 3.10 – Зміна вигляду товарів

Параметри сортування та фільтрації: функції сортування та фільтрації, щоб допомогти користувачам уточнити пошук і легко знайти потрібні продукти (сортування за ціною, розміром, кольором, стилем і наявністю) (рис. 3.11).



Рисунок 3.11 – Сортування товарів

Категорії та підкатегорії продуктів: розділіть продукти на відповідні категорії та підкатегорії, щоб полегшити навігацію (рис. 3.12). Посилання на сторінку з детальним описом товару (рис. 3.13).

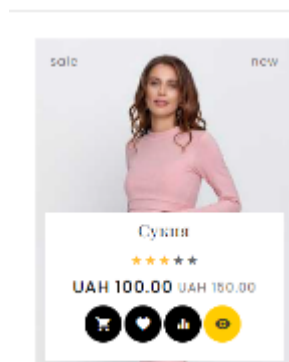


Рисунок 3.12 – Опис товару

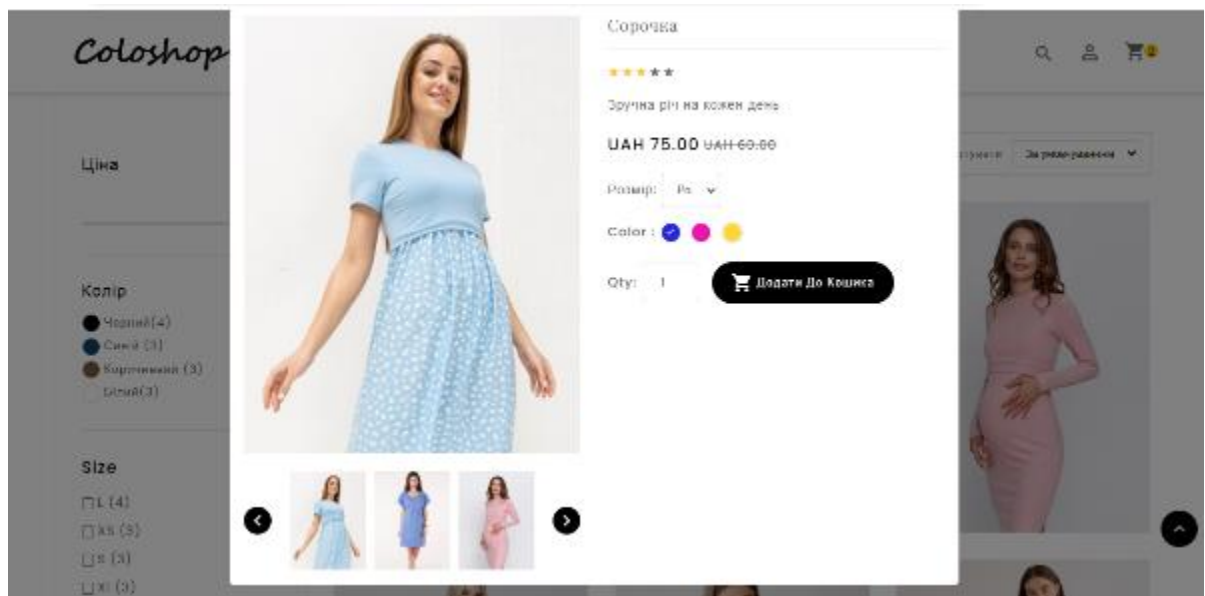


Рисунок 3.13 – Збільшення опису товару

Сторінка «Про нас» містить такі блоки (рис. 3.14). Логотип і фірмова символіка магазину.

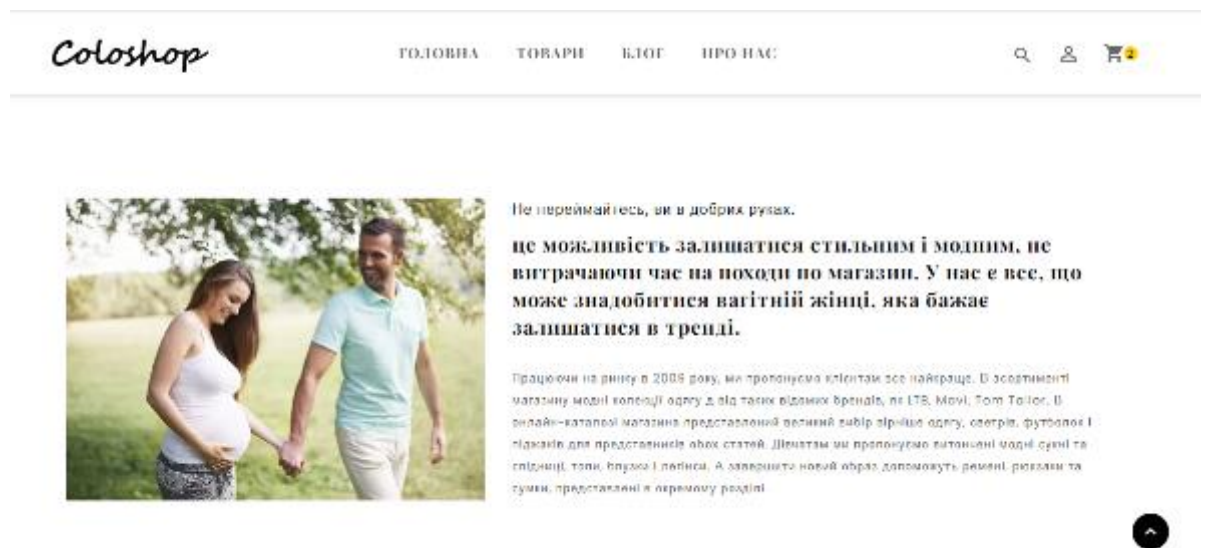


Рисунок 3.14 – Сторінка «Про нас»

Розроблен мотиваційний блок для клієнтів, щоб спонукати відвідувачів зробити покупку (рис. 3.15).



Рисунок 3.15 – Сторінка «Про нас» - мотивація

Команда працівників магазину (рис. 3.16).

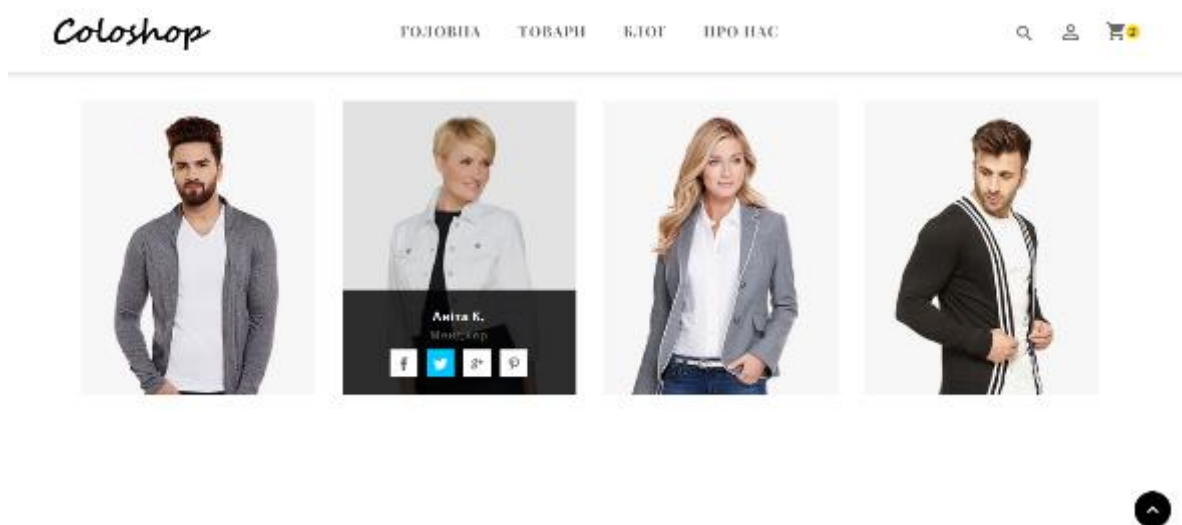


Рисунок 3.16 – Сторінка «Про нас» - команда

Після вибору товару можна натиснути на значок кошика. Товар автоматично додається до попереднього списку замовлення (рис. 3.17). Зменшений перегляд останнього замовлення, загальна сума та кнопки перейти до «Кошика» або до «Замовлення».



Рисунок 3.17 – Покупки

Форма кошика (рис. 3.18). Зображення продукту, назва продукту, ціна продукту, кількість, проміжний підсумок, видалити.

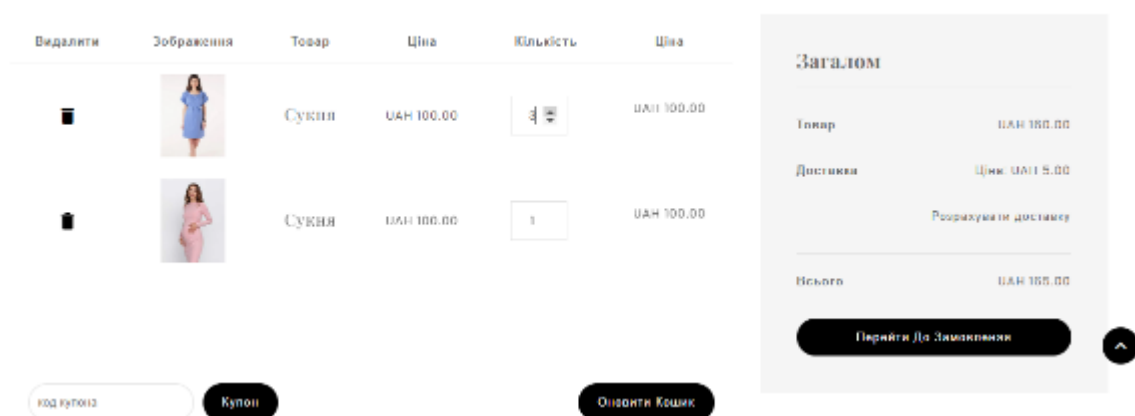


Рисунок 3.18 – Форма «Кошик»

Форма оформлення замовлення: адреса доставки, платіжна адреса, спосіб оплати (кредитна картка, PayPal тощо), інформація про покупця, розмістити замовлення (рис. 3.19).

JavaScript — популярна мова програмування, яка зазвичай використовується для розробки вебсайтів. Вона надає інтерактивні та динамічні елементи для покращення взаємодії з користувачем.

JavaScript надає можливості для створення анімації та візуальних ефектів на вебсторінках. Можна анімувати елементи, створювати переходи та

застосовувати такі ефекти, як згасання, слайди та прокручування, щоб зробити вебсайт більш привабливим і візуально привабливим.

Адреса доставки

Ім'я* Пов'нення*

Електронна пошта*

Адреса*

Місто*

Ваш кошик

Супина	20
Супина	20
Проміжок	-5
Всього:	35

☐ Кредитна картка
☐ Visa
☐ Mastercard

☐ Доставка
☐ Адреса доставки?
☐ Обробка

Рисунок 3.19 – Форма «Замовлення»

Бібліотеки та фреймворки сторонніх розробників: JavaScript має багату екосистему бібліотек і фреймворків, таких як React, Angular і Vue.js, які спрощують і оптимізують процес розробки. Ці фреймворки забезпечують багаторазові компоненти, ефективну обробку даних і розширену функціональність для створення складних вебдодатків. На рисунк 3.20 показані підключені Java Script.

```
<!-- Bootstrap core JavaScript
===== -->
<!-- Placed at the end of the document so the pages load faster -->
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/owl.carousel.min.js"></script>
<script src="js/custom.js"></script>
<script src="js/parallax.js"></script>
<script src="js/lightbox-2.6.min.js"></script>
<script src="js/ResizeSensor.min.js"></script>
<script src="js/theia-sticky-sidebar.min.js"></script>
<script src="js/inview.js"></script>
<script src="js/cookiealert.js"></script>
<script src="js/jquery.countdown.min.js"></script>
<script src="js/masonry.pkgd.min.js"></script>
<script src="js/imagesloaded.pkgd.min.js"></script>
<script src="js/jquery.zoom.min.js"></script>
<script src="js/jquery.lazy.min.js"></script>
```

Рисунок 3.20 - Java Script

Адмін панель

На сайті

Додати товар

Список товарів

ID товару	Код товару	Назва товару	Ціна	Категорія товару		
8	3543	Рубашка Time style	76	Рубашка		
12	3242	Футболка NIKE	30	Футболка		
13	2342	Футболка Adidas	32	Футболка		
15	3234	Майка Pull	15	Майка		
17	345	Майка puma	16	Майка		
35	23	Футболка Puma	27	Футболка		
40	54356	Рубашка Oodji	75	Рубашка		
41	4355	Рубашка Oodji	73	Рубашка		
42	243423	Рубашка Oodji	80	Рубашка		
43	3421	Брюки Oodji	57	Брюки		
44	6575	Брюки Levi's 511 Slim 5 Pocket SE Pewter Bl	66	Брюки		

Рисунок 3.22 – Редагування товару

Вона забезпечує платформу для налаштування та оптимізації вмісту, що дозволяє магазину ефективно спілкуватися зі своїми клієнтами (рис. 3.23).

Додати новий товар

Назва товару

Код товару

Вартість, \$

Категорія

Футболки

Фотографія товару

Виберіть файл

Файл не вибрано

Характеристика товару

Новинка

Так

Статус

Відображається

Зберегти

Рисунок 3.23 – Додавання нового товару

Щоб працювати з панеллю адміністратора, авторизовані адміністратори мають доступ до неї через безпечний процес входу. Після входу в систему їм відкривається інформаційна панель, яка надає огляд ключових показників, таких як ефективність продажів, статус замовлення та відвідуваність вебсайту. З інформаційної панелі адміністратори можуть переходити до різних розділів панелі для виконання певних завдань.

Панель адміністратора зазвичай розробляється з використанням веб-технологій, таких як PHP і MySQL. Він використовує комбінацію програмування серверної частини та дизайну зовнішньої частини, щоб створити зручну та інтуїтивно зрозумілу взаємодію з користувачем. Сервер забезпечує обробку даних, взаємодію з базою даних і бізнес-логіку, а фронтенд зосереджується на представленні інформації та забезпеченні зручного інтерфейсу.

Загалом панель адміністратора дає адміністраторам магазину можливість ефективно керувати та контролювати різні аспекти вебсистеми, забезпечуючи безперебійну роботу та безперебійний досвід покупок для клієнтів. Він служить потужним інструментом для моніторингу та оптимізації роботи онлайн-магазину, підвищення продуктивності та сприяння розвитку бізнесу.

3.4 Висновки за розділом 3

Щоб забезпечити чітку та організовану структуру, проект було розділено на окремі каталоги. Компоненти, пов'язані з сервером, перемістили до папки «node_modules», а стилі CSS знайшли своє місце у визначеній папці з відповідною назвою. Графічні елементи та зображення були консолідовані в папці «img». Папка «vendor» містить відповідні сторінки для бездоганної інтеграції. Папка «js» для сценаріїв JavaScript.

На головній сторінці використовується комплексний дизайн заголовка, що включає різні елементи. Зазвичай вони включають фірмовий стиль, який

охоплює логотип, назву або слоган, корпоративні кольори та загальний корпоративний стиль. Ще одним важливим компонентом є розділ контактної інформації. Цей блок важливий не тільки для відвідувачів сайту, але і для пошукових систем. Роботи пошукових систем сканують контактні дані, вказані в заголовку, щоб зібрати інформацію про місцезнаходження компанії, згодом посилаючись на неї на сторінці «Контакти». Зазвичай заголовок містить важливі контактні дані, такі як номери телефонів і адреси електронної пошти.

Невід'ємною особливістю шапки є горизонтальне верхнє меню. Служить багатьом цілям, він додає стилю сторінці, привертає увагу та пропонує користувачам зручну навігацію сайтом.

4 SEO-ОПТИМІЗАЦІЯ ТА ТЕСТУВАННЯ ВЕБСИСТЕМИ

4.1 SEO-просування сайту

Щоб підвищити видимість вебсайту серед потенційних покупців в Інтернеті, ми запровадили стратегію семантичного ядра. Семантичне ядро відноситься до ретельно підбраного та організованого списку запитів користувачів або пошукових термінів, пов'язаних із конкретною темою вебсайту. Ці терміни пошуку впорядковано в порядку спадання залежно від частоти їх появи.

Створення семантичного ядра служить багатьом цілям. По-перше, це допомагає нам зрозуміти мову та термінологію, яку використовують потенційні клієнти під час пошуку продуктів або інформації, пов'язаної з нашим вебсайтом. Аналізуючи та визначаючи найпоширеніші пошукові терміни, ми отримуємо цінну інформацію про ключові слова, які мають найбільший обсяг пошуку та релевантність.

Семантичне ядро виступає основою для пошукової оптимізації (SEO). Інтегруючи ідентифіковані ключові слова та фрази у вміст вебсайту, мета-теги, заголовки та інші відповідні елементи, ми збільшуємо видимість вебсайту на сторінках результатів пошуку (SERP). Це, у свою чергу, підвищує шанси вебсайту бути знайденим потенційними покупцями, які активно шукають продукти або інформацію, пов'язану з нашими пропозиціями [17].

Крім того, семантичне ядро допомагає нам визначати пріоритети та націлювати конкретні ключові слова в наших стратегіях створення вмісту та маркетингу. Зосереджуючись на пошукових термінах, які найчастіше використовуються, ми можемо розробити відповідний і привабливий вміст, який безпосередньо відповідає потребам та інтересам нашої цільової аудиторії. Це не тільки покращує загальну взаємодію з користувачем, але й підвищує ймовірність залучення органічного трафіку та потенційних клієнтів на вебсайт.

Крім того, семантичне ядро служить цінним ресурсом для розширення

та вдосконалення вмісту нашого вебсайту в майбутньому. Постійно відстежуючи та оновлюючи список запитів користувачів, ми можемо адаптувати нашу стратегію вмісту відповідно до змін уподобань клієнтів і тенденцій пошуку. Це дозволяє нам залишатися актуальними та підтримувати конкурентоспроможність на онлайн-ринку.

Таким чином, розробка семантичного ядра є вирішальним кроком для оптимізації видимості вебсайту та залучення потенційних покупців. Він надає цінну інформацію про поведінку користувачів під час пошуку, інформує наші зусилля щодо оптимізації пошукових систем, спрямовує створення вмісту та сприяє постійному вдосконаленню стратегії вмісту нашого вебсайту. Використовуючи потужність семантичного ядра, ми можемо покращити нашу онлайн-присутність і максимізувати наші шанси зв'язатися з потрібною аудиторією в потрібний час.

На рисунку 4.1 показан перший етап формування ключових слів за фразою «Магазин одягу для вагітних» (рис. 4.1).

Введіть ключову фразу та виберіть пошукову базу



Рисунок 4.1 – Пошук ключових слів

Результат популярних пошукових запитів за темою наведен на рисунку 4.2.

Проаналізувавши список часто використовуваних ключових фраз, стало очевидно, що для магазину одягу важливо вказати конкретні деталі, такі як види пропонованих товарів, місце розташування (місто), цінова політика магазину.

Ключова фраза: магазин жіночого одягу

Операції: G Україна 1.04K

Колонки: X

Фільтри: X

Експорт: X

Фільтрація та сортування недоступні вашому тарифному плану. [Покращити план](#) [Покращте тариф](#) та знайдете нові інсайти, фільтруючи й сортуючи цю таблицю.

#	Ключові фрази	Мова	Складність	Частотність	Вартість (\$)	Конкуренція в PPC, %	Результатів
1	www.магазин жіночого одягу	uk	36	1K	0,04	100	3
2	магазин жіночого одягу	uk	31	1K	0,07	100	1.39
3	інтернет магазин жіночого одягу	uk	31	720	0,08	100	1.19
4	інтернет магазин жіночого одягу ...	uk	3	480	0,04	100	580
5	жіночий одяг інтернет магазин	uk	43	480	0,08	100	2
6	інтернет магазин жіночого одягу ...	uk	10	390	0,06	100	797
7	інтернет магазин жіночого одягу на до...	uk	31	320	0,06	100	169
8	інтернет магазин жіночого одягу ...	uk	38	320	0,05	100	685
9	магазини жіночого одягу львів	uk	5	110	0,02	100	
10	інтернет магазин жіночого одягу великих р...	uk	24	90	0,03	100	

Рисунок 4.2 – Результати списку ключових слів «Магазин одягу для вагітних»

Результат пошуку вимушено було змінено для знаходження більш широкого кола цільової аудиторії (рис. 4.3).

Ключова фраза: магазин одягу для вагітних

Операції: G Україна 82

Колонки: X

Фільтри: X

Експорт: X

Фільтрація та сортування недоступні вашому тарифному плану. [Покращити план](#) [Покращте тариф](#) та знайдете нові інсайти, фільтруючи й сортуючи цю таблицю.

#	Ключові фрази	Мова	Складність	Частотність	Вартість (\$)	Конкуренція в PPC, %	Резу...
1	магазин одягу для вагітних	uk	11	70	0,05	95	
2	магазин одягу для вагітних львів	uk	5	20	0,02	100	
3	магазин одягу для вагітних львів	uk	5	20	0,04	100	
4	одяг для вагітних онлайн магазини	uk	32	20	0,05	100	
5	одяг для вагітних магазини	uk	24	10	0,00	88	
6	одяг для вагітних магазини	uk	33	10	0,00	0	
7	інтернет магазин одягу для вагіт...	uk	7	10	0,00	0	
8	одяг для вагітних онлайн магазини	uk	30	10	0,00	0	
9	одяг для вагітних інтернет магаз...	uk	34	10	0,07	100	
10	магазин одягу для вагітних	uk	10	10	0,00	39	

Рисунок 4.3 – Детальний опис ключових слів після уточнення

Ці ключові елементи допомагають потенційним клієнтам знайти відповідну інформацію, яку вони шукають, і дозволяють вебсайту ефективно орієнтуватися на цільову аудиторію.

Щоб оптимізувати внутрішню структуру вебсайту для пошукових систем, було прийнято стратегічний підхід, реалізувавши різні теги заголовків

від h1 до h4 по всьому вмісту сайту. Тег h1 спеціально використовується в розділі заголовка, де застосовано атрибут стилю «display: none». Ця техніка гарантує, що тег h1 залишається прихованим від відвідувачів сайту, але залишається доступним для пошукових систем для аналізу та інтерпретації.

Усі назви категорій і розділів належним чином розміщуються в тегах h2, що підкреслює їхнє значення в ієрархії сайту. Крім того, назви конкретних типів продуктів у кожній категорії призначаються тегам h3, що забезпечує додаткову організацію та ясність. Інші рівні тегів заголовків, наприклад h4, реалізуються за потреби, узгоджуючи зі структурою вмісту та надаючи додатковий контекст, коли це необхідно.

Щоб покращити оптимізацію сайту для пошукових систем, кожен тег зображення супроводжується атрибутом alt. Цей атрибут служить багатьом цілям, у тому числі виступаючи в якості альтернативного тексту, який відображається браузером у таких ситуаціях, як повільне мережеве з'єднання або коли зображення вимкнено користувачем. Додавання змістовного альтернативного тексту не тільки покращує доступність, але й надає пошуковим системам відповідну інформацію про зображення на сайті, сприяючи загальній оптимізації пошукових систем.

Крім того, заголовок вебсайту постійно доступний на кожній сторінці, забезпечуючи легку навігацію для відвідувачів, надаючи гіперпосилання на всі існуючі сторінки на сайті. Це полегшує безперебійне дослідження та заохочує користувачів отримувати доступ до різних розділів і вмісту.

Щоб оцінити та покращити пошукову оптимізацію сайту, було проведено комплексний аудит SEO з використанням онлайн-інструментів, спеціально розроблених для цієї мети. Ці інструменти дають цінну інформацію про різні аспекти ефективності сайту, включаючи аналіз ключових слів, аналіз зворотних посилань, оцінку швидкості сайту та загальну перевірку працездатності SEO. Використання цих інструментів дозволяє постійно вдосконалювати та налаштовувати стратегії оптимізації вебсайту для підтримки сильної присутності в Інтернеті [18].

Завдяки впровадженню цих методів оптимізації та проведенню регулярних SEO-аудитів вебсайт магазину одягу може досягти кращої видимості, залучити цільовий органічний трафік і, зрештою, збільшити шанси зв'язатися з потенційними клієнтами, які активно шукають відповідні продукти та інформацію (рис. 4.4).



Рисунок 4.4 – Видність вебсайту

Вебсистему було розміщено на безкоштовному хостінгу, це дало можливість аналізувати показники в динаміці (рис. 4.5).



Рисунок 4.5 – Розподіл фраз по позиціям

За допомогою онлайн аналізатора було виявлено сайти-конкуренти з типовою структурою (рис. 4.6). Це дало ясність, що вебсайт конкурентноспроможний.

Фільтрація та сортування недоступні вашому тарифному плану. [Покращити план](#) Покращте тариф та знайдете нові інсайти, фільтруючи й сортуючи цю таблицю.

#	☐	URL	Популярність в Facebook	Кількість фраз у топ 10
1	☐	https://supermama.top/uk	34	14
2	☐	https://tobestore.com	5.4K	13
3	☐	https://kaaba.ua/uk/market/odezhda-dlya-behemennykh	2	11
4	☐	https://orlop.ua/ua/market-molodym-mamam	0	8
5	☐	https://mamcity.ua/odezhda-dlya-behemennykh	0	8
6	☐	https://mamcity.ua	3K	8
7	☐	https://antoshka.ua/tovary-dlya-mam/molodym-mamam.html	0	8
8	☐	https://syron.com.ua	1.8K	7
9	☐	https://prom.ua/ua/Tovary-dlya-behemennykh-i-kormyashchih	0	7
10	☐	https://mamulya.lviv.ua	1.4K	7

Рисунок 4.6 – Сайти-конкуренти

Для подальшої оптимізації треба врахувати ключові фрази, які відсутні в семантичній системі вебсайту (рис. 4.7).

Відсутні ключові слова

Ключові фрази	Частотність	Вартість, \$	Рівень конкуренції
інтернет магазин одягу для вагітних	10	0,00	100
сайт для вагітних одяг	1	0,00	0
інтернет магазин одягу для вагітних в Україні	10	0,00	0
одяг для вагітних сайти	1	0,00	0
одяг для вагітних каталог	1	0,00	0
сайти для вагітних	10	0,00	100
одяг для вагітних магазин	10	0,00	0
одяг вагітних інтернет магазин	1	0,00	0
одяг для вагітних київ	10	0,00	0
інтернет одяг для вагітних	1	0,00	0

Рисунок 4.7 – Результати перевірки сайту

4.2 Тестування кросбраузерності сайту

Програми та вебсайти можуть по-різному працювати в різних браузерах. Ця відмінність виникає через унікальні надбудови, плагіни та відмінності між настільною та мобільною версіями кожного браузера. Щоб усунути ці розбіжності та зробити розробку більш універсальною сумісністю, кросбраузерне тестування є важливим.

Помилки між браузерами виникають через кілька факторів:

- самі вебпереглядачі можуть містити помилки або реалізовувати функції по-іншому, часто через конкуренцію, щоб отримати перевагу;
- різні браузери можуть пропонувати різні рівні підтримки технологічних функцій. Наприклад, певні функції JavaScript можуть не працювати в старіших браузерах;
- обмеження пристрою можуть вплинути на продуктивність і відображення вебсайту. Наприклад, вміст, розроблений для комп'ютера, може бути важко читати на мобільних пристроях.

Кросбраузерне тестування слід проводити після інших типів тестування, щоб переконатися, що будь-які виявлені проблеми зі сценарієм пов'язані саме з функціями веб-переглядача та не пропущені на попередніх етапах. Важливо зауважити, що помилки можуть проявлятися не в усіх браузерах. Слід також приділити увагу тестуванню різних комбінацій операційних систем і браузерів, зосередившись на найбільш часто використовуваних.

Процес міжбраузерного тестування починається з вибору браузерів для тестування. Як правило, клієнт вказує браузери, з якими він хоче, щоб його програмне забезпечення було сумісним. Однак розробник і тестувальник зобов'язані рекомендувати основний браузер на основі статистики подібних додатків і переваг браузера цільової аудиторії.

Глобальна статистика використання веб-переглядача, наприклад доступна на <http://gs.statcounter.com>, може стати вказівкою [13]. Деякі популярні браузери та їх відповідний відсоток використання:

- Google Chrome (остання версія): 65,89% у всьому світі;
- Safari: 16,65% у всьому світі (35,4% у США);
- Mozilla Firefox (остання версія): 4,26% у всьому світі;
- Microsoft Edge: 16,74% у всьому світі;
- Opera: 2,05% у всьому світі.

Вебсайт, який успішно пройшов кросбраузерне тестування та добре

функціонує на широкому спектрі пристроїв, може мати подовжений життєвий цикл і ефективніше охоплювати цільову базу споживачів.

Після кросбраузерного тестування вебсайт буде відображатися правильно:

- через різні конфігурації клієнтів;
- на екранах з різним дозволом;
- незалежно від конфігурації машини та сервера.

Забезпечуючи сумісність між браузерами та пристроями, вебсайт може забезпечити зручну взаємодію з користувачем, розширити охоплення та ефективно обслуговувати цільову аудиторію (рис. 4.8, 4.9).

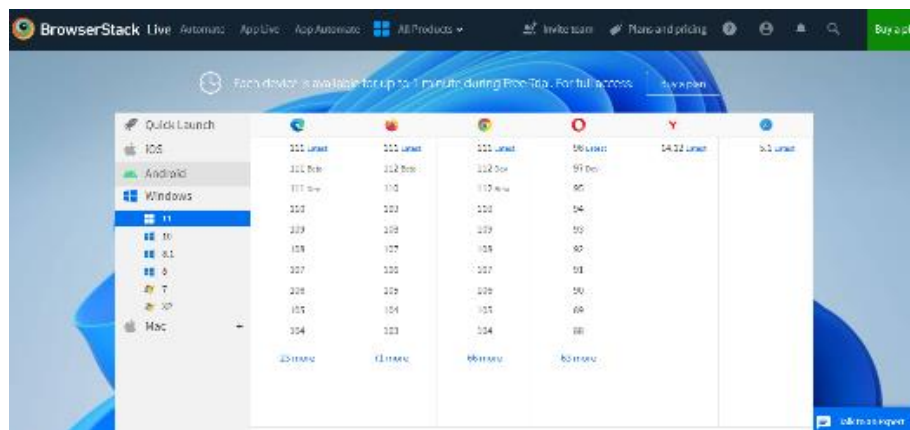


Рисунок 4.8 – Кросбраузерне тестування

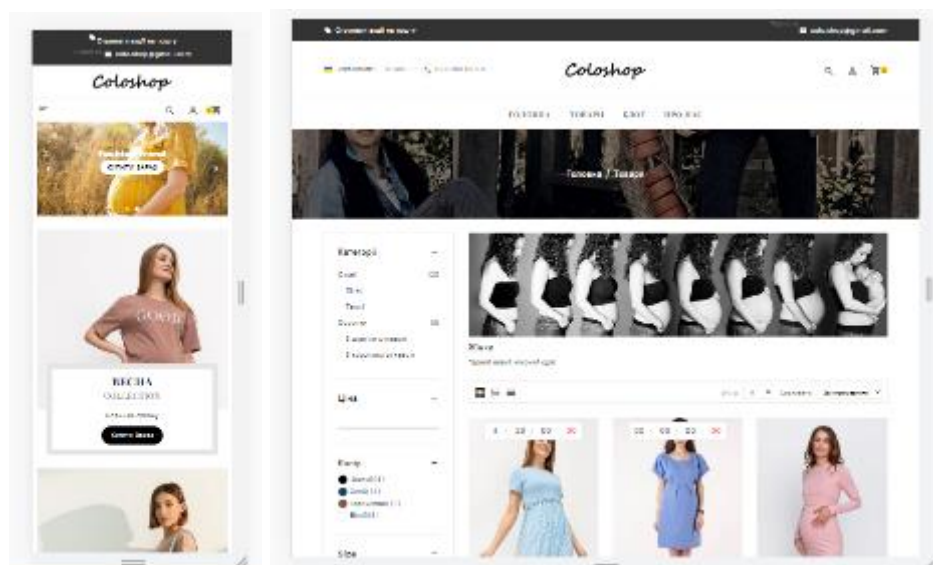


Рисунок 4.9 – Адаптованість вебсайту до різних розмірів екранів

4.3 Висновки за розділом 4

Оптимізація сайту має вирішальне значення для покращення взаємодії з користувачем, покращення видимості в пошуковій системі та максимізації ефективності вебсайту. Впроваджуючи різні методи оптимізації, підприємства можуть досягти наступних переваг. Покращена взаємодія з користувачем. Користувачі з більшою ймовірністю залишатимуться на сайті довше, вивчатимуть його пропозиції та перетворюватимуться на клієнтів.

Покращена видимість у пошуковій системі Використовуючи релевантні ключові слова, мета-теги та описові заголовки, компанії можуть збільшити свою видимість і залучити органічний трафік. Правильна оптимізація також передбачає оптимізацію структури вебсайту, створення якісних зворотних посилань і покращення продуктивності сайту, що сприяє підвищенню позицій у пошукових системах.

Швидкість вебсайту є критичним фактором для задоволення користувачів і рейтингу в пошукових системах. Оптимізація зображень, мінімізація коду, використання методів кешування та використання мереж доставки вмісту (CDN) можуть значно скоротити час завантаження сторінки. Швидше завантаження не тільки покращує взаємодію з користувачем, але й знижує показники відмов і підвищує коефіцієнт конверсії.

Оптимізація сайту для мобільних пристроїв гарантує, що він доступний і придатний для використання на екранах різних розмірів. Чуйний дизайн, адаптивні макети та спеціальні функції для мобільних пристроїв забезпечують безперебійну роботу користувача незалежно від пристрою, який використовується.

Сумісність із різними браузерами. Забезпечення стабільної роботи вебсайту в різних браузерах і пристроях має вирішальне значення для охоплення ширшої аудиторії. Кросбраузерне тестування та перевірки сумісності забезпечують безперебійну роботу користувача, незалежно від браузера чи пристрою, який використовується.

ВИСНОВКИ

В даній роботі була розроблена вебсистема магазину товарів для вагітних. Було проведено широкий аналіз сучасних інтернет-магазинів, щоб зрозуміти його вимоги та цілі. На основі цього аналізу було сформульовано детальну технічну специфікацію для керівництва процесом розробки.

Для вибору технологій, було проведено оцінку сучасних технологій, що призвело до прийняття стеку технологій, що складається з HTML, CSS, JavaScript і PHP. Ці технології були визнані придатними для впровадження бажаних функцій і забезпечення надійної та ефективної системи.

Процес розробки охоплював концептуальний дизайн, де були визначені сутності бази даних та їхні зв'язки, що призвело до створення моделі ER бази даних. Щоб створити основу для впровадження системи, було виконано як концептуальне, так і логічне проєктування.

Було розроблено зручний вебінтерфейс для полегшення взаємодії з базою даних, що призвело до створення функціонального вебсайту, який було розгорнуто в Інтернеті. Крім того, спеціально для менеджера була розроблена адміністративна панель для ефективного управління та оновлення вмісту сайту.

Система забезпечує швидкий пошук товарів і зручне оформлення замовлення завдяки інтуїтивно зрозумілому інтерфейсу. Зручність використання системи гарантує, що люди будь-якого походження можуть легко орієнтуватися та використовувати її функції.

Крім того, було проведено тестування та оптимізацію, щоб покращити онлайн-видимість і доступність системи. Це дозволяє користувачам отримувати доступ до системи з різних пристроїв і місць без будь-яких перешкод.

Підсумовуючи, цілі цього проєкту були успішно досягнуті. Розроблена вебсистема відповідає вимогам магазину одягу для вагітних, пропонуючи ефективну та зручну платформу для перегляду товарів, замовлення та

керування контентом. Система пройшла ретельне тестування та оптимізацію, що гарантує її ефективність та доступність для широкого кола користувачів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Апенько, Н. Аналіз технологій передачі даних / Наталія Апенько // ГРААЛЬ НАУКИ. – 2021. – С. 205–207.
2. Бондар, Д. Ю. Порівняльний аналіз методів створення веб-сторінок для підприємств телекомунікаційних послуг: thesis / Д. Ю. Бондар, М.М. Калюжний – [Б. м.], 2020.
3. Вавринчак, В. Д. Розробка веб-сайту інтернет-магазину для продажу одягу [Електрон. ресурс] : bachelor's thesis / Вавринчак Віталій Дмитрович, Vavrinchak Vitaliy. – [Б. м.], 2021. – Режим доступу: <http://elartu.tntu.edu.ua/handle/lib/35753>.
4. Полтавець, М. М. Сучасні технології інтернет-маркетингу [Електрон. ресурс] : thesis / Полтавець М. М. – [Б. м.], 2017. – Режим доступу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/7120>.
5. Рязанцев, І. О. Розробка веб-системи для створення інтернет-магазинів [Електрон. ресурс] : thesis / Рязанцев І. О. – [Б. м.], 2021. – Режим доступу: <https://openarchive.nure.ua/handle/document/16208>.
6. Слабінога, М. О. Розробка веб-додатків в контексті оптимізації їх швидкодії / М. О. Слабінога, С. В. Чабан // Таврійський науковий вісник. Серія: Технічні науки. – 2022. – № 3. – С. 63–69.
7. Цуненко, С. М. Особливості залучення клієнтів в Інтернет-магазин [Електрон. ресурс] : thesis / Цуненко С. М. – [Б. м.], 2015. – Режим доступу: <http://essuir.sumdu.edu.ua/handle/123456789/43802>.
8. Carneiro, N. Web 4 / Nehemias Carneiro. – [S. l.] : Independently published, 2019. – 64 p.
9. Cederholm, D. Applying CSS / Dan Cederholm // Web Standards Solutions. – Berkeley, CA, 2004. – P. 137–151.
10. CSS / Mario Heiderich [et al.] // Web Application Obfuscation. – [S. l.], 2011. – P. 125–149.
11. Fraser, S. My SQL / Sophia Fraser. – Chandni Chowk, Delhi : Global

Media, 2009. – 436 p.

12. Gans, M. HTML and CSS / Maya Gans, Toby Hodges, Greg Wilson // JavaScript for Data Science. – [S. l.], 2020. – P. 43–54.

13. Manganari, E. E. Store atmosphere in web retailing / Emmanouela E. Manganari, George J. Siomkos, Adam P. Vrechopoulos // European Journal of Marketing. – 2009. – Vol. 43, no. 9/10. – P. 1140–1153.

14. php. PHP Manual, Volume 1 / php. – [S. l.] : Iuniverse Inc, 2000. – 480 p.

15. php. PHP Manual, Volume 2 / php. – [S. l.] : Iuniverse Inc, 2000. – 524 p.

16. REACT [Electronic resource] / Mohit Upadhyay [et al.] // DAC '22: 59th ACM/IEEE Design Automation Conference, San Francisco California. – New York, NY, USA, 2022. – Access mode: <https://doi.org/10.1145/3489517.3530406>.

17. Teeter, B. R. A Cookbook for Java Script / B. R. Teeter // IEEE Multimedia. – 1999. – Vol. 6, no. 1. – P. 90.

18. Uzayr S. b. The Bootstrap Framework / Sufyan bin Uzayr // Bootstrap. – Boca Raton, 2022. – P. 19–70.

ДОДАТОК А**Програмний код роботи з замовленнями**

```

import React, { useState } from 'react';

const OrderForm = () => {
  const [name, setName] = useState("");
  const [surname, setSurname] = useState("");
  const [phone, setPhone] = useState("");
  const [email, setEmail] = useState("");
  const [quantity, setQuantity] = useState(0);
  const [price, setPrice] = useState(0);
  const [totalAmount, setTotalAmount] = useState(0);
  const [deliveryMethod, setDeliveryMethod] = useState("");
  const [paymentType, setPaymentType] = useState("");

  const handleSubmit = (event) => {
    event.preventDefault();

    // Create a JSON object with the order data
    const order = {
      name,
      surname,
      phone,
      email,
      quantity,
      price,
      totalAmount,
      deliveryMethod,
      paymentType,
    };

    // Reset form fields
    setName("");
    setSurname("");
    setPhone("");
    setEmail("");
    setQuantity(0);
    setPrice(0);
    setTotalAmount(0);
    setDeliveryMethod("");
    setPaymentType("");
  };

  return (
    <form onSubmit={handleSubmit}>

```

```

        <input type="text" value={name} onChange={(e) =>
setName(e.target.value)} placeholder="Name" required />
        <input type="text" value={surname} onChange={(e) =>
setSurname(e.target.value)} placeholder="Surname" required />
        <input type="tel" value={phone} onChange={(e) =>
setPhone(e.target.value)} placeholder="Phone" required />
        <input type="email" value={email} onChange={(e) =>
setEmail(e.target.value)} placeholder="Email" required />
        <input type="number" value={quantity} onChange={(e) =>
setQuantity(e.target.value)} placeholder="Quantity" required />
        <input type="number" value={price} onChange={(e) =>
setPrice(e.target.value)} placeholder="Price" required />
        <input type="number" value={totalAmount} onChange={(e) =>
setTotalAmount(e.target.value)} placeholder="Total Amount" required />
        <select value={deliveryMethod} onChange={(e) =>
setDeliveryMethod(e.target.value)} required>
            <option value="">Select Delivery Method</option>
            <option value="method1">Delivery Method 1</option>
            <option value="method2">Delivery Method 2</option>
            { /* Add more options if needed */ }
        </select>
        <select value={paymentType} onChange={(e) =>
setPaymentType(e.target.value)} required>
            <option value="">Select Payment Type</option>
            <option value="type1">Payment Type 1</option>
            <option value="type2">Payment Type 2</option>
            { /* Add more options if needed */ }
        </select>
        <button type="submit">Submit</button>
    </form>
);
};

```

```
export default OrderForm;
```

```
////////////////////////////////////
```

```

<?php
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $name = $_POST["name"];
    $surname = $_POST["surname"];
    $phone = $_POST["phone"];
    $email = $_POST["email"];
    $quantity = $_POST["quantity"];

```

```

$price = $_POST["price"];
$totalAmount = $_POST["total_amount"];
$deliveryMethod = $_POST["delivery_method"];
$paymentType = $_POST["payment_type"];

```

```

// Create a database connection (similar to step 1)

```

```

// Prepare the SQL statement

```

```

$stmt = $conn->prepare("INSERT INTO orders (name, surname, phone,
email, quantity, price, total_amount, delivery_method, payment_type) VALUES
(names, price, total, type)");

```

```

$stmt->bind_param("ssssiddss", $name, $surname, $phone, $email,
$quantity, $price, $totalAmount, $deliveryMethod, $paymentType);

```

```

// Execute the SQL statement

```

```

if ($stmt->execute()) {
    echo "Order placed successfully";
} else {
    echo "Error: " . $stmt->error;
}

```

```

$stmt->close();

```

```

}

```

```

$conn->close();

```

```

?>

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

import React, { useState } from 'react';

```

```

const LoginForm = () => {
    const [email, setEmail] = useState("");
    const [password, setPassword] = useState("");

```

```

    const handleLogin = (event) => {
        event.preventDefault();

```

```

        setEmail("");
        setPassword("");
    };

```

```

    return (
        <form onSubmit={handleLogin}>

```

```

        <input type="email" value={email} onChange={(e) =>
setEmail(e.target.value)} placeholder="Email" required />
        <input type="password" value={password} onChange={(e) =>
setPassword(e.target.value)} placeholder="Password" required />
        <button type="submit">Login</button>
    </form>
  );
};

const SignupForm = () => {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");

  const handleSignup = (event) => {
    event.preventDefault();

    setEmail("");
    setPassword("");
  };

  return (
    <form onSubmit={handleSignup}>
      <input type="email" value={email} onChange={(e) =>
setEmail(e.target.value)} placeholder="Email" required />
      <input type="password" value={password} onChange={(e) =>
setPassword(e.target.value)} placeholder="Password" required />
      <button type="submit">Sign Up</button>
    </form>
  );
};

const RegistrationForm = () => {
  const [name, setName] = useState("");
  const [surname, setSurname] = useState("");
  const [phone, setPhone] = useState("");
  const [email, setEmail] = useState("");

  const handleRegistration = (event) => {
    event.preventDefault();

    // Reset form fields
    setName("");
    setSurname("");
    setPhone("");

```

```

        setEmail("");
    };

    return (
        <form onSubmit={handleRegistration}>
            <input type="text" value={name} onChange={(e) =>
setName(e.target.value)} placeholder="Name" required />
            <input type="text" value={surname} onChange={(e) =>
setSurname(e.target.value)} placeholder="Surname" required />
            <input type="tel" value={phone} onChange={(e) =>
setPhone(e.target.value)} placeholder="Phone" required />
            <input type="email" value={email} onChange={(e) =>
setEmail(e.target.value)} placeholder="Email" required />
            <button type="submit">Register</button>
        </form>
    );
};

export { LoginForm, SignupForm, RegistrationForm };

```

ДОДАТОК Б
Слайди презентації

РОЗРОБКА ВЕБСИСТЕМИ МАГАЗИНУ ТОВАРІВ ДЛЯ ВАГІТНИХ.

Виконав: студент гр. КНТ-139
Керівник:

Дар'я САЙНІДІ
Наталія КУЛИКОВСЬКА

Рисунок Б.1 – Слайд 1

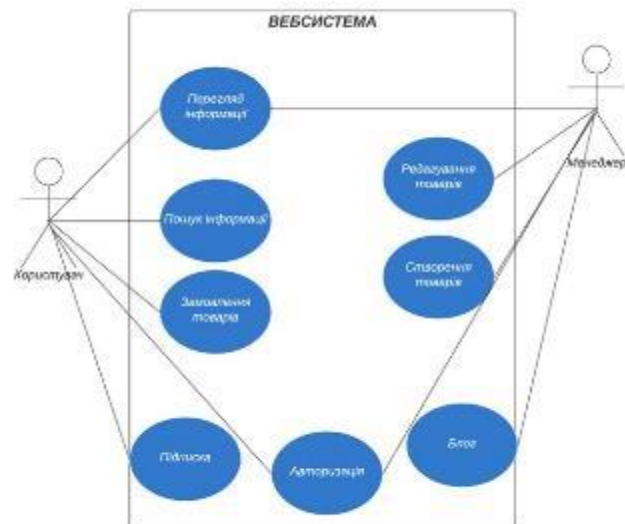
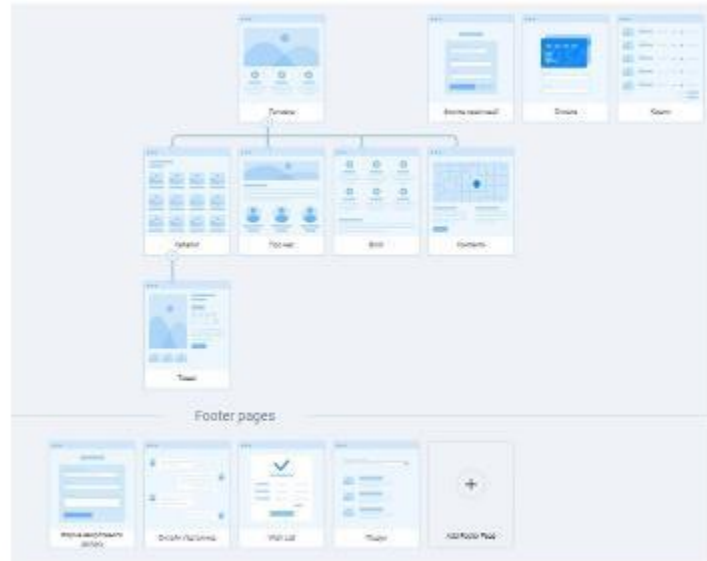


Рисунок Б..2 – Слайд 2

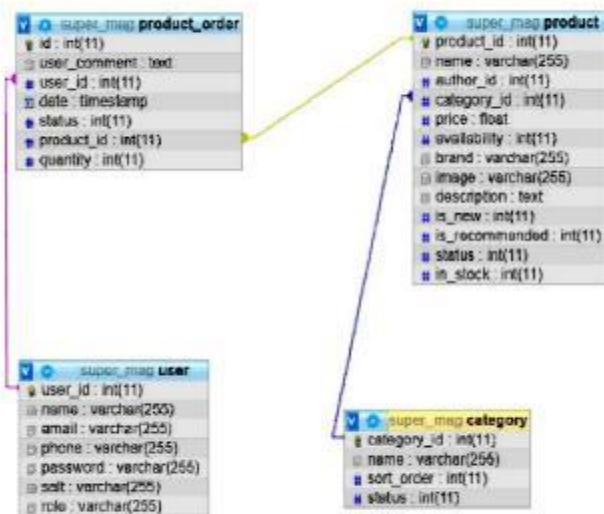
Структура вебсистеми магазину товарів для вагітних



3/13

Рисунок Б.3 – Слайд 3

База даних



4/13

Рисунок Б.4 – Слайд 4

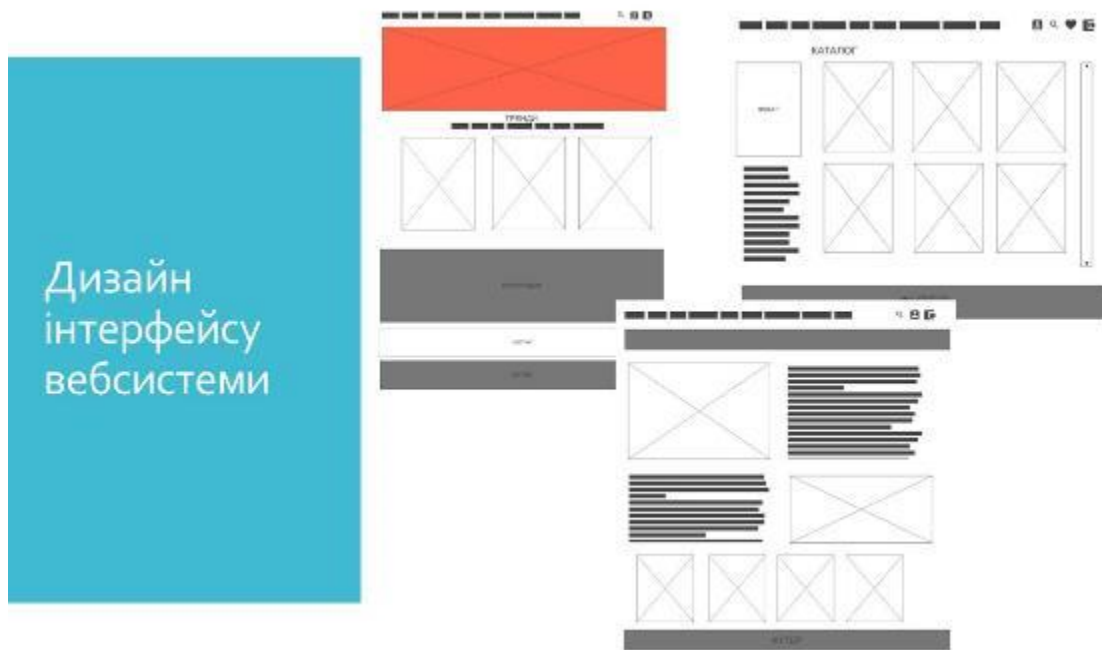


Рисунок Б.5 – Слайд 5

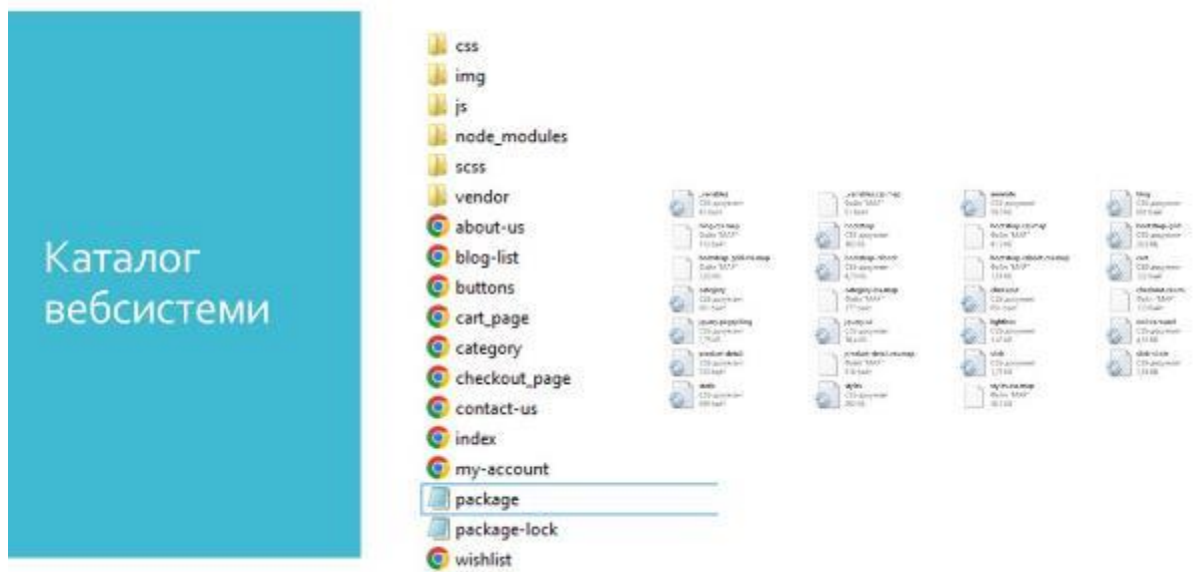
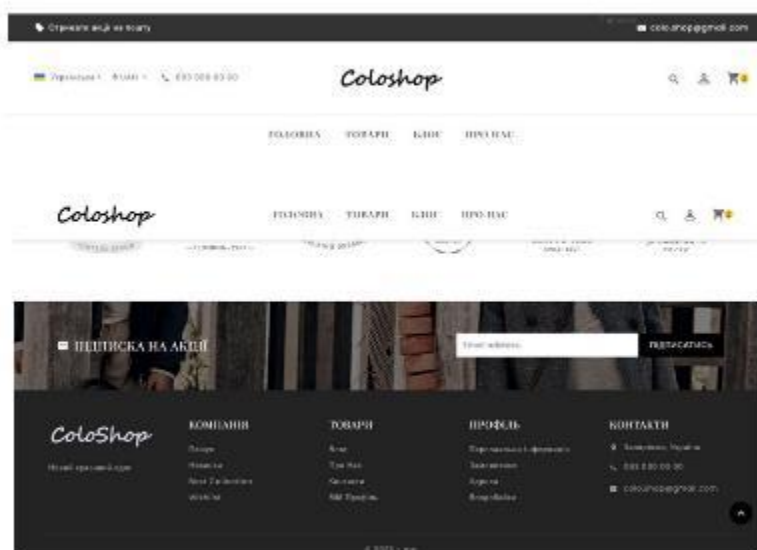
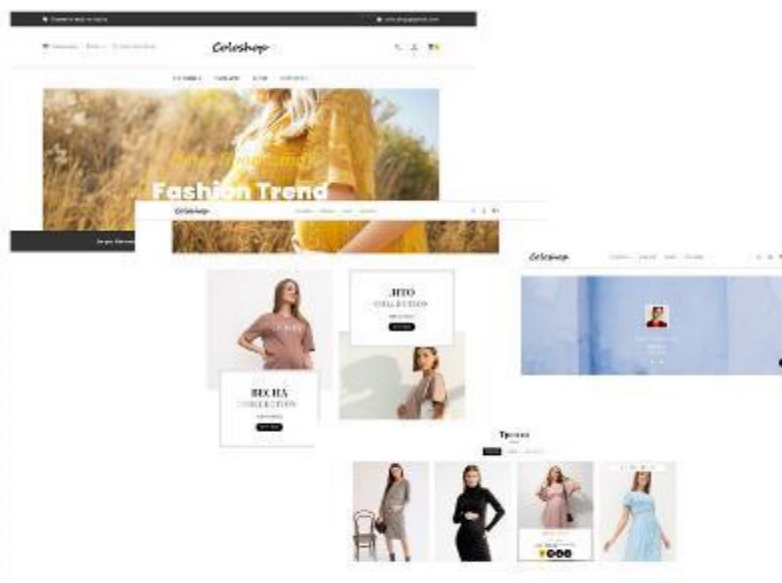


Рисунок Б.6 – Слайд 6



7/12

Рисунок Б.7 – Слайд 7



8/12

Рисунок Б.8 – Слайд 8

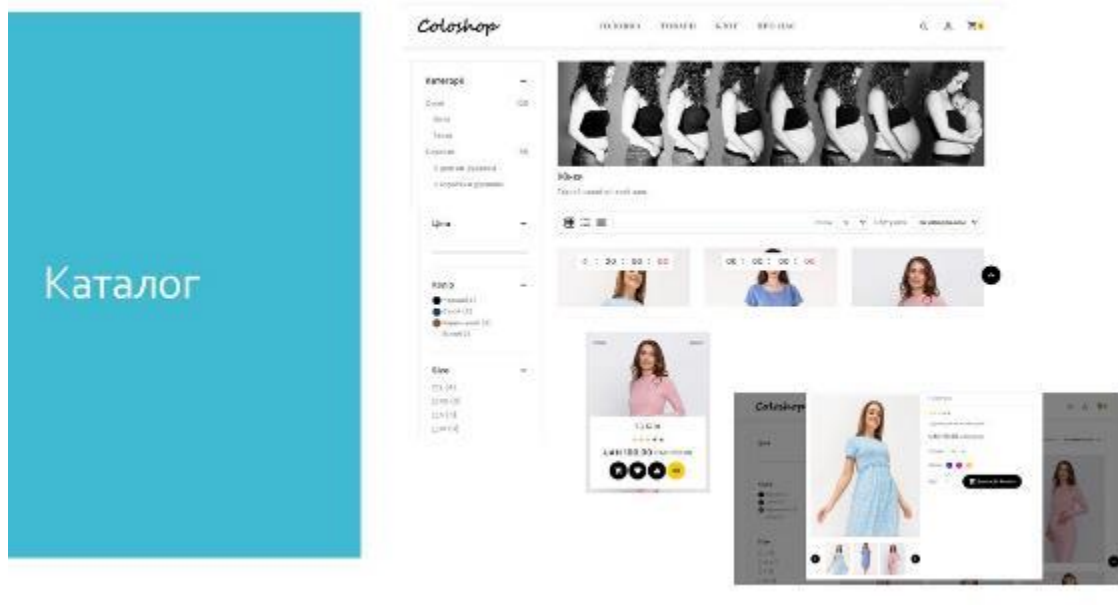


Рисунок Б.9 – Слайд 9

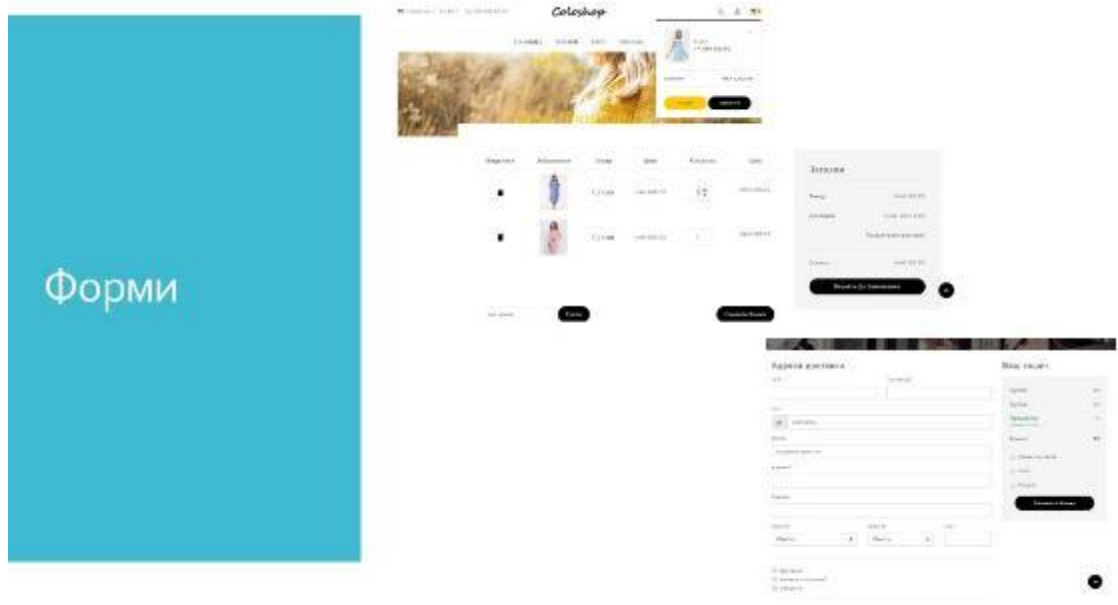


Рисунок Б.10 – Слайд 10



Рисунок Б.11 – Слайд 11

Висновки

Метою дипломної роботи за темою розробка вебсистеми магазину товарів для вагітних досягнена.

Задачі, які були вирішені в ході роботи:

- аналіз сучасних вебсистем;
- розробка бази даних;
- розробка дизайну сторінок;
- розробка вебсайту;
- SEO- оптимізація.

13/13

Рисунок Б.12 – Слайд 12