

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка вебсайту дитячого магазину "Моє дитя"

Виконав: студент IV курсу, групи СНс-41

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис)

Доберчак О. В.

(прізвище та ініціали)

Керівник

(підпис)

Дмитроца Л. П.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Марценко С. В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Микитишин А. Г.

(прізвище та ініціали)

Тернопіль
2024

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.
(підпис) (прізвище та ініціали)

«__» червня 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Доберчаку Олексію Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка вебсайту дитячого магазину "Моє дитя"

Керівник роботи Дмитроца Леся Павлівна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «29» квітня 2024 року № 4/7-470

2. Термін подання студентом завершеної роботи 24 червня 2024р.

3. Вихідні дані до роботи Структура сайту, перелік необхідної інформації яка буде розміщена на сайт

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

Вступ. Розділ 1. Аналіз вимог і постановка завдань для розробки веб-сайту "Моє дитя". 1.1

Аналіз вебсайтів дитячих магазинів. 1.2 Формулювання вимог до функціональності та

дизайну. 1.3 Вибір технологій для вебсайту "Моє дитя". 1.4 Вибір хостингу для вебсайту

"Моє дитя". 1.5 Огляд робочого простору обраного програмного забезпечення. 1.6 Висновок

до першого розділу. Розділ 2. Проектування вебсайту "Моє дитя". 2.1 Розробка архітектури

системи. 2.2 Моделювання бд для вебсайту. 2.3 Опис інтерфейсу. 2.4 Розробка файлової

структури проекту. 2.5 Дизайн інтерфейсу та вибір кольорової схеми. 2.6 Взаємодія з

Користувачем та забезпечення зручності навігації. 2.7 Висновок до другого розділу. Розділ 3.

3.1 Розробка серверної частини на PHP. 3.2 Створення клієнтської частини за допомогою JS,

Html CSS. 3.3 Інтеграція бд MySQL. 3.4 Налаштування серверного оточення та хостинг. 3.5

Тестування функціональності та забезпечення якості. 3.6 Висновки до третього розділу.

Розділ 4. Безпека життєдіяльності, основи охорони праці. 4.1 До лікарська медична допомога

при захворюваннях, травмах та в умовах надзвичайних ситуацій. 4.2 Вимоги безпеки до

робочих місць для виконання робіт. Висновок. Перелік джерел. Додатки

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

6. Консультанти розділів роботи

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

7. Дата видачі завдання 29 січня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	30.01.2024	Виконано
2.	Підбір джерел про вебсайт дитячого магазину “Моє дитя”	31.01.2024-03.02.2024	Виконано
3.	Опрацювання джерел по темі кваліфікаційної роботи	04.02.2024-12.02.2024	Виконано
4.	Виконання дослідження щодо вебсайту “Моє дитя”	07.02.2024-11.02.2024	Виконано
5.	Оформлення розділу «Аналіз вимог та формулювання завдань для розробки вебсайту “Моє дитя” »	03.06.2024-05.06.2024	Виконано
6.	Оформлення розділу «Проектування вебсайту “Моє дитя”»	06.06.2024-08.06.2024	Виконано
7.	Оформлення розділу «Реалізація вебсайту “Моє дитя”»	09.06.2024-11.06.2024	Виконано
8.	Виконання завдання до підрозділу «Безпека життєдіяльності»	12.06.2024-13.06.2024	Виконано
9.	Виконання завдання до підрозділу «Основи охорони праці»	14.06.2024-15.06.2024	Виконано
10.	Оформлення кваліфікаційної роботи	16.06.2024-17.06.2024	Виконано
11.	Нормоконтроль	18.06.2024-19.06.2024	
12.	Перевірка на плагіат	20.06.2024	
13.	Попередній захист кваліфікаційної роботи		
14.	Захист кваліфікаційної роботи		

Студент

(підпис)

Доберчак О. В.

(прізвище та ініціали)

Керівник роботи

(підпис)

Дмитроца Л. П.

(прізвище та ініціали)

АНОТАЦІЯ

Розробка вебсайту дитячого магазину “Моє дитя” // Кваліфікаційна робота освітнього рівня «Бакалавр» // Доберчак Олексій Володимирович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп’ютерно-інформаційних систем і програмної інженерії, кафедра комп’ютерних наук, група СНс-41 // Тернопіль, 2024 // С. 64, рис. – 9, табл. – 1, додат. – 2.

Ключові слова: інтернет-магазин, дитячі товари, база даних, хостинг, сервер, вебсайт, моє дитя.

Кваліфікаційна робота присвячена розробці вебсайту "Моє дитя" для надання інформації про асортимент дитячих товарів та здійснення покупок онлайн, що підвищить задоволеність клієнтів магазину та збільшить продаж дитячих товарів.

В першому розділі кваліфікаційної роботи було проведено аналіз вебсайтів дитячих товарів. Описано функціональні та дизайнерські вимоги, розглянуто вибір технологій для розробки, що забезпечують динамічність і зручність користування.

В другому розділі кваліфікаційної роботи розглянуто проектування вебсайту “Моє дитя”. Розроблено архітектуру системи, яка базується на вкладочній структурі, що забезпечує зручність навігації та доступність основних функцій для користувачів. Було описано деталі моделювання бази даних та дизайну інтерфейсу, що включає кольорові схеми і взаємодію з користувачами.

В третьому розділі кваліфікаційної роботи здійснено повну реалізацію вебсайту на базі сучасних веб-технологій. Серверну частину реалізовано за допомогою мови програмування PHP, що забезпечило надійну обробку запитів від клієнтів та взаємодію з базою даних MySQL. Також проведено тестування функціональності та забезпечення якості, що гарантує стабільну роботу і безпеку системи.

ANNOTATION

Website Development of the Children's Store "My Baby" // Qualification work of the educational level "Bachelor" // Doberchak Olexiy Volodymyrovych // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SNs-41 // Ternopil, 2024 // P. 64, fig. - 9, tabl. - 1, annexes. – 2.

Keywords: online store, children's goods, database, hosting, server, website, my child.

The qualification work is dedicated to the development of the "My Child" website to provide information about the range of children's products and make purchases online, which will increase the satisfaction of the store's customers and increase the sale of children's products.

In the first section of the qualification work, an analysis of websites for children's products was carried out. Functional and design requirements are described, the choice of technologies for development, which ensure dynamism and ease of use, is considered.

In the second section of the qualification work, the design of the "My Child" website is considered. The system architecture is developed, which is based on the tab structure, which ensures the convenience of navigation and the availability of basic functions for users. Details of database modeling and interface design were described, including color schemes and user interaction.

In the third section of the qualification work, the full implementation of the website was carried out on the basis of modern web technologies. The server part was implemented using the PHP programming language, which ensured reliable processing of requests from clients and interaction with the MySQL database. Functionality testing and quality assurance were also carried out, which guarantees the stable operation and security of the system.

ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ

CSS – Cascading Style Sheets, технологія для стилізації HTML-документів.

HTML – HyperText Markup Language, стандартна мова розмітки для створення веб-сторінок.

JavaScript (JS) – динамічна мова програмування, що використовується для створення інтерактивного контенту на веб-сторінках.

jQuery – швидка, маленька і багатофункціональна JavaScript-бібліотека, яка спрощує обробку HTML-документів, обробку подій, анімацію і Ajax.

MySQL – популярна система керування базами даних з відкритим кодом, що використовується для управління даними в інтернет-магазині.

PHP – мова скриптів загального призначення, яка особливо підходить для веб-розробки та може бути вбудована в HTML.

SQL – Structured Query Language, мова для управління і обробки даних в реляційних базах даних.

SSL – Secure Sockets Layer, криптографічний протокол для забезпечення безпечної передачі даних через Інтернет.

XAMPP – безкоштовний і відкритий кросплатформенний веб-серверний пакет, що складається з Apache, MySQL, PHP і Perl.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ВИМОГ І ПОСТАНОВКА ЗАВДАНЬ ДЛЯ РОЗРОБКИ ВЕБСАЙТУ “МОЄ ДИТЯ”	10
1.1 Аналіз вебсайтів дитячих магазинів	10
1.2 Формулювання вимог до функціональності та дизайну.....	12
1.3 Вибір технологій для вебсайту “Моє дитя”	13
1.4 Вибір хостингу для вебсайту “Моє дитя”.....	15
1.5 Огляд робочого простору обраного програмного забезпечення.....	17
1.6 Висновок до першого розділу	19
РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБСАЙТУ “МОЄ ДИТЯ”	20
2.1 Розробка архітектури системи	20
2.2 Моделювання бази даних для вебсайту.....	23
2.3 Опис інтерфейсу дитячого-магазину “Моє дитя”	26
2.4 Розробка файлової структури проекту	28
2.5 Дизайн інтерфейсу та вибір кольорової схеми.....	29
2.6 Взаємодія з користувачем та забезпечення зручності навігації	34
2.7 Висновок до другого розділу	35
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ВЕБСАЙТУ “МОЄ ДИТЯ”.....	36
3.1 Розробка серверної частини на PHP.....	36
3.2 Створення клієнтської частини за допомогою JavaScript, HTML та CSS	41
3.3 Інтеграція бази даних MySQL	44
3.4 Налаштування серверного оточення та хостинг.....	47
3.5 Тестування функціональності та забезпечення якості	49
3.6 Висновки до третього розділу	51
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	53
4.1 Долікарська медична допомога при захворюваннях, травмах та в умовах надзвичайних ситуацій.....	53

4.2	Вимоги безпеки до робочих місць для виконання робіт.....	55
4.3	Висновок до четвертого розділу.....	57
	ВИСНОВКИ	59
	ПЕРЕЛІК ДЖЕРЕЛ.....	61
	ДОДАТКИ	

ВСТУП

Актуальність теми. Сучасний розвиток інформаційних технологій та електронної комерції значно змінив спосіб роботи магазинів. Вебсайти стали невід'ємною частиною бізнес-діяльності багатьох, надаючи широкому колу споживачів швидкий доступ до товарів і послуг. В умовах глобалізації та посилення конкуренції особливого значення набуває наявність, ефективне функціонування та управління вебсайтом.

Вебсайти мають багато переваг як для бізнесу, так і для споживачів. З точки зору бізнесу, це дозволяє зменшити витрати на оренду фізичного простору, знизити операційні витрати та охопити більшу аудиторію без географічних обмежень. Споживачі мають можливість здійснювати покупки в будь-який час, порівнюючи ціни та читаючи відгуки інших користувачів. Розробка вебсайту включає не лише технічні аспекти, але й економічні, маркетингові та соціальні питання. Важливо, щоб вебсайт був забезпечений наступним:

- ергономічність користувацького інтерфейсу;
- безпеку обробки персональних даних і фінансових операцій;
- належні умови праці персоналу, який працює з комп'ютерною технікою.

З огляду на зростаюче значення цифрової економіки та онлайн-комерції, важливість розробки вебсайту “Моє дитя” не викликає сумнівів. Це не тільки вимога часу, але й необхідність забезпечення конкурентоспроможності магазину в сучасних ринкових умовах.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи є розробка вебсайту “Моє дитя” для забезпечення батьків зручним та ефективним інструментом придбання дитячих товарів, а також надання інформації про асортимент продукції, акції та новинки. Для досягнення цієї мети необхідно виконати завдання:

- провести аналіз вебсайтів для дитячого магазину, щоб визначити найкращі практики та функціональні вимоги;
- проаналізувати технології, що використовуються для розробки вебсайтів, в тому числі мови програмування, фреймворки;
- розробити дизайн вебсайту дитячого магазину “Моє дитя” включаючи макет та прототипування, щоб створити зручний та цікавий інтерфейс;
- обґрунтувати структуру сайту та його сторінок, визначити логічні зв'язки між розділами та категоріями дитячих товарів;
- розробити необхідні елементи управління сайтом, такі як кошики для покупок, системи реєстрації користувачів, пошукові системи та системи обробки замовлень;
- реалізувати програмування вебсайту, використовуючи обрані технології та інструменти;
- забезпечити безпеку обробки персональних даних та фінансових операцій відповідно до сучасних стандартів безпеки;
- протестувати розроблений вебсайт для виявлення та виправлення помилок і забезпечення стабільної роботи.

Реалізація цих завдань дозволить створити ефективний, зручний та безпечний сайт “Моє дитя”, який відповідатиме сучасним вимогам та потребам споживачів.

Практичне значення одержаних результатів. Розроблений вебсайт “Моє дитя” сприяє кращому доступу до товарів для батьків та осіб, які здійснюють догляд за дітьми, і він не тільки забезпечує зручність покупок, але й дозволяє ознайомитись користувачам з асортиментом продукції. Це також посилить конкурентоспроможність магазину “Моє дитя” в сучасному середовищі електронної комерції, що потенційно призведе до збільшення обсягу продажів і зростання доходів.

РОЗДІЛ 1. АНАЛІЗ ВИМОГ І ПОСТАНОВКА ЗАВДАНЬ ДЛЯ РОЗРОБКИ ВЕБСАЙТУ “МОЄ ДИТЯ”

1.1 Аналіз вебсайтів дитячих магазинів

В сучасному світі Інтернет став невід’ємною частиною нашого життя. Завдяки йому можна легко знайти всю необхідну інформацію, але на цьому його можливості не обмежуються: можна знайти і замовити будь-який товар, віртуально відвідати будь-яку країну, виконати перерахунок грошей та навіть знайти свою половинку [1]. Незважаючи на велику кількість ресурсів в Інтернеті, найбільш ефективними для користувачів, зацікавлених у конкретній інформації, є ті, що максимально відповідають їх потребам. Важливо, щоб ці ресурси створювали додатковий зв’язок між користувачем і власником, іншими користувачами та адміністратором. Такі ресурси часто групуються і розміщуються на тематичних веб-сайтах, які є легко доступними для користувачів.

Подібно до цього, батьки, які шукають дитячі товари, змушені витратити багато часу на пошуки найякісніших, найбезпечніших і найвигідніших покупок. Це стосується іграшок, одягу, продуктів харчування та інших товарів першої необхідності. Практично неможливо фізично порівняти десятки варіантів одночасно і врахувати всі їхні особливості. Існують великі магазини, де можна купити багато необхідні для дитини речі, але дослідження торгових залів зазвичай займає багато часу.

Вебсайти мають перевагу: досліджувати світ дитячих іграшок та інших товарів у віртуальному просторі набагато зручніше. Не потрібно мружитися, розглядаючи етикетки та наклейки з дрібним шрифтом, вибирати категорії товарів, вся необхідна інформація представлена на сайті в зручному і доступному вигляді. Каталог товарів дозволяє користувачам переглядати товари за допомогою фотографій, часто знаходити кілька зображень в одній

галереї товарів, надавати комплексний огляд товарів, користувачі можуть вибрати потрібний розділ, не витрачаючи час на не пов'язані між собою товари, що є поширеною проблемою у фізичних магазинах.

Іншою, не менш важливою перевагою вебсайтів дитячих магазинів є економія часу та зусиль. Якщо користувач хоче вибрати щось разом з дитиною, йому не потрібно турбуватися про те, що вона втомиться або шукати місце для годування, тому що все буде вдома, і можна легко дістатися до потрібного місця, навіть якщо дитина втомилася або процес займає занадто багато часу [2]. Отже, основні переваги здійснення покупок на вебсайті магазину:

- легкий доступ до широкого асортименту товарів, спрощують процес покупок і дозволяють батькам приймати обґрунтовані рішення, не витрачаючи зайвого часу. Це економить час при пошуку, виборі та порівнянні товарів;
- на відміну від фізичних магазинів з обмеженим простором на полицях, вебсайти дозволяють показати всі доступні товари;
- клієнти можуть зручно оформити замовлення та отримувати сповіщення, коли бажані товари знову з'являться на складі;
- на вебсайтах можна вибрати спосіб доставки, який найкраще відповідає потребам клієнта;
- детальні описи товарів та зображення з високою роздільною здатністю допомагають зробити усвідомлений вибір без необхідності фізичного огляду;
- товари класифіковані за віком, тому можна легко визначити придатність товарів для своїх дітей;
- часто пропонують прості варіанти повернення та обміну товарів, щоб клієнти були спокійні;
- підтримка та консультації легко доступні через різні онлайн-канали, які надають допомогу в зручному для користувача форматі.

Тому найбільшою перевагою дитячого вебсайту є економія часу, енергії та нервів. Дійсно, що може бути краще, ніж робити покупки з дому, не

турбуючись про те, що дитина зголодніє, захоче в туалет, занудьгує і буде вередувати або влаштує істеріку на людях, тому що ви хочете щось інше в магазині.

Звичайно, є деякі недоліки при покупці через вебсайти, але їх небагато, і якщо користувач неодноразово здійснював покупки в інтернеті, то він точно знає, як зробити правильний вибір, зробити покупку, і знизити ці ризики до нуля [3].

1.2 Формулювання вимог до функціональності та дизайну

Для того, щоб розробити вебсайт, необхідно чітко визначити функціональні вимоги, які забезпечать ефективне використання сайту та задовольнять потреби користувачів.

Щоб його створити, потрібно ретельно проаналізувати і виконати деякі важливі вимоги.

- По-перше, функціонал магазину повинен включати всі необхідні елементи для зручного та ефективного здійснення онлайн-операцій. Однією з головних вимог є реєстрація та аутентифікація користувачів, що дозволяє клієнтам створювати особистий кабінет. Пошук товарів повинен бути інтуїтивно зрозумілим і надавати можливості сортування та фільтрації за різними критеріями, такими як ціна та категорія. Рядок пошуку та функція розшифровки пропозицій товарів при введенні запитів також значно підвищує зручність використання.

- По-друге, кошик повинен бути простим у використанні і дозволяти додавати, редагувати і видаляти товари. Вміст кошика повинен зберігатися тільки під час активної сесії користувача, але можна використовувати вкладку «Вибране» для збереження товарів, які зацікавили користувача. Система замовлення і доставки повинна бути зручною для користувача і надавати можливість відстежувати статус замовлення в режимі реального часу з різними способами доставки і оплати.

Вимоги до дизайну вебсайту також є важливими. Вони впливають на створення позитивного враження та задоволення потреб клієнтів. Основними вимогами є:

- інтерфейс повинен мати привабливий, сучасний і професійний вигляд, який відображає бренд дитячого магазину і викликає довіру у покупців. Яскраві кольори та знайомі образи допоможуть привернути увагу батьків;
- важлива проста навігація: структура сайту заснована на логічному розташуванні товарних категорій і розділів, щоб можна було швидко знайти потрібний товар;
- швидкість завантаження сайту дуже важлива для підвищення задоволеності користувачів. Використання методів оптимізації зображень і коду, а також кешування може допомогти забезпечити швидке завантаження сторінок.

Підтримка користувачів також повинна бути швидкою і доступною через різні канали, такі як телефон та електронна пошта.

З технічної точки зору та з точки зору безпеки важливо використовувати сучасні методи шифрування для захисту даних користувачів від несанкціонованого доступу. Регулярне оновлення програмного забезпечення та моніторинг безпеки мають важливе значення для забезпечення високого рівня захисту [4].

Визначення цих вимог є важливим кроком у процесі розробки вебсайту для забезпечення ефективної функціональності та задоволення потреб користувачів.

1.3 Вибір технологій для вебсайту “Моє дитя”

При розробці вебсайту “Моє дитя”, спрямованого на створення зручного і ефективного сервісу для батьків, які шукають якісні товари для своїх дітей, також важливо вибрати оптимальну технологію, яка забезпечить не тільки потреби клієнтів, але і якість, швидкість і надійність їх роботи.

Перш за все, метою вибору технології є створення інтуїтивно зрозумілого, зручного для користувача і естетично привабливого інтерфейсу. Також необхідно забезпечити ефективний контроль над контентом і продуктами, безпеку і захист персональних даних. При виборі технології слід враховувати майбутній потенціал масштабування проекту і його адаптованість до мінливих потреб ринку [5].

У процесі розробки використовують різні технології і програмні засоби, включаючи PHP, JavaScript, MySQL, HTML та CSS.

Ознайомлення з кожною мовою.

- PHP – це мова програмування з відкритим кодом, спеціально розроблена для веб-розробки, яка може бути вбудована в HTML. На даний момент вона підтримується майже всіма хостинг-провайдерами і є однією з провідних мов для створення динамічних веб-сторінок. Статичні сторінки не завжди задовольняють потреби користувачів, тому PHP використовується для оновлення інформації на сайті в реальному часі, наприклад, для зміни прогнозу погоди або поточного курсу валют [6]. Використовуючи лише HTML, програміст не може вирішити такі завдання, тому він звертається до функціональності PHP. Ця мова дозволяє виконувати серверні скрипти на запит веб-браузера користувача. За допомогою PHP можна створювати професійні програми та вебсайти швидко і ефективно. PHP постійно оновлюється і залишається однією з найпопулярніших мов програмування завдяки своїй доступності та популярності, лідируючи в цьому списку [7].

- JavaScript – це інтерпретована мова програмування високого рівня. Спочатку використовуваний у браузерах, JS став одним з основних, поряд з мовою гіпертекстової розмітки HTML і мовою стилів CSS. JavaScript надає можливість додавати динамічну поведінку до вебсторінки, таке як обробка подій. Асинхронна взаємодія з сервером завантаження даних [8].

- HTML, також відомий в сфері інформаційних технологій як мова розмітки гіпертекстових документів, насправді являє собою спеціальний набір елементів-тегів. Вони допомагають структурувати кожен сторінку вебсайту [9].

- CSS – це мова, призначена для дизайну веб-сторінок. Основна функція CSS полягає у визначенні того, як буде виглядати вміст веб-сторінки, включаючи колір, шрифт, розмір, розташування елементів та інші аспекти візуального представлення. Основною перевагою використання CSS є зручність і простота використання. CSS дозволяє створювати складні стилі, використовуючи простий синтаксис, що полегшує розробку та підтримку вебсайту. Він також забезпечує сумісність між браузерами, тобто однаковий вигляд веб-сторінок у різних браузерах та на різних пристроях [10].

- MySQL – це система управління базами даних. У реляційній базі даних дані зберігаються в окремій таблиці, а не в одному місці, що робить її швидкою та гнучкою. Оскільки таблиці пов'язані між собою відносинами, можна об'єднувати дані з різних таблиць під час виконання запиту [11].

Для досягнення поставленої мети було проведено аналіз сучасного стану веб-ринку та ухвалено рішення розробляти систему, використовуючи мову програмування PHP та систему керування базами даних MySQL [12]. PHP використовувалася для створення серверних частин сховища та взаємодії з базами даних. JavaScript для створення динамічних елементів інтерфейсу, таких як анімація та взаємодія з користувачем. HTML та CSS для створення структури та дизайну веб-сторінок магазину. MySQL використовувалася для управління базами даних, зберігання та обробки даних, що дозволяє ефективно працювати з великими обсягами інформації та забезпечувати швидкий доступ до даних.

1.4 Вибір хостингу для вебсайту “Моє дитя”

Вибір надійного хостинг-провайдера є важливим етапом при створенні вебсайту, оскільки від цього залежить стабільність його роботи, швидкість завантаження сторінок, безпека даних клієнтів та можливість масштабування. Для вебсайту “Моє дитя” ретельно розглянуто кілька ключових критеріїв при виборі хостингу.

Хостинг – це послуга, що забезпечує розміщення вебсайту на серверах, постійно підключених до Інтернету. Це необхідний компонент для будь-якого сайту, адже він робить ваш сайт доступним для відвідувачів з будь-якого куточка світу, де є Інтернет [13]. Одним з основних критеріїв є надійність і безпека. Хостинг-провайдер повинен забезпечувати високу доступність сервісу, що включає мінімум 99,9% часу безвідмовної роботи. Крім того, важливо, щоб провайдер мав ефективні засоби захисту від кіберзагроз.

Не менш важливим є питання продуктивності. Швидкість завантаження сторінок відіграє ключову роль у задоволенні користувачів і впливає на пошукову оптимізацію. Тому обраний хостинг повинен забезпечувати достатню потужність серверів та можливість оптимізації швидкості роботи сайту.

Значну увагу слід приділити також масштабованості. Вебсайт може зростати, тому важливо, щоб хостинг-провайдер пропонував можливість легко збільшувати ресурси, такі як оперативна пам'ять, дисковий простір і процесорні потужності, відповідно до потреб бізнесу [14].

Враховуючи ці аспекти, було проведено детальний аналіз і порівняння кількох провідних хостинг-провайдерів для визначення оптимального варіанту для вебсайту “Моє дитя”. Для локального тестування і розробки було обрано ХАМРР. Це безкоштовний і простий у використанні локальний веб-сервер, який надає всі необхідні інструменти для розробки і тестування веб-додатків. Він включає такі компоненти, як Apache HTTP Server, MySQL, PHP і Perl, що дозволяє створювати повноцінне середовище для розробки.

ХАМРР забезпечує зручний спосіб розгортання веб-сервера на локальній машині. Він легко встановлюється та налаштовується, що дозволяє швидко розпочати роботу. Однією з ключових переваг ХАМРР є його портативність, що означає можливість перенесення налаштованого середовища з одного комп'ютера на інший без необхідності повторної інсталяції компонентів. Для розміщення вебсайту в мережі Інтернет був обраний хостинг Infinitifree. Цей хостинг-провайдер пропонує безкоштовні послуги з розміщення вебсайтів, що є вигідним рішенням на початкових етапах розвитку проєкту. Infinitifree

забезпечує базові ресурси для розміщення сайту, такі як необмежений дисковий простір і трафік, підтримку PHP та MySQL, а також простий у використанні інтерфейс для керування сайтом [15].

Для розміщення вебсайту в мережі Інтернет був обраний хостинг Infinitifree. Цей хостинг-провайдер пропонує безкоштовні послуги з розміщення вебсайтів, що є вигідним рішенням на початкових етапах розвитку проєкту. Infinitifree забезпечує базові ресурси, такі як необмежений дисковий простір і трафік, підтримку PHP та MySQL, а також простий у використанні інтерфейс для керування сайтом. Infinitifree надає кілька ключових переваг. Відсутність обмежень на дисковий простір і трафік дозволяє розміщувати велику кількість продуктів і медіа-контенту. Підтримка PHP та MySQL забезпечує можливість створення динамічних веб-сторінок і ефективне управління контентом. Простий у використанні інтерфейс дозволяє легко налаштовувати та керувати вебсайтом, навіть для користувачів без глибоких технічних знань [16].

Таким чином, вибір хостингу для вебсайту базувався на аналізі потреб проєкту, з урахуванням надійності, продуктивності, безпеки і масштабованості. Використання XAMPP для локальної розробки та Infinitifree для інтернет-хостингу дозволяє забезпечити ефективне і стабільне функціонування вебсайту на різних етапах його розвитку.

1.5 Огляд робочого простору обраного програмного забезпечення

Для успішної розробки було обрано кілька програмних засобів, кожен з яких виконує специфічні функції, необхідні для створення, тестування та розгортання веб-додатків. Правильний вибір програмного забезпечення значно впливає на ефективність робочого процесу, зручність роботи розробників та кінцеву якість продукту.

Основні програми, які було обрано для реалізації проєкту, включають Visual Studio, FileZilla та Figma. Кожне з цих програмних засобів відіграло важливу роль у створенні, тестуванні та розгортанні веб-додатку, і всі вони

мають свої унікальні можливості та переваги, які значно сприяли успішному втіленню проєкту.

Короткий опис кожної програми.

- Visual Studio – це інтегроване середовище розробки, що надає широкий набір інструментів для написання, редагування, налагодження та тестування коду. Воно підтримує різні мови програмування і фреймворки, що робить його ідеальним для створення як серверної логіки, так і фронтенд-розробки. Окрім цього, воно включає компілятор, інструменти для введення коду, керування версіями та розширення, що спрощує процес розробки та забезпечує зручність для розробників [17];

- FileZilla, популярний клієнт FTP, був обраний для передачі файлів між локальним комп'ютером та віддаленим сервером. Цей інструмент забезпечує надійне і швидке з'єднання, що полегшує процес оновлення та управління файлами на сервері. Завдяки FileZilla, ми змогли ефективно підтримувати і розвивати вебсайт дитячого магазину “Мое дитя”. Крім того, важливо зазначити, що дана програма також використовувалась як файловий менеджер для хостингу infinitifree, що спростило процес управління та маніпулювання файлами на веб-сервері [18];

- Figma – це онлайн-редактор, спеціально розроблений для дизайнерів з метою створення макетів, прототипів та дизайну користувацького інтерфейсу. Цей потужний інструмент дозволяє працювати над проєктами в реальному часі, спільно вносити зміни та ділитися своїми ідеями з колегами завдяки хмарному збереженню. Крім того, завдяки вбудованій функціональності спільної роботи, дизайнери можуть легко отримувати зворотній зв'язок від інших учасників проєкту та швидко внести необхідні зміни. Такий підхід до роботи сприяє швидкому та ефективному розвитку проєктів, забезпечуючи їхню якість та функціональність [19].

Visual Studio використовувалось для написання структури та коду вебсайту “Мое дитя”, забезпечуючи широкий набір інструментів для серверної логіки та фронтенд-розробки. Figma використовувалась для дизайну інтерфейсу

користувача та створення макетів та прототипів, дозволяючи спільно працювати над проектом у реальному часі. FileZilla служила для з'єднання з хостингом та передачі файлів між локальним комп'ютером та віддаленим сервером, спрощуючи процес оновлення та управління файлами на сервері. Використання цих програмних засобів сприяло успішному втіленню вебсайту, забезпечивши ефективний робочий процес та високу якість результуючого продукту.

1.6 Висновок до першого розділу

На початку кваліфікаційної роботи проведено аналіз вебсайтів дитячих магазинів, який показує, що вони мають значні переваги для батьків завдяки широкому асортименту, економії часу, зручності перегляду та можливостям доставки. Важливими аспектами є функціональні та дизайнерські вимоги до таких магазинів, що включають зручну реєстрацію, пошук, кошик, систему замовлення та доставки, а також привабливий і простий у навігації інтерфейс.

Для розробки вебсайту обрано технології PHP, JavaScript, HTML, CSS та MySQL, що забезпечують динамічність, зручність і безпеку. Вибір хостингу базувався на критеріях надійності, продуктивності та масштабованості, а для тестування та розробки використовувався XAMPP, тоді як для розміщення обрано Infinitifree. У процесі розробки застосовувалися Visual Studio для написання коду, FileZilla для передачі файлів та Figma для створення дизайну, що сприяло ефективному та якісному створенню вебсайту.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБСАЙТУ “МОЄ ДИТЯ”

2.1 Розробка архітектури системи

Архітектура системи вебсайту “Моє дитя” базується на вкладочній структурі, що забезпечує зручність навігації та доступність основних функцій для користувачів. Основними компонентами системи є головна сторінка, сторінка товарів, вкладка входу, вкладка обраного та кошик товарів. Було розглянуто деталі архітектури системи:

- головна сторінка вебсайту містить базову інформацію про сайт та його можливості. Вона слугує стартовою точкою для користувачів та забезпечує швидкий доступ до інших розділів сайту через навігаційну панель;
- сторінка товарів містить каталог продукції, яку можна придбати на вебсайті. Товари розділені на категорії для зручності пошуку та перегляду. Користувачі можуть переглядати деталі товарів, додавати їх до обраного чи кошика;
- вкладка входу надає можливість користувачам зареєструватись або авторизуватись на сайті. Реєстрація включає введення особистих даних та створення паролю, який шифрується для забезпечення безпеки;
- вкладка обраного зберігає товари, які користувач додав до списку обраного. Це дозволяє користувачам швидко знаходити та переглядати товари, які вони планують придбати в майбутньому;
- кошик товарів містить перелік товарів, які користувач планує придбати. Тут можна переглядати вміст кошика, змінювати кількість товарів та оформлювати замовлення. При виході з акаунту, вміст кошика очищується, що забезпечує безпеку даних;
- навігаційна панель забезпечує зручність переміщення між основними розділами сайту. Вона постійно доступна на всіх сторінках, дозволяючи користувачам швидко переходити від одного розділу до іншого без необхідності повертатись на головну сторінку.

Особлива увага приділяється безпеці користувачів, зокрема захисту їх особистих даних. Паролі користувачів шифруються, що значно ускладнює можливість їх злому. Щоб запобігти небезпеці злому, слід дотримуватися певних правил. Одним із найпростіших, але водночас важливих заходів є встановлення надійного пароля. Надійний пароль забезпечує першу лінію захисту від несанкціонованого доступу до пристрою та особистої інформації, що знижує ризик зловмисних атак.

Оскільки планується, що на сайті буде максимум близько 20-30 користувачів одночасно, вимоги до масштабованості системи не є суворими. Проте, архітектура системи передбачає можливість подальшого розширення та адаптації під збільшення кількості користувачів при необхідності.

Архітектура розроблена з урахуванням основних потреб користувачів, забезпечуючи зручність навігації, безпеку даних та простоту використання. Вкладочна структура дозволяє легко переміщуватись між розділами сайту, роблячи процес покупки інтуїтивно зрозумілим та приємним для користувачів.

На рисунку 2.1 представлено діаграму use case, на якій наведено основні сценарії використання вебсайту та взаємодію між користувачами і системою.

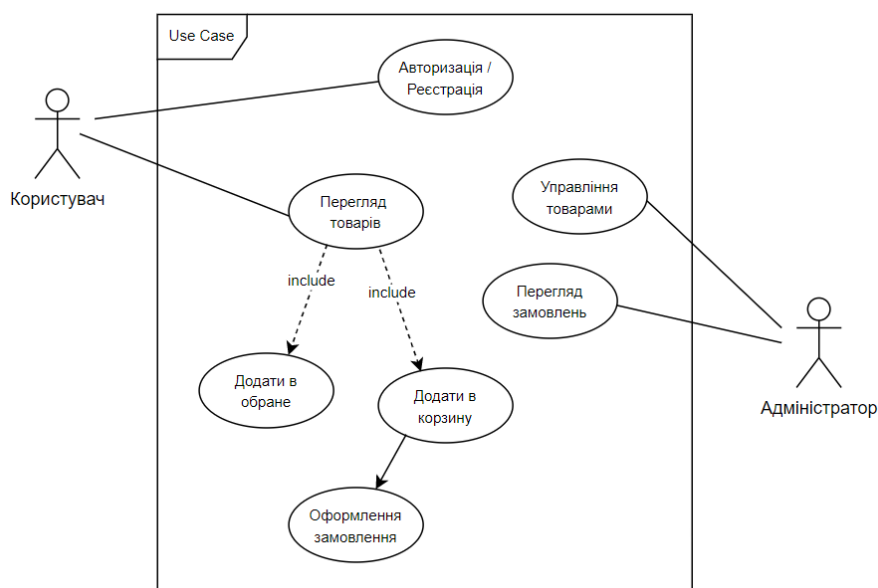


Рисунок 2.1 – Діаграма варіантів використання для вебсайту “Моє дитя”

Діаграма Use Case є важливим інструментом у моделюванні функціональних вимог системи. Вона показує взаємодію між зовнішніми користувачами та системою через різні сценарії використання. Кожен сценарій описує певну функцію або серію дій, яку система може виконувати у відповідь на запити акторів. Використання діаграми Use Case допомагає розробникам і замовникам краще зрозуміти, які саме функції має виконувати система, та забезпечує наочне представлення зв'язків між цими функціями [20].

Актори:

1. Користувач.
 - Це основний актор, який взаємодіє з вебсайтом для виконання різних операцій, таких як перегляд товарів, додавання товарів до обраного чи кошика, реєстрація/авторизація, та оформлення замовлень.
2. Адміністратор.
 - Це актор, який має доступ до функцій управління товарами та перегляду замовлень. Адміністратор відповідає за підтримку та адміністрування вебсайту.

Основні сценарії (Use Cases):

1. Авторизація/Реєстрація.
 - Користувач може авторизуватись або зареєструватись на сайті для доступу до персоналізованих функцій, таких як додавання товарів до обраного чи кошика.
2. Перегляд товарів.
 - Користувач може переглядати каталог товарів, доступних у магазині.
3. Додати в обране.
 - Користувач може додавати товари до списку обраного для швидкого доступу до них у майбутньому. Цей сценарій включається до сценарію перегляду товарів.
4. Додати в кошик.

- Користувач може додавати товари до кошика для подальшого оформлення замовлення. Цей сценарій також включається до сценарію перегляду товарів.

5. Оформлення замовлення.

- Користувач може оформити замовлення на товари, додані до кошика. Це наступний крок після додавання товарів до кошика.

6. Управління товарами.

- Адміністратор може додавати, редагувати або видаляти товари з каталогу.

7. Перегляд замовлень.

- Адміністратор може переглядати замовлення, зроблені користувачами.

Взаємозв'язки:

1. Include.

- Сценарії “Додати в обране” та “Додати в кошик” включаються в сценарій "Перегляд товарів", оскільки ці дії можливі тільки після перегляду товару.

Діаграма Use Case показує взаємодію між користувачами та системою. Вона чітко відображає основні функції, які підтримує система, та зв'язки між цими функціями. Це дозволяє зрозуміти, як різні частини системи взаємодіють між собою та з користувачами, забезпечуючи необхідний функціонал для роботи вебсайту.

2.2 Моделювання бази даних для вебсайту

Для вебсайту “Моє дитя” було створено базу даних із назвою if0_36202968_mychild. У цій базі даних створено чотири таблиці: Comments, Orders, Products та Users. В таблиці 2.1 наведено назви таблиць та стовпців, які мають відповідні таблиці.

База даних – це організована колекція даних, яка зберігається та отримується електронно. Бази даних використовуються для зберігання, управління та вилучення великої кількості інформації ефективним способом. Вони структуровані таким чином, щоб полегшити зберігання, вилучення, модифікацію та видалення даних у поєднанні з різними операціями обробки даних. Управління базами даних здійснюється системами управління базами даних, які надають користувачам та додаткам можливості створювати, запитувати, оновлювати та адмініструвати дані [21].

Таблиця 2.1 – Таблиці та стовпці БД вебсайту

№	Назва	Стовпці
1	Comments	id, initials, comment_date, comment
2	Orders	id, initials, products, phone_number, delivery_method, delivery_city, pickup_point, address, total_price, order_status, product_quantities
3	Products	id, product_name, price, trendy, image, global_group, characteristics, response, count_products, description
4	Users	id, phone_number, password, initials, favorites

Далі наведено детальний опис структури кожної таблиці, їхні зв'язки та ролі в системі.

Перша таблиця Comments, використовується для зберігання коментарів користувачів про товари або сервіс. Вона дозволяє зберігати ідентифікатор коментаря, ініціали користувача, який залишив коментар, дату створення коментаря та текст самого коментаря. Ця таблиця допомагає забезпечити

зворотний зв'язок від користувачів, що може бути корисним для покращення товарів та послуг.

Друга – це Orders. Вона містить інформацію про замовлення, зроблені користувачами. Вона зберігає унікальний ідентифікатор замовлення, ініціали користувача, перелік замовлених товарів, номер телефону користувача, метод доставки, місто доставки, пункт самовивозу, адресу доставки, загальну вартість замовлення, статус замовлення та кількість товарів. Ця таблиця є ключовою для обробки та відстеження замовлень у системі.

Третя таблиця Products, зберігає інформацію про товари, доступні в магазині. Вона містить унікальний ідентифікатор товару, назву товару, ціну, показник популярності, зображення товару, групу товару (наприклад, харчування, іграшки, одяг, коляски), характеристики товару, відгуки про товар, кількість товару на складі та опис товару. Ця таблиця дозволяє системі управляти асортиментом товарів та надавати користувачам необхідну інформацію про товари.

І остання таблиця Users, зберігає інформацію про користувачів магазину, включаючи унікальний ідентифікатор користувача, номер телефону, пароль, ініціали та список улюблених товарів. Вона є основою для управління користувачами, забезпечення безпеки та персоналізації користувацького досвіду.

Зв'язки між таблицями забезпечують цілісність даних та їх узгодженість:

- таблиця Orders має зв'язок з таблицею Users через стовпець initials.

Це дозволяє визначити, хто зробив конкретне замовлення;

- таблиця Orders також має зв'язок з таблицею Products через стовпець products, де зберігаються дані про замовлені товари;

- таблиця Users має зв'язок з таблицею Products через стовпець favorites, де зберігаються ID улюблених товарів.

На рисунку 2.1 представлена діаграма бази даних, яка відображає взаємодію між таблицями Users, Comments, Orders та Products.

Діаграма показує логічні зв'язки між таблицями через відповідні стовпці, що допомагає візуально розуміти взаємодію між різними частинами системи. Зв'язки між таблицями Users та Orders через стовпець initials і між таблицями Orders та Products через стовпець products виглядають логічно і коректно. Таблиця Comments правильно пов'язана з таблицею Users через стовпець initials.

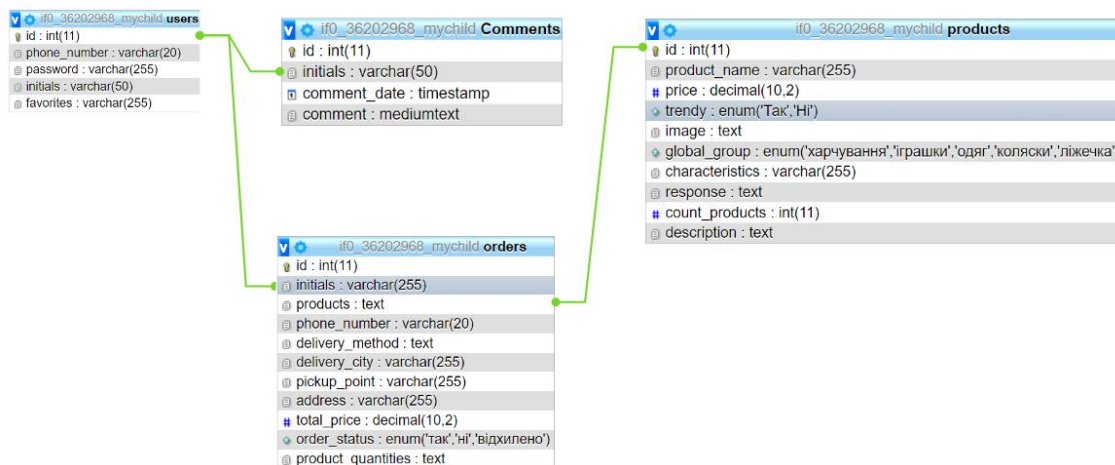


Рисунок 2.2 – Діаграма взаємодії таблиць бази даних вебсайту “Мое дитя”

Моделювання бази даних для вебсайту “Мое дитя” включає створення чотирьох основних таблиць. Ці таблиці пов'язані між собою для забезпечення цілісності даних та підтримки основних функцій магазину, таких як управління товарами, замовленнями та користувачами.

Така структура бази даних забезпечує ефективне зберігання та обробку даних, необхідних для функціонування вебсайту, забезпечуючи зручність використання та інтеграцію різних компонентів системи.

2.3 Опис інтерфейсу дитячого-магазину “Мое дитя”

Інтерфейс розроблений з урахуванням зручності користування та естетичної привабливості. Він складається з кількох основних сторінок, кожна з яких має специфічні елементи та функції.

Головна сторінка є вітриною вебсайту. Вона містить велике рекламне вікно, яке змінюється з певним періодом, демонструючи різні рекламні пропозиції. Нижче розташовані прокручуваний блок з випадковими товарами, що дозволяє користувачам швидко ознайомитися з доступними товарами. Під цим блоком знаходяться категорії товарів, включаючи “Одяг”, “Харчування”, “Коляски” та “Іграшки”. Прокрутивши далі, користувачі побачать додатковий прокручуваний блок з товарами, а ще нижче – відгуки клієнтів. Внизу сторінки розміщений футер, який містить контактну інформацію, тоді як зверху знаходиться навігаційна панель для зручної навігації по сайту.

Сторінка товарів продовжує зручну навігацію, маючи зверху навігаційну панель і знизу футер. Ліва частина сторінки зайнята фільтрами та категоріями товарів, що знаходяться в окремих рамках. Це дозволяє користувачам швидко знаходити потрібні товари за певними параметрами. Праворуч розміщені самі товари з інформацією про них. Знизу сторінки розташований блок пагінації, який дозволяє переміщатися між сторінками з товарами.

Вкладка входу забезпечує простий та інтуїтивний процес авторизації. В центрі сторінки розташований острівцець з полями для введення логіну та паролю. Користувачі можуть швидко зареєструватися або авторизуватися для доступу до персоналізованих функцій магазину.

Сторінка обраного відображає всі обрані товари користувача. Якщо є обрані товари, вони показуються зі значком обраного червоного кольору, що дозволяє легко ідентифікувати вибрані позиції. Користувачі можуть переглядати та видаляти товари зі списку обраного за потреби.

Кошик товарів представлений у вигляді вертикального списку, де кожен товар займає всю ширину контейнера. Це забезпечує зручність перегляду та управління замовленнями. Коли користувач готовий оформити замовлення, відкривається додатковий острівцець для введення персональних даних, таких як адреса доставки та контактна інформація.

Навігаційна панель є ключовим елементом інтерфейсу, що дозволяє користувачам легко переміщатися між різними розділами сайту. Ліворуч

розташований логотип, в центрі – посилання на головну сторінку та сторінку товарів. Справа знаходяться посилання на сторінку входу, сторінку обраного та кошик.

Цей інтерфейс забезпечує зручну та інтуїтивно зрозумілу навігацію по сайту, дозволяючи користувачам легко знаходити та купувати необхідні товари.

2.4 Розробка файлової структури проекту

Файлова структура проекту вебсайту “Мое дитя” організована таким чином, щоб забезпечити логічний розподіл файлів за їх призначенням та полегшити підтримку і розвиток проекту. Основними папками є: `css`, `js`, `php`, `img` та `font`. Кожна директорія містить файли, які відповідають за певний аспект функціонування сайту, що дозволяє легко орієнтуватися в коді та робити необхідні зміни.

1. Папка `css` містить файли з розширенням `.css`, які відповідають за стилізацію всіх сторінок сайту. Всі стилі, які визначають зовнішній вигляд елементів сайту, такі як кольори, шрифти, відступи, межі, розташування елементів тощо, зберігаються в цій директорії.

2. Папка `js` містить файли з розширенням `.js`, які реалізують різні функції для взаємодії користувача із сайтом. JavaScript використовується для додавання динамічної поведінки на сторінках, таких як реагування на натискання кнопок, валідація форм, анімації та інші інтерактивні елементи.

3. Папка `php` містить файли з розширенням `.php`, які виконують серверну логіку сайту. PHP-скрипти обробляють запити користувачів, взаємодіють з базою даних, генерують HTML-код для відображення на клієнтській стороні та виконують інші серверні завдання.

4. Папка `img` містить файли зображень з розширенням `.png` та іншими графічними форматами. Ця директорія використовується для зберігання зображень, які є частиною контенту сайту, таких як декоративні елементи, фотографії товарів, логотипи тощо.

5. Папка font містить файли шрифтів з розширенням .woff2, які використовуються для відображення спеціальних шрифтів на сайті. Використання власних шрифтів дозволяє зробити дизайн сайту унікальним та відповідним до стилю бренду.

Файли стилів, які визначають зовнішній вигляд сайту, включають оформлення тексту, розміщення елементів, кольори, розміри та інші стилі. JavaScript-файли додають інтерактивність на сторінки сайту і можуть містити функції для обробки подій, анімації, валідації форм та інших динамічних елементів. PHP-файли виконують серверну логіку, включаючи обробку форм, роботу з базою даних, генерування HTML-коду та інші серверні задачі. Зображення використовуються для графічного оформлення сайту, включаючи декоративні елементи, фотографії товарів та інші візуальні компоненти, а файли шрифтів дозволяють зробити дизайн унікальним та відповідним до стилю бренду.

2.5 Дизайн інтерфейсу та вибір кольорової схеми

Дизайн інтерфейсу – це процес створення інтерфейсу користувача для програмного забезпечення або вебсайту, з акцентом на зовнішній вигляд і взаємодію з користувачем. Основною метою дизайну інтерфейсу є забезпечення ефективної, зручної та привабливої взаємодії між користувачем та системою [22].

Дизайн інтерфейсу розроблений з урахуванням зручності користувачів та привабливості для цільової аудиторії – батьків та дітей. Основні елементи дизайну, які можна побачити на головній сторінці, включають зручну навігаційну панель, яскраві банери, категорії товарів та відгуки клієнтів. На рисунку 2.3 видно, що навігаційна панель розташована зверху сторінки і містить логотип сайту, посилання на головну сторінку та сторінку товарів, а також значки для входу, обраного та кошика.

Логотип і значки мають привабливий дизайн, що добре узгоджується з загальною стилістикою сайту. Банер, що змінюється, займає центральне місце і привертає увагу користувачів до актуальних пропозицій і новинок, що дає можливість обрати одні з найпопулярніших товарів.

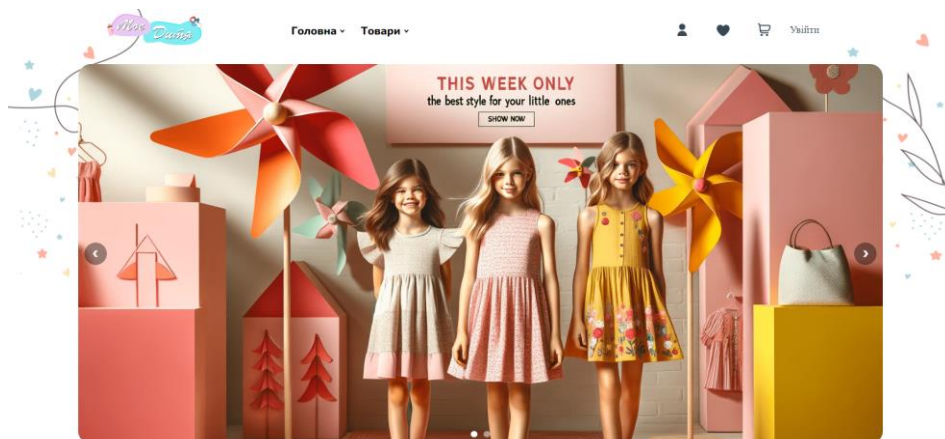


Рисунок 2.3 – Головна сторінка вебсайту з банером

Нижче банеру розташовані категорії товарів, представлені у вигляді круглих зображень з підписами, що полегшує користувачам пошук необхідних товарів. Категорії включають “Іграшки”, “Коляски”, “Одяг” та “Харчування”. Кожна категорія має своє унікальне зображення, що дозволяє швидко зорієнтуватися на сторінці. Далі розміщено блок з новинками, де користувачі можуть переглянути трендові товари.

Під блоком новинок розташовані відгуки клієнтів, що створюють додаткову довіру до магазину (див. рисунок 2.4).

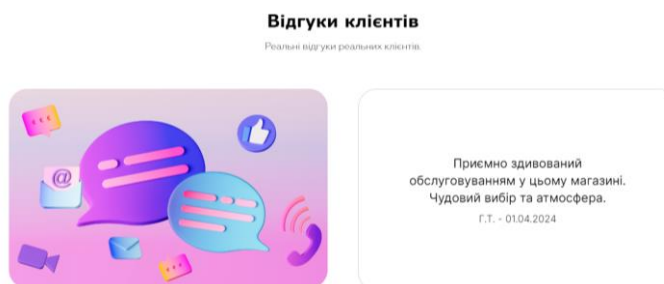


Рисунок 2.4 – Відгуки клієнтів на головній сторінці

Відгуки оформлені в сучасному стилі з використанням графічних елементів, що робить їх привабливими для читання.

Футер містить контактну інформацію, а також додаткові посилання на інформаційні сторінки, обробку замовлень, новини та соціальні мережі. Він включає номери телефонів, адресу електронної пошти для підтримки клієнтів, посилання на політику конфіденційності та умови використання сайту. Використання кольорової схеми в футері робить його легко помітним, але не відволікаючим від основного контенту сторінки, забезпечуючи гармонійний вигляд всього інтерфейсу.

Далі розглянемо дизайн пошуку товару. На рисунку 2.5 видно, що сторінка пошуку товару містить зручний інтерфейс для вибору товарів. З лівої сторони розміщені глобальні категорії товарів, такі як "Одяг", "Іграшки", "Харчування", "Коляски", "Ліжечка". Нижче знаходяться фільтри, що дозволяють користувачам швидко знайти необхідний товар. З правої сторони розміщені товари, які відповідають заданим критеріям, з детальною інформацією про кожен товар та кнопками додати до кошика та обраного.

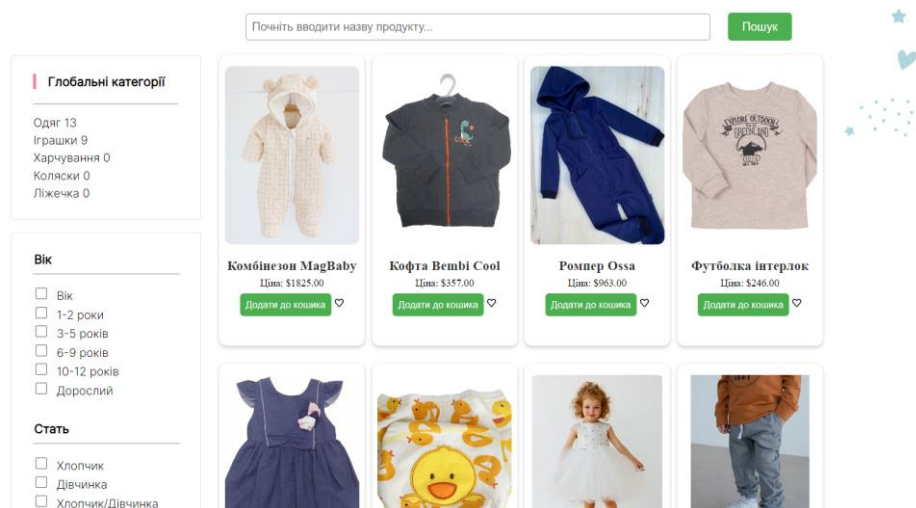


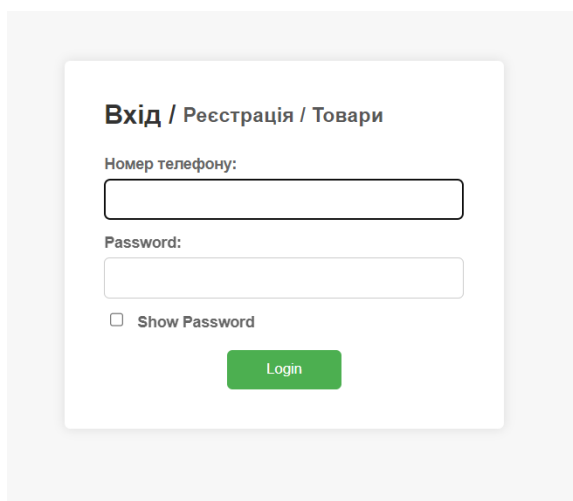
Рисунок 2.5 – Сторінка пошуку товару з фільтрами та результатами пошуку

Користувач може скористатися пошуковим рядком у верхній частині сторінки для швидкого пошуку товару за назвою. Такий дизайн дозволяє

користувачам легко знайти потрібні товари та додати їх до кошика, що робить процес покупки швидким та зручним.

Вікно авторизації має простий та інтуїтивно зрозумілий дизайн. У верхній частині розташований заголовок з посиланнями на сторінки реєстрації та товарів, що полегшує навігацію (див. рисунок 2.6).

Основна частина інтерфейсу включає два поля для введення: номер телефону та пароль. Поруч з полем пароля розташований чекбокс "Show Password", який дозволяє відобразити введений пароль для перевірки його правильності. Під полями введення розміщена зелена кнопка "Login", яка привертає увагу і сигналізує про дію, яку потрібно виконати для входу на сайт. Такий мінімалістичний підхід забезпечує зручність і простоту використання, покращуючи користувацький досвід.



The image shows a login form titled "Вхід / Реєстрація / Товари". It contains two input fields: "Номер телефону:" and "Password:". Below the password field is a checkbox labeled "Show Password". At the bottom of the form is a green button labeled "Login".

Рисунок 2.6 – Авторизація на вебсайті “Мое дитя”

Інтерфейс кошика на вебсайті є зрозумілим і функціональним (див. рисунок 2.7). У верхній частині сторінки розташований заголовок "Кошик". Під заголовком знаходиться список товарів з їх зображеннями, назвами, цінами, кількістю та загальною сумою. Користувач може змінювати кількість товару, і загальна сума автоматично оновлюється. Праворуч від товарів розміщені кнопки для видалення обраних товарів і додавання деталей замовлення.

Нижче знаходиться форма для введення додаткових полів, таких як ПІБ, номер телефону та місце доставки. Користувач може вибрати один з трьох способів доставки: у відділення, у поштомат або кур'єром, за допомогою радіокнопок. В кінці форми розташована кнопка "Підтвердити замовлення", яка завершує процес оформлення замовлення.

Кольорова схема вебсайту вибрана таким чином, щоб створити дружелюбну та привабливу атмосферу для користувачів, яка особливо підходить для цільової аудиторії – батьків та дітей. Основні кольори включають м'які відтінки синього, рожевого та зеленого, що створюють гармонійний та заспокійливий вигляд:

- світло-блакитний: #ADD8E6;
- рожевий: #FFC0CB;
- зелений: #32CD32.

Кошик

Виділити всі

Музичний мобіль
2899.00грн 1 шт.

Загальна сума: 2899.00 грн

Видалити обрані товари Додати деталі

Додаткові поля

ПІБ:
Іван Іванович Івано

Номер телефону:
+380123456789

Доставка:
Місце доставки:
Київська область, місто Київ

У відділення У поштомат Кур'єром

Підтвердити замовлення

Рисунок 2.7 – Товар в кошику та додаткові поля для оформлення замовлення

Ці кольори використовуються для фону та акцентних елементів, що допомагає створити відчуття тепла та турботи. Зелені акценти застосовуються для кнопок і важливих елементів, таких як "Додати до кошика" та "Підтвердити

замовлення", що робить їх легко помітними та привертає увагу користувачів до ключових дій.

2.6 Взаємодія з користувачем та забезпечення зручності навігації

Взаємодія з користувачем на сайті орієнтована на забезпечення максимальної зручності та інтуїтивної зрозумілості. Користувачі можуть виконувати ряд основних дій, таких як додавання товарів до кошика та в обране, видалення їх, оформлення замовлень, авторизація або реєстрація, перегляд окремих товарів, а також використання фільтрів та пошуку для швидкого знаходження потрібних продуктів. Організація навігації на сайті дозволяє користувачам легко знаходити потрібні розділи та здійснювати покупки [23].

Швидкість завантаження сторінок забезпечується зберіганням усіх зображень на хмарі Dropbox. Dropbox – це популярний сервіс для зберігання і синхронізації файлів у хмарі, який дозволяє зберігати файли на віддалених серверах і отримувати до них доступ з будь-якого пристрою, підключеного до Інтернету. Використання Dropbox дозволяє витягувати зображення не з бази даних, а з хмари, що значно знижує навантаження на сервер і покращує продуктивність сайту. Оптимізація завантаження зображень гарантує швидкий доступ до контенту, що є важливим для підтримки позитивного користувацького досвіду [24].

Зворотній зв'язок з користувачами також організований просто і зручно. У футері сайту розміщена контактна інформація, включаючи номер телефону та електронну пошту, за якими користувачі можуть звертатися за допомогою або залишати свої відгуки. Це забезпечує можливість швидко отримати підтримку та вирішити будь-які питання, що виникають у користувачів під час користування сайтом.

Можливість персоналізації надається через збереження обраних товарів. Користувачі можуть додавати товари до списку обраного, що дозволяє швидко знаходити і переглядати їх у майбутньому.

Таким чином, взаємодія з користувачем на сайті “Моє дитя” забезпечується через зручну навігацію, швидке завантаження сторінок, легкий доступ до контактної інформації та можливість збереження обраних товарів. Усі ці елементи сприяють створенню позитивного і приємного досвіду для користувачів.

2.7 Висновок до другого розділу

У другому розділі кваліфікаційної роботи було детально розглянуто проектування вебсайту “Моє дитя”. Було розроблено архітектуру системи, яка базується на вкладочній структурі, що забезпечує зручність навігації та доступність основних функцій для користувачів. Моделювання бази даних включало створення таблиць для зберігання ключових даних про товари, замовлення та користувачів, що дозволяє ефективно управляти інформацією. Опис інтерфейсу включав аналіз основних сторінок сайту, таких як головна сторінка, сторінка товарів, вкладка входу, обране та кошик, з акцентом на зручність та інтуїтивність користування.

Було визначено файлову структуру проекту, що полегшує підтримку та розвиток сайту. Дизайн інтерфейсу та вибір кольорової схеми спрямовані на створення дружлюбної та привабливої атмосфери для користувачів. Взаємодія з користувачем забезпечується через зручну навігацію, швидке завантаження сторінок завдяки використанню хмарного сервісу Dropbox та організацію зворотного зв'язку через контактну інформацію у футері сайту. Усі ці аспекти сприяють створенню позитивного користувацького досвіду та підвищенню ефективності вебсайту “Моє дитя”.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ВЕБСАЙТУ “МОЄ ДИТЯ”

3.1 Розробка серверної частини на PHP

Бекенд – це серверна частина сайту або програми, яка відповідає за всі процеси, що відбуваються за лаштунками і не видно користувачеві. Назва “Back-end” вказує на те, що це прихована сторона програмного забезпечення, яка забезпечує його функціональність [25].

Серверна частина магазину “Моє дитя” реалізована на мові програмування PHP. Вона відіграє ключову роль у забезпеченні функціонування всього вебсайту, відповідаючи за обробку запитів від клієнтської частини, взаємодію з базою даних, забезпечення безпеки та виконання інших важливих завдань. Для ілюстрації принципу роботи серверної частини було розглянуто взаємодію користувача із сторінкою пошуку товарів та продемонстровано, як реалізовано фільтрацію товарів за категоріями, цінами та іншими параметрами.

Код починається з визначення активної категорії товарів, що зберігається у змінній “\$group” (див. лістинг 3.1).

Лістинг 3.1 – Визначення активної категорії та підключення до бази

```
$group = isset($_GET['menu-link']) ? $_GET['menu-link'] :
(isset($_SESSION['group']) ? $_SESSION['group'] : 'іграшки');
$active_checkbox_groups = $checkbox_groups_by_category[$group];
$_SESSION['group'] = $group;

$servername = "sql311.infinityfree.com";
$username = "if0_36202968";
$password = "Qwerty193758246";
$dbname = "if0_36202968_mychild";

$conn = new mysqli($servername, $username, $password, $dbname);
mysqli_set_charset($conn, "utf8");

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
```

Якщо користувач вибрав категорію через GET-запит, ця категорія буде встановлена як активна. Якщо категорія не вибрана, використовується значення із сесії, або за замовчуванням встановлюється категорія "іграшки". Потім обрана категорія зберігається в сесії для подальшого використання. Після цього встановлюються потрібні чекбокси для фільтрації товарів у обраній категорії. Далі виконується підключення до бази даних MySQL.

На цьому етапі отримується пошуковий запит користувача із GET-параметрів і ініціалізуються змінні для фільтрації за ціною (див. лістинг 3.2). Далі виконується SQL-запит для отримання товарів з таблиці `products` на основі пошукового запиту і обраної категорії. Результати запиту використовуються для підрахунку загальної кількості товарів і кількості сторінок для пагінації. Потім отримуються товари для поточної сторінки з урахуванням пагінації.

Лістинг 3.2 – Ініціалізація змінних для фільтрації за ціною та SQL-запит для отримання товарів

```
$search = $_GET['search'] ?? '';
$min_price = isset($_GET['min_price']) ? (float)$_GET['min_price']
: null;
$max_price = isset($_GET['max_price']) ? (float)$_GET['max_price']
: null;

$query = "SELECT product_name, price, image FROM products WHERE
(product_name LIKE '%$search%' OR global_group = '$group') AND
(global_group = '$group') AND count_products > 0";
$result = mysqli_query($conn, $query);

$total_products = mysqli_num_rows($result);
$per_page = 28;
$num_pages = ceil($total_products / $per_page);

$current_page = isset($_GET['page']) ? (int)$_GET['page'] : 1;

$offset = ($current_page - 1) * $per_page;

$query = "SELECT product_name, price, image FROM products WHERE
(product_name LIKE '%$search%' OR global_group = '$group') AND
(global_group = '$group') AND count_products > 0 LIMIT $offset,
$per_page";
$result = mysqli_query($conn, $query);
```

Наступний фрагмент коду формує умови фільтрації для чекбоксів (див. лістинг 3.3). Для кожної активної групи чекбоксів перевіряється, чи обрані користувачем певні характеристики. Якщо так, вони додаються до умов фільтрації. Умови об'єднуються за допомогою логічних операторів "OR" та "AND". Потім виконується SQL-запит для вибірки товарів з таблиці `products`, що відповідають пошуковому запиту, обраній категорії та умовам фільтрації чекбоксів.

Лістинг 3.3 – Формування умов фільтрації для чекбоксів та SQL-запит

```

$where_conditions = array();
foreach ($active_checkbox_groups as $category => $checkboxes) {
    $group_conditions = array();
    foreach ($checkboxes as $checkbox_id => $checkbox_label) {
        if (isset($_GET[$checkbox_id])) {
            $group_conditions[] = "characteristics LIKE '%" .
                $checkbox_label . "%'";
        }
    }
    if (!empty($group_conditions)) {
        $where_conditions[] = "(" . implode(" OR ",
            $group_conditions) . ")";
    }
}

$checkbox_condition = '';
if (!empty($where_conditions)) {
    $checkbox_condition = " AND (" . implode(" AND ",
        $where_conditions) . ")";
}

$query = "SELECT product_name, price, image FROM products WHERE
(product_name LIKE '%$search%' OR global_group = '$group')
$checkbox_condition AND (global_group = '$group') AND
count_products > 0";
$result = mysqli_query($conn, $query);

$filtered_total_products = mysqli_num_rows($result);
$num_pages = ($filtered_total_products <= $per_page) ? 1 :
ceil($filtered_total_products / $per_page);
$offset = ($current_page - 1) * $per_page;

```

В лістингу 3.4 фрагменті коду визначається умова фільтрації за ціною на основі мінімальної та максимальної ціни, які були введені користувачем. Потім

виконується SQL-запит для вибірки товарів з урахуванням всіх фільтрів: пошукового запиту, обраної категорії, фільтрації чекбоксів та цінкових обмежень.

Лістинг 3.4 – Визначення умов фільтрації за ціною та SQL-запит

```
$price_condition = '';
if ($min_price !== null || $max_price !== null) {
    $price_condition = "AND price BETWEEN ";
    if ($min_price !== null) {
        $price_condition .= $min_price;
    } else {
        $price_condition .= "0";
    }
    $price_condition .= " AND ";
    if ($max_price !== null) {$price_condition .= $max_price;}
else {$price_condition .= "1000";}}

$query = "SELECT id ,product_name, price, image FROM products
WHERE (product_name LIKE '%$search%' OR global_group = '$group')
$checkbox_condition AND (global_group = '$group') $price_condition
AND count_products > 0";
$result = mysqli_query($conn, $query);
```

Наступний фрагмент коду визначає найбільшу групу товарів на основі результатів пошуку і фільтрації (див. лістинг 3.5). Виконується SQL-запит для підрахунку кількості товарів у кожній групі, сортування за кількістю у спадному порядку та вибірка найбільшої групи. Якщо результат не порожній, зберігається інформація про найбільшу групу в сесії для подальшого використання. Якщо результат порожній, використовується попередня знайдена група.

Лістинг 3.5 – Визначення найбільшої групи товарів та збереження в сесії

```
$group_condition = ($group != '') ? "AND global_group = '$group'"
: "";
$group_query = "SELECT global_group, COUNT(*) as count FROM
products WHERE (product_name LIKE '%$search%' OR global_group =
'$group') $checkbox_condition $group_condition AND count_products
> 0 GROUP BY global_group ORDER BY count DESC LIMIT 1";
$group_result = mysqli_query($conn, $group_query);
$largest_group_info = mysqli_fetch_assoc($group_result);
```



```

if ($largest_group_info !== null) {} else {
    $largest_group_info['global_group'] =
isset($_SESSION['last_group']) ? $_SESSION['last_group'] : $group;
}
$_SESSION['last_group'] = $largest_group_info['global_group'];

```

Залежно від найбільшої групи товарів, виконується динамічне генерування CSS-стилів для відображення відповідних елементів на сторінці. Це дозволяє показувати або приховувати певні категорії товарів залежно від результатів пошуку. У кінці коду отримуються товари, які відповідають фільтрам, і зберігаються у масиві `\$info` для подальшого відображення користувачу.

Лістинг 3.6 – Динамічне генерування CSS-стилів та збереження товарів

```

$info = [];

if (mysqli_num_rows($result) > 0) {
    while ($row = mysqli_fetch_assoc($result)) {
        $info[] = $row;
    }
}

```

Таким чином, серверна частина магазину надає користувачам можливість фільтрувати товари за різними параметрами, такими як категорії, ціни та характеристики, забезпечуючи зручний і ефективний пошук товарів в інтернет-магазині. Більш детально код наведено в додатку А.

Окрім фільтрації товарів, серверна частина включає реалізацію інших важливих функцій, таких як додавання товарів до кошика, обраного, видалення товарів, а також оформлення замовлень. Наприклад, додавання товару до обраного виконується через PHP-скрипт, який взаємодіє з базою даних, щоб зберегти інформацію про вибрані користувачем товари. Це дозволяє користувачам зберігати свої уподобання та зручно повертатися до них у майбутньому.

Скрипти, які обробляють додавання товарів до обраного або до кошика, виконують перевірку авторизації користувача, взаємодію з базою даних для

зберігання або видалення відповідної інформації, а також надають користувачеві зворотний зв'язок про успішність або помилки виконаної операції. Такі скрипти також відповідають за обробку даних сесії користувача, зберігаючи інформацію про обрані товари між сеансами. Це забезпечує безперебійний та комфортний процес покупок для користувачів, надаючи їм можливість легко керувати своїм кошиком та списком обраного, а також оформлювати замовлення.

3.2 Створення клієнтської частини за допомогою JavaScript, HTML та CSS

Клієнтська частина, або Frontend розробка, включає CSS верстку, створення макетів і шаблонів для сайту, а також розробку користувацького інтерфейсу і спеціальних скриптів, що відповідають за візуалізацію та анімацію [26].

Більшість сторінок сайту мають загальні елементи, такі як footer, header та навігаційну панель (nav). Основний контент сторінки, зокрема товари, розміщували у секції з класом list-goods. Це дозволяє забезпечити єдиний стиль і зручну навігацію по сайту.

HTML-файли були структуровані таким чином, що кожна сторінка має свою унікальну розмітку, але зберігає спільні елементи, такі як header та footer. Наприклад, на головній сторінці містяться елементи для відображення банерів та категорій товарів, тоді як сторінка товарів зосереджена на відображенні списку товарів з можливістю фільтрації та сортування.

CSS-файли, що використовувалися для стилізації сторінок, включають products.css, index.css та log-reg.css. Ці стилі охоплюють всі сторінки сайту, забезпечуючи узгоджений дизайн і відповідність бренду. У products.css зберігали стилі для сторінки товарів, включаючи відображення карток товарів та фільтрів. index.css містить стилі для головної сторінки, такі як банери та категорії товарів. log-reg.css використовується для стилізації сторінок

авторизації та реєстрації. Переважно використовували flex та його допоміжні властивості для розміщення елементів.

JavaScript забезпечував інтерактивність та обробку подій на сторінках. Основні функції включають оновлення значень текстових полів введення ціни, обробку кнопок, показ попапів і затемнення фону, перевірку наявності товару в обраному та перевірку заповнення полів при оформленні замовлення. Для прикладу було розглянуто код, що реалізував додавання товару до обраного (див. лістинг 3.7).

Лістинг 3.7 – Додавання товару до обраного

```
function addToFavorites(productId) {
    var xhr = new XMLHttpRequest();
    xhr.open("POST", "../php/add_to_favorites.php", true);
    xhr.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");
    xhr.onreadystatechange = function () {
        if (xhr.readyState === 4 && xhr.status === 200) {
            location.reload();
        }
    };
    xhr.send("productId=" + productId);
}
```

Ця функція виконує запит AJAX для додавання товару до обраних. Вона відправляє POST-запит до сервера, де обробляється відповідний PHP-скрипт. Якщо запит був успішний, сторінка оновлювалюється, щоб відобразити зміни.

Крім цієї функції, JavaScript також забезпечує інші важливі функції, такі як обробка кошика. Наприклад, для додавання товару до кошика використовується код, код який надається в лістингу 3.8.

Лістинг 3.8 – Обробка для додання товару до кошику

```
function addToCart(productId) {
    var xhr = new XMLHttpRequest();
    xhr.open("POST", "../php/add_to_cart.php", true);
    xhr.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded");
    xhr.onreadystatechange = function () {
        if (xhr.readyState === 4 && xhr.status === 200) {
```

```

        alert("Товар успішно додано до кошика");
    }
};
xhr.send("productId=" + productId);
}

```

Клієнтська частина також використовує бібліотеки jQuery та Ajax для спрощення роботи з DOM і асинхронними запитами. Це дозволяє значно спростити написання коду та забезпечити більш ефективну обробку подій.

Для обробки замовлення використовується функція JS для валідації форми (див. лістинг 3.2). Ця функція перевіряє, чи всі необхідні поля були заповнені, і чи обраний метод доставки. Усі функції даного файлу наведено в додатку Б.

Лістинг 3.9 – Валідація форми замовлення

```

function validateForm() {
    var                initials                =
document.getElementById("initials").value.trim();
    var                phoneNumber            =
document.getElementById("phone_number").value.trim();
    var                deliveryCity          =
document.getElementById("delivery_city").value.trim();

    var                deliveryMethod        =
document.querySelector('input[name="delivery_method"]:checked');

    if (!deliveryMethod) {
        alert("Будь ласка, оберіть метод доставки.");
        return false;
    }

    if (initials === "" || phoneNumber === "" || deliveryCity ===
    "") {
        alert("Будь ласка, заповніть всі поля.");
        return false;
    }
    return true;
}

```

jQuery — це швидка, маленька і багатофункціональна JavaScript бібліотека, яка спрощує написання сценаріїв HTML на стороні клієнта. Однією з ключових особливостей jQuery є підтримка Ajax, що дозволяє створювати

асинхронні веб-застосунки [27]. Ајах дає змогу оновлювати частини веб-сторінки без перезавантаження всієї сторінки, покращуючи користувацький досвід. Завдяки jQuery розробники можуть легко використовувати Ајах для динамічної взаємодії з сервером, що робить веб-застосунки більш інтерактивними та швидкими [28].

Ајах та jQuery дозволяють не перезавантажувати сторінку без потреби, забезпечуючи взаємодію з користувачем при натисканні, наприклад, на кнопку додати в кошик. Після натискання на кнопку сторінка не перезавантажується, але всі дані посилаються на сервер для обробки.

HTML і CSS відповідають за відображення сторінки та її дизайн, тоді як JavaScript виконує основні дії, пов'язані з взаємодією користувача. Наприклад, JS обробляє натискання кнопок, введення даних та навігацію, забезпечуючи інтерактивність і динамічність сторінок.

Зворотний зв'язок з користувачем забезпечується через різні візуальні ефекти та повідомлення. Повідомлення про успішне додавання товару та помилки виводилися за допомогою попапів. Це забезпечувало безперебійний та комфортний процес покупок для користувачів, надаючи їм можливість легко керувати своїм кошиком та списком обраного, а також оформлювати замовлення на вебсайті “Моє дитя”.

Таким чином, клієнтська частина вебсайту забезпечує зручність користувачів, інтерактивність і привабливий дизайн за допомогою технологій JavaScript, HTML та CSS, а також бібліотек jQuery та Ајах.

3.3 Інтеграція бази даних MySQL

Інтеграція бази даних MySQL у вебсайті “Моє дитя” відіграє ключову роль у забезпеченні зберігання та обробки даних користувачів, товарів та замовлень. У цьому розділі розглянемо процес підключення до бази даних, основні SQL-запити та заходи безпеки.

Підключення до бази даних MySQL було реалізовано за допомогою PHP. Для підключення використовуються такі параметри, як ім'я сервера, ім'я користувача, пароль та ім'я бази даних. Приклад коду підключення до бази даних наведено в лістингу 3.10.

Лістинг 3.10 – Підключення до БД MySQL

```
$servername = "*****";
$username = "*****";
$password = "*****";
$dbname = "*****";

$conn = new mysqli($servername, $username, $password, $dbname);
mysqli_set_charset($conn, "utf8");

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
```

У цьому коді здійснюється підключення до бази даних MySQL, встановлення кодування UTF-8 та перевірка з'єднання. Якщо з'єднання не вдалося, виводиться повідомлення про помилку.

Основні SQL-запити, що використовуються для взаємодії з базою даних, включають вибірку, оновлення та видалення даних. Наприклад, запит для отримання списку обраних товарів користувача з бази даних наведено нижче.

```
$user_id = $_SESSION['user_id'];
$sql = "SELECT favorites FROM users WHERE id = $user_id";
$result = $conn->query($sql);
```

Цей запит отримує список обраних товарів для конкретного користувача на основі його ID, збереженого в сесії. Ще один приклад SQL-запиту для отримання товарів з таблиці products описано нижче.

```
$product_ids_string = implode(",", $product_ids);
$sql = "SELECT * FROM products WHERE id IN ($product_ids_string)";
$products_result = $conn->query($sql);
```

Цей запит отримує всі товари, чий ID вказані у масиві `$product_ids`. Це дозволяє вибирати конкретні товари з бази даних для відображення на сторінці.

Заходи безпеки включають хешування паролів користувачів за допомогою функції `password_hash`, використання підготовлених запитів для захисту від SQL-ін'єкцій та контроль доступу до бази даних. Наприклад, для збереження паролю користувача використовується код, який наведено в лістингу 3.11.

Цей код хешує пароль користувача перед збереженням у базі даних, що забезпечує додатковий рівень безпеки. Використання `password_hash` гарантує, що паролі зберігаються у безпечному вигляді, унеможливаючи їх легке зламування.

Лістинг 3.11 – Хешування паролю користувача

```
$password = "пароль користувача";
$hashedPassword = password_hash($password, PASSWORD_DEFAULT);
$sql = "INSERT INTO users (phone_number, password) VALUES
('$phoneNumber', '$hashedPassword')";
if ($conn->query($sql) === TRUE) {
    echo "Користувача успішно зареєстровано!";
} else {
    echo "Помилка: " . $conn->error;
}
```

Крім того, використовуються підготовлені запити для запобігання SQL-ін'єкціям. Це забезпечує безпечну взаємодію з базою даних, оскільки параметри запитів передаються окремо від SQL-коду (див. лістинг 3.12).

Лістинг 3.12 – Використання підготовлених запитів

```
$stmt = $conn->prepare("SELECT product_name, price, image FROM
products WHERE global_group = ? AND price BETWEEN ? AND ?");
$stmt->bind_param("sdd", $group, $min_price, $max_price);
$stmt->execute();
$result = $stmt->get_result();
```

У цьому прикладі використовується підготовлений запит для вибірки товарів, що відповідають певній групі та ціновому діапазону. Це забезпечує додатковий рівень безпеки та гнучкість у написанні SQL-запитів.

Отже інтеграція бази даних MySQL на вебсайті забезпечує надійне зберігання та обробку даних, забезпечуючи високу продуктивність та безпеку системи. Використання хешування паролів, підготовлених запитів та інших заходів безпеки гарантує захист даних користувачів та стабільну роботу магазину.

3.4 Налаштування серверного оточення та хостинг

Для налаштування серверного оточення та хостингу вебсайту “Моє дитя” було використано хостинг-провайдера InfinityFree, який пропонує безкоштовний хостинг з підтримкою PHP та MySQL. Вибір цього хостинг-провайдера було описано в розділі 1.4.

Налаштування серверного оточення починалося з конфігурації веб-сервера. У файлі конфігурації Apache було вказано основні налаштування для домену вебсайту. Приклад конфігурацій наведено в лістингу 3.13.

Лістинг 3.13 – Конфігурація Apache

```
<VirtualHost *:80>
    ServerAdmin moye-dytya@gmail.com
    DocumentRoot /moye-dytya.000.pe/htdocs
    ServerName moye-dytya.000.pe
    ServerAlias www.moye-dytya.000.pe

    <Directory /moye-dytya.000.pe/htdocs>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Ця конфігурація забезпечувала належну роботу веб-сервера та правильне відображення вмісту сайту. Параметр ServerAdmin вказує електронну адресу адміністратора сайту. DocumentRoot визначає кореневу директорію, де

знаходяться файли сайту. `ServerName` та `ServerAlias` встановлюють основне та альтернативне ім'я домену відповідно. Директорія `/moye-dytya.000.pe/htdocs` налаштовується так, щоб дозволяти індексацію, слідувати символічним посиланням та дозволяти перезаписування налаштувань.

Для забезпечення безпеки та надійності хостингу також було налаштовано регулярне резервне копіювання даних, щоб уникнути втрати важливої інформації у разі технічних збоїв або інших непередбачених обставин. Це включало автоматичне створення резервних копій баз даних і файлів сайту з можливістю їх відновлення в разі потреби.

Також було налаштовано розмежування доступу до файлів і директорій за допомогою налаштувань у файлі `.htaccess`. Цей файл дозволяв обмежити доступ до деяких частин сайту, запобігаючи несанкціонованому доступу до адміністративних і конфіденційних даних. Наприклад, заборонялося пряме доступ до конфігураційних файлів:

Лістинг 3.14 – Налаштування обмеження доступу до конфіденційних файлів у конфігурації Apache

```
<FilesMatch "\.(htaccess|htpasswd|ini|log|sh|sql)$">  
    Order Allow,Deny  
    Deny from all  
</FilesMatch>
```

Для поліпшення продуктивності було використано кешування на стороні сервера. Включення кешування статичних файлів, таких як зображення, стилі CSS та скрипти JavaScript, допомогло зменшити навантаження на сервер і прискорити завантаження сторінок для користувачів.

Після налаштування серверного оточення та хостингу інтернет-магазин був готовий до роботи, забезпечуючи надійне зберігання та обробку даних користувачів, а також високий рівень продуктивності та безпеки.

3.5 Тестування функціональності та забезпечення якості

Тестування вебсайту є важливим етапом розробки, який включає перевірку його функціональності, продуктивності, юзабіліті та безпеки. Мета тестування полягає в тому, аби забезпечити відповідність продукту вимогам проєкту та всім технічним стандартам, а також задовольнити потреби кінцевих користувачів [29].

Процес тестування розпочався з перевірки основних функцій сайту на головній сторінці та під час авторизації користувачів. Перейшовши на головну сторінку вебсайту та виконавши авторизацію, було визначено, що коректність та функціонал знаходяться на високому рівні, як показано на рисунку 3.1. Важливо було переконатися, що користувачі можуть успішно увійти на сайт, використовуючи свої облікові дані, та отримати доступ до персоналізованих функцій.

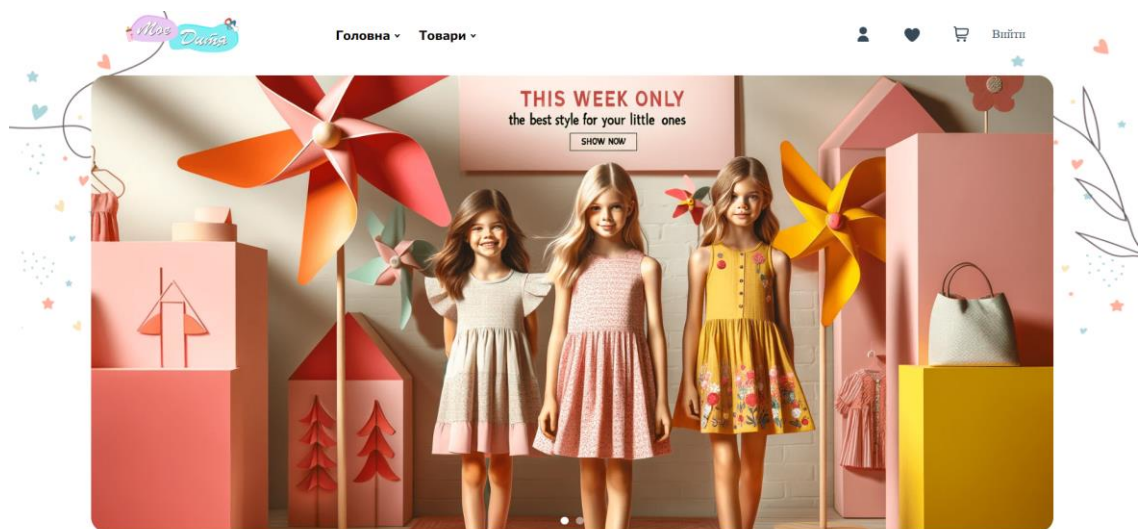


Рисунок 3.1 – Зображення працездатності вебсайту

Далі було проведено комплексне тестування всіх інших сторінок та вкладок сайту. Здійснивши перевірку баз даних, ми переконалися, що всі дані коректно зберігаються та обробляються, а всі функції, такі як реєстрація, аутентифікація, пошук товарів, сортування та фільтрація, працюють належним

чином. Особливу увагу було приділено рядку пошуку та функції розшифровки пропозицій товарів, що дозволяє користувачам швидко знаходити необхідні товари. Під час тестування продуктивності було перевірено швидкість завантаження сторінок і обробку запитів. Сайт продемонстрував високу продуктивність, забезпечуючи швидкий відгук навіть при високому навантаженні. Це досягнуто завдяки оптимізації коду та налаштуванню серверного оточення.

Проведений аналіз завантаження ресурсів сайту за допомогою інструментів розробника в браузері (Developer Tools) показав, що значна частина ресурсів завантажується успішно з HTTP-статусом 200, це можна переглянути на рисунку 3.2.

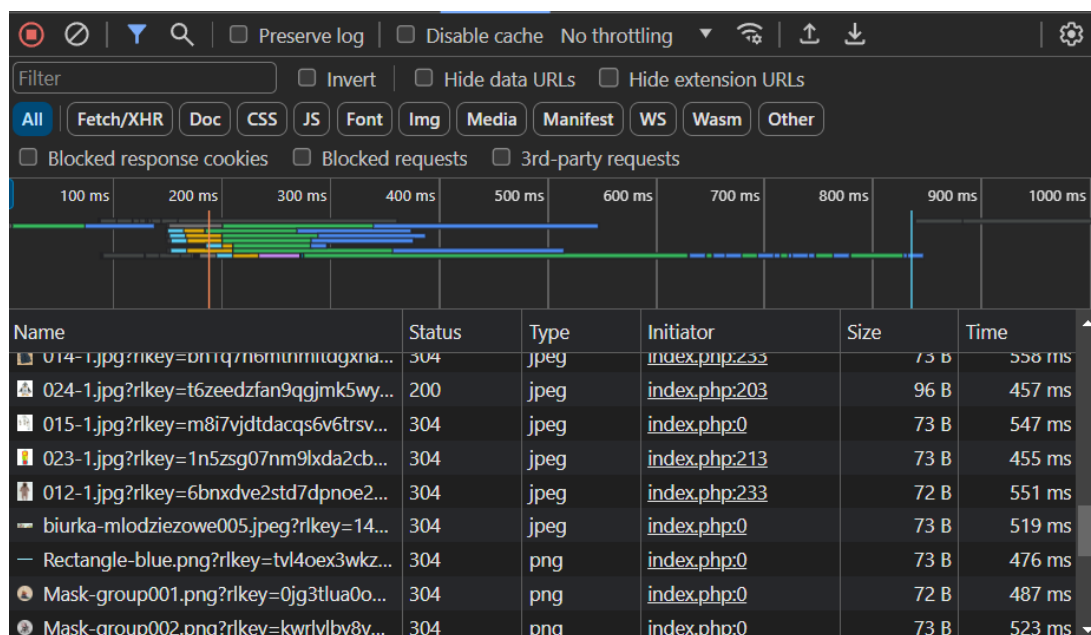


Рисунок 3.2 – Аналіз завантаження ресурсів сайту

Основні ресурси, такі як файли JavaScript, шрифти, зображення, та PHP-скрипти, мають різний час завантаження, що впливає на загальну продуктивність сайту. Графік часу завантаження показав, що деякі ресурси займають значний час для завантаження, зокрема зображення, що може бути областю для оптимізації. Цей аналіз допоміг ідентифікувати проблеми з

продуктивністю, такі як довгий час завантаження великих файлів або некешовані ресурси, які потім було вирішено.

Забезпечення безпеки також було важливим аспектом тестування. Перевірка наявності вразливостей та захисту даних користувачів показала, що сайт відповідає високим стандартам безпеки. Використання хешування паролів та захист від SQL-ін'єкцій забезпечує надійність зберігання та обробки персональних даних користувачів.

Важливо зазначити, що перед початком роботи були визначені основні умови, яким повинен відповідати вебсайт: реєстрація та аутентифікація користувачів, пошук товарів, сортування та фільтрація, рядок пошуку та функція розшифровки пропозицій товарів, можливість додавати, видаляти та редагувати товари в кошику. Всі ці вимоги були враховані та успішно реалізовані під час розробки та тестування.

Таким чином, наш вебсайт “Моє дитя” відповідає всім вимогам, поставленим на початку роботи. Проведене тестування підтвердило високу якість і надійність функціонування сайту, забезпечуючи комфортне та безпечне користування для кінцевих користувачів.

3.6 Висновки до третього розділу

У третьому розділі було здійснено повну реалізацію вебсайту “Моє дитя” на базі сучасних веб-технологій. Серверну частину реалізовано за допомогою мови програмування PHP, що забезпечує надійну обробку запитів від клієнтів, взаємодію з базою даних MySQL та підтримку безпеки системи. Було детально описано основні аспекти фільтрації товарів, додавання товарів до обраного та кошика, а також інші важливі функції, що забезпечують безперебійний процес покупок для користувачів.

Клієнтська частина була створена з використанням HTML, CSS та JavaScript, що забезпечило привабливий дизайн, зручність користування та інтерактивність вебсайту. Для спрощення роботи з DOM та асинхронними

запитами використовувалися бібліотеки jQuery та Ajax. Завдяки адаптивному дизайну сайт комфортно відображається на різних пристроях, що покращує користувацький досвід.

Інтеграція бази даних MySQL забезпечила надійне зберігання та обробку даних користувачів. Впроваджені заходи безпеки, такі як хешування паролів та захист від SQL-ін'єкцій, гарантують надійність і безпеку персональних даних користувачів. Це дозволяє забезпечити високий рівень захисту інформації та підтримувати довіру клієнтів. Налаштування серверного оточення та хостинг на базі InfinityFree забезпечили стабільну роботу вебсайту з високою продуктивністю.

Проведене тестування функціональності та забезпечення якості підтвердило високу надійність і продуктивність вебсайту. Всі функції, такі як реєстрація, авторизація, пошук та фільтрація товарів, додавання до кошика та оформлення замовлень, працюють коректно. Забезпечення безпеки та відповідність технічним стандартам гарантує захист даних користувачів.

Таким чином, вебсайт відповідає всім вимогам та стандартам якості, які були поставленні ще на початку роботи, а саме реєстрація та аутентифікація користувачів, пошук товарів, сортування та фільтрація, рядок пошуку та функція розшифровки пропозицій товарів, а також можливість додавати, видаляти та редагувати товари в кошику. Це забезпечує комфортне та безпечне користування для кінцевих користувачів. Інтеграція сучасних технологій та ретельне тестування функціональності дозволяють надавати високоякісний сервіс, що задовольняє потреби та очікування клієнтів.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Долікарська медична допомога при захворюваннях, травмах та в умовах надзвичайних ситуацій

Долікарська медична допомога є важливим елементом у наданні негайної допомоги потерпілим до прибуття медичних працівників. Правильне і вчасне надання першої допомоги може значно зменшити ризик ускладнень, прискорити процес одужання, запобігти інвалідності та, у багатьох випадках, врятувати життя [30]. В сучасних умовах, коли існує висока ймовірність різного роду нещасних випадків, таких як осколкові поранення чи інші травми, викликані бойовими діями, дуже важливо мати базові знання і навички надання першої допомоги.

Станом на сьогоднішній час, якщо стався якийсь нещасний випадок чи осколки скла чи ще чогось попали в тіло людини, першим завданням є стабілізація стану потерпілого до прибуття медичних працівників. Допомога повинна включати:

1. Перев'язування рани.
2. Нерухома фіксація при тяжких травмах.
3. Оживляючі заходи.
4. Видача знеболюючих засобів.
5. Перенесення і транспортування потерпілих.

Ознаки життя включають:

- серцебиття;
- пульс;
- дихання;
- реакція зіниці на світло.

У випадку поранень, які можуть статись при паданні уламків будівлі чи ще якихось матеріалів, які є порушенням цілісності шкіри та слизових

оболонок, і можуть супроводжуватися пошкодженням м'язів, судин, нервів та внутрішніх органів, необхідно вжити таких заходів:

1. Підняття кінцівки та притискання артерії.
2. Накладання кровоспинного джгута або тугої пов'язки.
3. Накриття рани стерильною серветкою та закріплення її. Особливу увагу потрібно приділити часу накладання джгута – не більше однієї години, щоб уникнути пошкодження тканин.

Переломи є порушенням цілісності кістки, і можуть бути повними або неповними (тріщини, надломи). Переломи поділяються на:

- закриті;
- відкриті.

Заходи при закритих переломах:

1. Забезпечити спокій потерпілому.
2. Надати знеболюючий засіб.
3. Визначити можливість переміщення постраждалого.
4. Імобілізувати пошкоджену ділянку.

Заходи при відкритих переломах:

1. Зупинити кровотечу.
2. Обробити рану.
3. Накласти бинтову пов'язку.
4. Імобілізувати пошкоджену ділянку.

Належне виконання цих заходів є запорукою успішного надання першої долікарської допомоги та зменшення ризику подальших ускладнень для потерпілого [31]. В умовах сучасних реалій, особливо під час війни, необхідність володіння навичками надання долікарської медичної допомоги стає надзвичайно актуальною. Це стосується як розробників, що працюють над створенням дитячого вебсайту “Моє дитя”, так і самих користувачів цього ресурсу. Будь-яка людина, незалежно від професії чи місця перебування, може опинитися в ситуації, коли потрібне негайне втручання до прибуття медиків.

Крім того, навчання та поширення інформації про першу допомогу серед користувачів вебсайту сприятиме підвищенню загальної обізнаності та підготовленості суспільства. Адже під час війни та інших надзвичайних ситуацій знання про те, як правильно надати першу допомогу, є не просто корисним – воно є необхідним. Важливо пам'ятати, що навіть базові знання першої допомоги можуть значно вплинути на стан потерпілого і забезпечити йому шанс на подальше життя і відновлення.

Зокрема, в умовах сучасних реалій, де ризики отримання травм або поранень зростають, знання та навички надання першої допомоги стають не лише корисними, а й необхідними для виживання. Війна та інші надзвичайні ситуації можуть поставити будь-яку людину в умови, де відсутність медичної допомоги протягом тривалого часу може мати фатальні наслідки. Отже, розуміння основних принципів та методик надання першої допомоги є ключовим фактором для виживання та підтримання здоров'я в критичних ситуаціях.

4.2 Вимоги безпеки до робочих місць для виконання робіт.

При організації робочих місць важливо враховувати фізичне навантаження, яке відчують працівники. Це є надзвичайно важливим аспектом, який впливає на здоров'я та продуктивність співробітників. Для роботи, яка потребує високої фізичної активності, краще використовувати стояче положення, тоді як для менш інтенсивної роботи – сидяче. Зміна позиції протягом робочого дня допомагає зменшити втому та запобігти розвитку професійних захворювань.

Навантаження на м'язи змінюється в залежності від положення тіла. Якщо прийняти навантаження в прямій сидячій позі за одиницю, то у прямій стоячій воно збільшується в 1,6 рази. В похилій сидячій позі навантаження зростає в 4 рази, а в похилій стоячій – в 10 разів. Ці дані свідчать про те, що неправильна

постава може значно підвищити фізичне навантаження, що призводить до швидшої втоми і потенційних травм [32].

Робота у стоячому положенні призводить до більшої втоми, тому важливо мати можливість змінювати позицію під час роботи. Часті зміни положення допомагають уникнути професійних захворювань, таких як варикозне розширення вен при тривалому стоянні або геморої при тривалій роботі сидячи. Практика зміни положення також допомагає підтримувати нормальну циркуляцію крові та зменшувати навантаження на окремі групи м'язів.

Крісло для сидячої роботи повинно відповідати вимогам ДСТУ 8604:2015. Це означає, що крісло має бути ергономічним, забезпечувати підтримку спини та знижувати навантаження на хребет. Сидіння має бути регульованим по вертикалі та горизонталі, мати підлокітники, опору для спини і підставку для ніг. Регулювання висоти і нахилу сидіння допомагає підлаштувати робоче місце під індивідуальні потреби кожного працівника, що сприяє більшому комфорту і продуктивності [33].

Робочі місця для стоячої роботи повинні відповідати ДСТУ 7950:2015. Це включає в себе не тільки правильну висоту робочої поверхні, але й можливість регулювання висоти столу. Ручні органи управління повинні розташовуватись на висоті 1000—1600 мм від рівня підлоги для стоячої роботи і 600—1200 мм для сидячої. Це дозволяє працівникам зручно дотягуватися до необхідного обладнання, знижуючи ризик травмування та покращуючи ефективність роботи.

Потрібно враховувати антропометричні дані при проектуванні робочих місць і обладнання, беручи до уваги фізіологічні можливості людини. Знання про середні розміри і фізичні можливості різних працівників допомагають створити робоче місце, яке буде комфортним і безпечним для всіх. Це включає правильне розташування елементів управління, підбору меблів та інструментів, що відповідають потребам працівників.

Критерії оцінки умов праці визначені в Гігієнічній класифікації праці за показниками шкідливості і небезпечності факторів виробничого середовища,

важкості й напруженості трудового процесу (наказ МОЗ України від 05.01.2022 р. № 7). Цей документ використовується для гігієнічної оцінки умов праці, встановлення пріоритетів оздоровчих заходів, розробки рекомендацій щодо професійного відбору, прогнозування професійних захворювань, створення бази даних про умови праці, атестації робочих місць, впровадження диференційованих внесків у Фонд соціального страхування та визначення економічних санкцій за несприятливі умови праці [34].

Основні вимоги до робочого місця включають забезпечення комфорту для працівників. Робоче місце повинно бути спроектоване так, щоб мінімізувати фізичне навантаження на працівника та забезпечити зручність під час виконання завдань. Наприклад, стіл для роботи за комп'ютером має дозволяти змінювати позицію з сидячої на стоячу і навпаки, що допомагає зменшити навантаження та уникнути професійних захворювань. Також важливо, щоб робоче місце було обладнане належним освітленням, що знижує навантаження на зір і допомагає підтримувати високу продуктивність.

Гігієна на робочому місці також є важливою складовою безпечної праці. Робоче місце повинно бути чистим, добре вентильованим і безпечним. Це включає регулярне прибирання, забезпечення доступу до питної води, а також наявність засобів особистої гігієни. Дотримання цих вимог допомагає зменшити ризик захворювань та покращити загальний стан здоров'я працівників [35].

4.3 Висновок до четвертого розділу

В четвертому розділі кваліфікаційної роботи розглянуто актуальність долікарської медичної допомоги при захворюваннях, травмах та в умовах надзвичайних ситуацій а також вимоги безпеки до робочих місць для виконання робіт.

Правильне і своєчасне надання першої допомоги значно зменшує ризик ускладнень, прискорює одужання, запобігає інвалідності та може врятувати

життя. В умовах надзвичайних ситуацій, таких як війна, ці знання є особливо актуальними і важливими для всіх, включаючи розробників та користувачів вебсайту “Моє дитя”.

Правильна організація робочих місць, особливо для роботи за комп'ютером, з можливістю зміни позиції з сидячої на стоячу, зменшує втому та запобігає професійним захворюванням. Дотримання стандартів ДСТУ та забезпечення комфортних умов праці є критичними.

ВИСНОВКИ

У кваліфікаційній роботі освітнього рівня “Бакалавр” проведено детальний аналіз вимог та задач для розробки вебсайту дитячого магазину “Моє Дитя”. Розглянуто аналіз існуючих рішень, формулювання вимог до функціональності та дизайну, вибір технологій для розробки, створення архітектури системи, розробку інтерфейсу, інтеграцію бази даних та тестування функціональності вебсайту. Весь процес розробки був спрямований на створення зручного, функціонального та безпечного вебсайту, який відповідає потребам сучасних користувачів.

У першому розділі проведено порівняльний аналіз вебсайтів дитячих магазинів, виявлено їхні основні переваги та недоліки. Описано функціональні та дизайнерські вимоги до створення нового вебсайту, включаючи інтерфейс, навігацію, швидкість завантаження. Визначено ключові критерії для забезпечення ефективної роботи магазину. Основною метою було визначити, як зробити процес покупок зручнішим і ефективнішим для користувачів, враховуючи їхні потреби та очікування.

Другий розділ присвячено проектуванню вебсайту. Визначено архітектуру системи, яка забезпечує зручність навігації та доступність основних функцій для користувачів. Розроблено інтерфейс з урахуванням зручності користувачів та естетичної привабливості. Проведено моделювання бази даних для зберігання інформації про користувачів, товари та замовлення.

У третьому розділі розглянуто реалізацію вебсайту на основі PHP, JavaScript, HTML та CSS. Описано процес створення серверної та клієнтської частин сайту, інтеграцію бази даних MySQL, налаштування серверного оточення та хостингу. Проведено тестування функціональності, яке підтвердило відповідність вебсайту всім поставленим вимогам та високий рівень його якості. Забезпечено оптимізацію коду та налаштування серверного оточення для забезпечення високої продуктивності сайту навіть при високому навантаженні.

У розділі «Безпека життєдіяльності, основи охорони праці» розглянуто надання долікарської медичної допомоги при захворюваннях, травмах та в умовах надзвичайних ситуацій. Описано вимоги безпеки до робочих місць, включаючи ергономіку, гігієну та організацію праці. Визначено ключові заходи для забезпечення здорових і безпечних умов праці, що знижує ризики для здоров'я працівників і підвищує ефективність їхньої роботи.

Таким чином, у кваліфікаційній роботі закладено основи для створення ефективного та зручного вебсайту дитячого магазину “Моє Дитя”, що відповідає сучасним вимогам користувачів та забезпечує безпеку та комфорт під час роботи. Створено зручний інтерфейс, забезпечено швидкість завантаження, інтеграцію з базою даних та високу продуктивність сайту. Це сприяє покращенню процесу покупки дитячих товарів, економії часу та зусиль батьків, а також підвищенню рівня обслуговування клієнтів. Впровадження результатів роботи дозволило створити конкурентоспроможний продукт, який задовольняє потреби ринку і забезпечує стабільний розвиток дитячого магазину.

ПЕРЕЛІК ДЖЕРЕЛ

1. CMS системи як сучасний засіб розробки веб-сайтів [Електронний ресурс] – Режим доступу до ресурсу: <https://elartu.tntu.edu.ua/bitstream/123456789/4981/1/225-226.pdf> (дата звернення 03.06.2024).
2. Чому вигідно купувати дитячі товари в інтернет-магазині [Електронний ресурс] – Режим доступу до ресурсу: <https://kidsstore.com.ua/chomu-vyhidno-kupuvaty-dytiachi-tovary-v-internet-mahazyni/> (дата звернення 04.06.2024).
3. Чому вигідно купити дитячі іграшки та товари для дітей в інтернет-магазині? [Електронний ресурс] – Режим доступу до ресурсу: <https://rozvyvashky.com.ua/chomu-vyhidno-kupyty-dytiachi-ihrashky-ta-tovary-dlia-ditei-v-internet-mahazyni/> (дата звернення 04.06.2024).
4. Безпека програмного забезпечення [Електронний ресурс] – Режим доступу до ресурсу: <https://www.44.ua/news/3698042/bezpeka-programnogo-zabezpecenna-sucasni-zagrozi-ta-metodi-zahistu> (дата звернення 05.06.2024).
5. Посібник із узгодженості дизайну забезпечення [Електронний ресурс] – Режим доступу до ресурсу: <https://www.uxpin.com/studio/blog/guide-design-consistency-best-practices-ui-ux-designers/> (дата звернення 05.06.2024).
6. Що таке PHP? [Електронний ресурс] – Режим доступу до ресурсу: <https://hyperhost.ua/uk/wiki/chto-takoe-php> (дата звернення 05.06.2024).
7. Найкращі практики PHP для веб-розробників [Електронний ресурс] – Режим доступу до ресурсу: <https://www.etatvasoft.com/blog/php-best-practices/> (дата звернення 05.06.2024).
8. Javascript (JS): що це таке, для чого потрібен? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.ukrlib.com.ua/pub/article.php?articleid=1990> (дата звернення 05.06.2024).

9. HTML: мова розмітки гіпертексту [Електронний ресурс] – Режим доступу до ресурсу: <https://webdoky.org/uk/docs/Web/HTML/> (дата звернення 08.06.2024).
10. CSS мова опису стилю сторінок [Електронний ресурс] – Режим доступу до ресурсу: <https://www.wikiwand.com/uk/CSS> (дата звернення 08.06.2024).
11. База даних MySQL [Електронний ресурс] – Режим доступу до ресурсу: <https://promoter.net.ua/articles/baza-danix-mysql.html> (дата звернення 05.06.2024).
12. Розробка та дослідження системи управління вмістом веб – ресурсу [Електронний ресурс] – Режим доступу до ресурсу: https://elartu.tntu.edu.ua/bitstream/123456789/15977/2/Conf_2008_Mlinko_B-Rozrobka_ta_doslidzhennia_systemy_107.pdf (дата звернення 11.06.2024).
13. Що таке Хостинг? [Електронний ресурс] – Режим доступу до ресурсу: <https://romankoff.com/blog/it-dovidnyk/hosting> (дата звернення 08.06.2024).
14. Як обрати якісний хостинг в Україні [Електронний ресурс] – Режим доступу до ресурсу: <https://capital360.com.ua/sponsory/yak-obraty-yakisnyu-hostynh-v-ukraini-porady-ta-rekomendatsii/> (дата звернення 11.06.2024).
15. Що таке ХАМРР? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.apachefriends.org/> (дата звернення 08.06.2024).
16. Що таке Infinityfree [Електронний ресурс] – Режим доступу до ресурсу: <https://itest.com.ua/instrumenty/infinityfree/> (дата звернення 05.06.2024).
17. Середовище розробки Microsoft Visual Studio [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.ztu.edu.ua/mod/page/view.php?id=9974> (дата звернення 08.06.2024).
18. FileZilla: Посібник з налаштування та використання для початківців [Електронний ресурс] – Режим доступу до ресурсу: <https://infobriz.com.ua/tehnika/filezilla-posibnyk-z-nalashtuvannya-ta-vykorystannya-dlya-pochatkivciv/> (дата звернення 05.06.2024).

19. Що таке Figma і навіщо вона потрібна [Електронний ресурс] – Режим доступу до ресурсу: <https://lemon.school/blog/chtotakoe-figma-i-zachem-ona-nuzhna> (дата звернення 05.06.2024).
20. Діаграма варіантів використання (use case diagram) [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/9828815/> (дата звернення 06.06.2024).
21. Що таке бази даних? [Електронний ресурс] – Режим доступу до ресурсу: <https://university.sigma.software/what-is-database/> (дата звернення 08.06.2024).
22. Введення в UI дизайн: створення ефективних та привабливих інтерфейсів [Електронний ресурс] – Режим доступу до ресурсу: https://cloud.itstep.org/blog_3/introduction-to-ui-design-creating-effective-and-attractive-interfaces (дата звернення 08.06.2024).
23. 13 найкращих практик веб-дизайну для покращення UX [Електронний ресурс] – Режим доступу до ресурсу: <https://www.hotjar.com/web-design/best-practices/> (дата звернення 12.06.2024).
24. Dropbox – що це і як ним користуватися [Електронний ресурс] – Режим доступу до ресурсу: <https://smartandyoung.com.ua/dropbox-shho-ce-i-jak-nim-koristuvatisja> (дата звернення 06.06.2024).
25. Що таке Back-end розробка? [Електронний ресурс] – Режим доступу до ресурсу: <https://wezom.com.ua/ua/blog/chtotakoe-back-end-razrabotka> (дата звернення 06.06.2024).
26. Розробка та просування веб-проектів [Електронний ресурс] – Режим доступу до ресурсу: <https://sitepractic.jimdofree.com/середовища-розробки/frontend-та-backend/> (дата звернення 07.06.2024).
27. Що таке JQuery? [Електронний ресурс] – Режим доступу до ресурсу: <https://jquery.com/> (дата звернення 08.06.2024).
28. Отримання даних із сервера [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Client-side_web_APIs/Fetching_data (дата звернення 09.06.2024).

29. Все що потрібно знати про тестування сайтів [Електронний ресурс] – Режим доступу до ресурсу: <https://wezom.com.ua/ua/blog/kak-protestirovat-sajt> (дата звернення 08.06.2024).

30. Перша долікарська допомога при нещасних випадках [Електронний ресурс] – Режим доступу до ресурсу: https://pidru4niki.com/1574022538230/bzhd/persha_dolikarska_dopomoga_pri_nesc_hasnih_vipadkah (дата звернення 04.06.2024).

31. Правила надання першої медичної допомоги [Електронний ресурс] – Режим доступу до ресурсу: https://health.24tv.ua/persha_medichna_dopomoga_pravila_yiyi_nadannya_n861201 (дата звернення 04.06.2024).

32. Вимоги безпеки до робочого місця. Ергономічні вимоги [Електронний ресурс] – Режим доступу до ресурсу: https://pidru4niki.com/16850303/bzhd/vimogi_bezpeki_robochogo_mistsya_ergonomichni_vimogi (дата звернення 09.06.2024).

33. Система стандартів безпеки праці Робоче місце при виконанні робіт сидячи [Електронний ресурс] – Режим доступу до ресурсу: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=28895 (дата звернення 08.06.2024).

34. Гігієнічної класифікації праці за показниками шкідливості та небезпечності факторів виробничого середовища [Електронний ресурс] – Режим доступу до ресурсу: <https://ips.ligazakon.net/document/MOZ1630> (дата звернення 08.06.2024).

35. Ергономічні вимоги до організації робочих місць [Електронний ресурс] – Режим доступу до ресурсу: https://pidru4niki.com/14821111/bzhd/ergonomichni_vimogi_organizatsiyi_robochih_mists (дата звернення 04.06.2024).

ДОДАТКИ

Фільтрація товарів

```

$group      =      isset($_GET['menu-link'])      ?      $_GET['menu-link']      :
      (isset($_SESSION['group']) ? $_SESSION['group'] : 'іграшки');
$active_checkbox_groups = $checkbox_groups_by_category[$group];
$_SESSION['group'] = $group;

$servername = "sql311.infinityfree.com";
$username = "if0_36202968";
$password = "Qwerty193758246";
$dbname = "if0_36202968_mychild";

$conn = new mysqli($servername, $username, $password, $dbname);
mysqli_set_charset($conn, "utf8");

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$search = $_GET['search'] ?? '';
$min_price = isset($_GET['min_price']) ? (float)$_GET['min_price'] :
    null;
$max_price = isset($_GET['max_price']) ? (float)$_GET['max_price'] :
    null;

$query = "SELECT product_name, price, image FROM products WHERE
    (product_name LIKE '%$search%' OR global_group = '$group') AND
    (global_group = '$group') AND count_products > 0";
$result = mysqli_query($conn, $query);

$total_products = mysqli_num_rows($result);

$per_page = 28;
$num_pages = ceil($total_products / $per_page);
$current_page = isset($_GET['page']) ? (int)$_GET['page'] : 1;
$offset = ($current_page - 1) * $per_page;

$query = "SELECT product_name, price, image FROM products WHERE
    (product_name LIKE '%$search%' OR global_group = '$group') AND
    (global_group = '$group') AND count_products > 0 LIMIT $offset,
    $per_page";
$result = mysqli_query($conn, $query);

$where_conditions = array();
foreach ($active_checkbox_groups as $category => $checkboxes) {
    $group_conditions = array();
    foreach ($checkboxes as $checkbox_id => $checkbox_label) {
        if (isset($_GET[$checkbox_id])) {
            $group_conditions[] = "characteristics LIKE '%" .
                $checkbox_label . "%'";
        }
    }
    if (!empty($group_conditions)) {

```

```

        $where_conditions[] = "(" . implode(" OR ",
$group_conditions) . ")";
    }
}

$checkbox_condition = '';
if (!empty($where_conditions)) {
    $checkbox_condition = " AND (" . implode(" AND ",
$where_conditions) . ")";
}

$query = "SELECT product_name, price, image FROM products WHERE
(product_name LIKE '%$search%' OR global_group = '$group')
$checkbox_condition AND (global_group = '$group') AND count_products
> 0";
$result = mysqli_query($conn, $query);

$filtered_total_products = mysqli_num_rows($result);

$num_pages = ($filtered_total_products <= $per_page) ? 1 :
ceil($filtered_total_products / $per_page);
$offset = ($current_page - 1) * $per_page;

$price_condition = '';
if ($min_price !== null || $max_price !== null) {
    $price_condition = "AND price BETWEEN ";
    if ($min_price !== null) {
        $price_condition .= $min_price;
    } else {
        $price_condition .= "0";
    }
    $price_condition .= " AND ";
    if ($max_price !== null) {
        $price_condition .= $max_price;
    } else {
        $price_condition .= "1000";
    }
}

$query = "SELECT id ,product_name, price, image FROM products WHERE
(product_name LIKE '%$search%' OR global_group = '$group')
$checkbox_condition AND (global_group = '$group') $price_condition
AND count_products > 0";
$result = mysqli_query($conn, $query);

$group_condition = ($group != '') ? "AND global_group = '$group'" : "";
$group_query = "SELECT global_group, COUNT(*) as count FROM products
WHERE (product_name LIKE '%$search%' OR global_group = '$group')
$checkbox_condition $group_condition AND count_products > 0 GROUP BY
global_group ORDER BY count DESC LIMIT 1";
$group_result = mysqli_query($conn, $group_query);
$largest_group_info = mysqli_fetch_assoc($group_result);

if ($largest_group_info !== null) {
} else {
    $largest_group_info['global_group'] =
isset($_SESSION['last_group']) ? $_SESSION['last_group'] : $group;
}

```

```

$_SESSION['last_group'] = $largest_group_info['global_group'];

if ($largest_group_info !== null) {
    if ($largest_group_info['global_group'] == 'одяг') {
        echo '<style>.clothes { display: block; }</style>';
        echo '<style>.toys { display: none; }</style>';
        echo '<style>.food { display: none; }</style>';
        echo '<style>.strollers { display: none; }</style>';
        echo '<style>.cot { display: none; }</style>';
    } else if ($largest_group_info['global_group'] == 'іграшки') {
        echo '<style>.clothes { display: none; }</style>';
        echo '<style>.toys { display: block;}</style>';
        echo '<style>.food { display: none; }</style>';
        echo '<style>.strollers { display: none; }</style>';
        echo '<style>.cot { display: none; }</style>';
    } else if ($largest_group_info['global_group'] == 'харчування') {
        echo '<style>.clothes { display: none; }</style>';
        echo '<style>.toys { display: none;}</style>';
        echo '<style>.food { display: block; }</style>';
        echo '<style>.strollers { display: none; }</style>';
        echo '<style>.cot { display: none; }</style>';
    } else if ($largest_group_info['global_group'] == 'коляски') {
        echo '<style>.clothes { display: none; }</style>';
        echo '<style>.toys { display: none;}</style>';
        echo '<style>.food { display: none; }</style>';
        echo '<style>.strollers { display: block; }</style>';
        echo '<style>.cot { display: none; }</style>';
    } else if ($largest_group_info['global_group'] == 'ліжечка') {
        echo '<style>.clothes { display: none; }</style>';
        echo '<style>.toys { display: none;}</style>';
        echo '<style>.food { display: none; }</style>';
        echo '<style>.strollers { display: none; }</style>';
        echo '<style>.cot { display: block; }</style>';
    }
} else {
}

$info = [];

if (mysqli_num_rows($result) > 0) {
    while ($row = mysqli_fetch_assoc($result)) {
        $info[] = $row;
    }
}

```

Загальні функції JS

```

function validateForm() {
    var initials = document.getElementById("initials").value.trim();
    var
        phoneNumber
document.getElementById("phone_number").value.trim();
    var
        deliveryCity
document.getElementById("delivery_city").value.trim();
    var
        deliveryMethod
document.querySelector('input[name="delivery_method"]:checked');

    if (!deliveryMethod) {
        alert("Будь ласка, оберіть метод доставки.");
        return false;
    }

    if (initials === "" || phoneNumber === "" || deliveryCity === "") {
        alert("Будь ласка, заповніть всі поля.");
        return false;
    }

    return true;
}

function handleDeliveryMethodChange() {
    var
        selectedDeliveryMethod
document.querySelector('input[name="delivery_method"]:checked');

    if (selectedDeliveryMethod) {
        console.log("Вибрано
        selectedDeliveryMethod.value);
    }
}

var
    deliveryMethodRadios
document.querySelectorAll('input[name="delivery_method"]');

deliveryMethodRadios.forEach(function(radio) {
    radio.addEventListener('change', handleDeliveryMethodChange);
});

handleDeliveryMethodChange();

function showFields(deliveryMethod) {
    var
        additionalFieldsContainer
document.getElementById("additional-fields");
    additionalFieldsContainer.innerHTML = "";

    if (deliveryMethod === "in_department" || deliveryMethod ===
    "to_postamat") {
        additionalFieldsContainer.innerHTML +=
        "<label
        for='pickup_point'>Пункт самовивозу:</label>";
    }
}

```

```

        additionalFieldsContainer.innerHTML += "<input type='text'
id='pickup_point' name='pickup_point' placeholder='Адреса або номер
відділення' required><br>";
    } else if (deliveryMethod === "courier") {
        additionalFieldsContainer.innerHTML += "<label
for='street'>Вулиця:</label>";
        additionalFieldsContainer.innerHTML += "<input type='text'
id='street' name='street' placeholder='Вулиця' required><br>";
        additionalFieldsContainer.innerHTML += "<label
for='house'>Будинок:</label>";
        additionalFieldsContainer.innerHTML += "<input type='text'
id='house' name='house' placeholder='Будинок' required><br>";
        additionalFieldsContainer.innerHTML += "<label
for='apartment'>Квартира:</label>";
        additionalFieldsContainer.innerHTML += "<input type='text'
id='apartment' name='apartment' placeholder='Квартира'
required><br>";
    }
}

document.addEventListener('DOMContentLoaded', function() {
    var popup = document.getElementById('popup');
    var overlay = document.getElementById('overlay');

    popup.style.display = 'block';
    overlay.style.display = 'block';
});

function closePopup() {
    var popup = document.getElementById('popup');
    var overlay = document.getElementById('overlay');

    popup.style.display = 'none';
    overlay.style.display = 'none';
}

```