

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 веб-сайту магазину "Сімейний шопінг"

Виконав: студент курсу IV у, групи СНС-42

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

Попович В.В.  
(прізвище та ініціали)

Керівник Литвиненко Я.В.  
(прізвище та ініціали)

Нормоконтроль Шимчук Г.В.  
(прізвище та ініціали)

Завідувач кафедри Боднарчук І.О.  
(прізвище та ініціали)

Рецензент Золотий Р.З.  
(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)

« 26 » 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 Попович Валерій Валерійович  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка веб-сайту магазину "Сімейний шопінг"

Керівник роботи Литвиненко Ярослав Володимирович, д.т.н., про. професор кафедри КН  
(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 26.06.2024р.

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

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

Вступ. Розділ 1. Розгляд типових рішень та засобів для розробки веб-сайту. 1.1 Вибір технологій для розробки веб сайту інтернет магазину 1.2 Node.js з фреймворком Express 1.3 Вибір між фреймворками React.js та Angular.js 1.4 Мова розробки веб-застосунків JavaScript 1.5 Фреймворк React.js 1.6 Фреймворк Node.js 1.7 Фреймворк Express.js 1.8 Вибір системи керування базою даних 1.9 Вибір редактора для написання коду на JavaScripti . Розділ 2. Розробка веб-сайту. 2.1 Опис загальних кроків розробки веб-сайту. 2.2 Дизайн сторінок веб-сайту. 2.3 Принцип роботи сайту. 2.5 Ввід інформації в базу даних. 2.6 Висновок до другого розділу. Розділ 3. Безпека життєдіяльності, основи охорони праці. 3.1 Перелік заходів захисту від небезпечних і шкідливих чинників. 3.2 Безпечні приклади роботи та охорона праці на робочому місці. 3.3 Оцінка ризиків БЖД у життєдіяльності людини. 3.4 Висновок до третього розділу. Висновки.

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

Слайди: Тема 1. Мета та задачі 2. Структурна схема веб-сайту 3. Вигляд головної сторінки 4. Вигляд другої сторінки 5. Вигляд третьої сторінки 6. Вигляд четвертої сторінки 7. Вигляд елементів сторінок веб-сайту 8. Висновки 9.

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	к.т.н., доц. Сенчишин Віктор Степанови	12.06.2024	15.06.2024

7. Дата видачі завдання 23 січня 2023 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	30.01.2024	
2.	Підбір джерел необхідних для розробки веб-сайту	31.01.2024-03.02.2024	
3.	Опрацювання джерел по темі кваліфікаційної роботи	04.02.2024-06.02.2024	
4.	Виконання дослідження щодо розробки веб-сайту	07.02.2024-11.02.2024	
	Розроблення веб-сайту		
5.	Оформлення розділу «Розгляд типових рішень та засобів для розробки веб-сайту»	03.06.2024-05.06.2024	
6.	Оформлення розділу «Розробка веб-сайту»	06.06.2024-08.06.2024	
7.	Виконання завдання до підрозділу «Безпека життєдіяльності»	09.06.2024-11.06.2024	
8.	Виконання завдання до підрозділу «Основи охорони праці»	12.06.2024-13.06.2024	
9.	Оформлення кваліфікаційної роботи	14.06.2024-15.06.2024	
10.	Нормоконтроль	16.06.2024-19.06.2024	
11.	Перевірка на плагіат	20.06.2024	
12.	Попередній захист кваліфікаційної роботи	21.06.2024	
13.	Захист кваліфікаційної роботи	29.06.2024	

Студент

\_\_\_\_\_ (підпис)

Попович В.В.

\_\_\_\_\_ (прізвище та ініціали)

Керівник роботи

\_\_\_\_\_ (підпис)

Литвиненко Я.В.

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

Розробка веб-сайту магазину "Сімейний шопінг" // Кваліфікаційна робота освітнього рівня «Бакалавр» // Попович Валерій Валерійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНс-42 // Тернопіль, 2024 // С. , рис. – , табл. – , кресл. – , додат. – , бібліогр. – .

**Ключові слова:** web-сайт, сайт, скрипт, браузер, web-сторінка, сервер.

Кваліфікаційна робота присвячена розробці web-сайту магазину "Сімейний шопінг". У першому розділі кваліфікаційної роботи наведені технології створення web-сайтів, застосуванням технології JavaScript, СУБД MySQL. У другому розділі описано розробку веб-сайту його структури та принципів роботи.

У третьому розділі кваліфікаційної роботи описано основні питання безпеки життєдіяльності, перелік заходів захисту від небезпечних і шкідливих чинників. При цьому, описано основні вимоги безпеки при роботі оператора ПЕОМ. Окрім цього описані безпечні приклади роботи та охорона праці на робочому місці. У разі виникнення аварійних ситуацій описані кроки які необхідно зробити. Окрім цього описано оцінку ризиків БЖД у життєдіяльності людини.

**Об'єкт дослідження:** процес розробки веб-сайту.

**Предмет дослідження:** технології та методи які застосовуються при розробці веб-сайтів.

## ANNOTATION

Development of the Website of the "Family Shopping" Store // Qualification work of the educational level "Bachelor" // Popovych Valeriy Valeriyovych // Ivan Pulyuy Ternopil National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Sciences, group CHc-42 // Ternopil, 2024 // C. , fig. - , tab. - , armchair. - , add. – , bibliography -

**Keywords:** website, site, script, browser, web page, server.

The qualification work is dedicated to the development of the website of the "Family Shopping" store website. In the first section of the qualification work, the technologies for creating websites using JavaScript technology and the MySQL DBMS are given. The second chapter describes the development of the website, its structure and principles of operation.

The third section of the qualification work describes the main issues of life safety, a list of measures to protect against dangerous and harmful factors. At the same time, the main safety requirements for the work of the PC operator are described. In addition, safe examples of work and occupational health and safety at the workplace are described. In case of emergency situations, the steps to be taken are described. In addition, the assessment of the risks of BZD in human life is described.

Research object: website development process.

Research subject: technologies and methods used in the development of websites.

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

БЖД – Безпека життєдіяльності.

СУБД – Система управління базами даних.

CSS – Cascading Style Sheets

HTML – HyperText Markup Language.

## ЗМІСТ

ВСТУП .....	7
РОЗДІЛ 1. РОЗГЛЯД ТИПОВИХ РІШЕНЬ ТА ЗАСОБІВ ДЛЯ РОЗРОБКИ ВЕБ-САЙТУ .....	8
1.1 Вибір технологій для розробки веб сайту інтернет магазину .....	8
1.2 Node.js з фреймворком Express .....	9
1.3 Вибір між фреймворками React.js та Angular.js .....	11
1.4 Мова розробки веб-застосунків JavaScript.....	14
1.5 Фреймворк React.js.....	15
1.6 Фреймворк Node.js .....	17
1.7 Фреймворк Express.js .....	19
1.8 Вибір системи керування базою даних .....	20
1.9 Вибір редактора для написання коду на JavaScriptі .....	22
1.10 Висновок до першого розділу .....	23
РОЗДІЛ 2. РОЗРОБКА ВЕБ-САЙТУ .....	24
2.1 Основні кроки при розробці веб-сайту .....	24
2.2 Розробка дизайну веб-сайту. ....	28
2.3 Дизайн сторінок та кольорова гамма .....	34
2.4 Принцип роботи сайту .....	35
2.5 Адміністрування бази даних MongoDB .....	36
2.6 Тестування веб-сайту інструментом Selenium .....	38
2.7 Висновок до другого розділу .....	40
РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	41
3.1 Перелік заходів захисту від небезпечних і шкідливих чинників .....	41
3.2 Безпечні приклади роботи та охорона праці на робочому місці .....	41
3.3 Оцінка ризиків БЖД у життєдіяльності людини .....	44
3.4 Висновок до третього розділу .....	46
ВИСНОВКИ.....	47
ПЕРЕЛІК ДЖЕРЕЛ .....	48

## ДОДАТКИ



## ВСТУП

**Актуальність.** Мабуть кожен день в мережі інтернет створюються різноманітні сайти в тому числі і інтернет магазини. З'являються не тільки інтернет магазини а і більш прості веб-сайти, наприклад такі як власні веб-сайти, сайти візитки.

Сучасні технології дозволяють розробляти привабливі на вигляд веб-сайти, які швидко завантажуються, цікаву інформацію і зручні функції у керуванні.

Багато підприємств, магазинів, установ мають власний веб-сайт. Зручно на веб-сторінці розміщувати різноманітну інформацію про фірму чи магазин, представити її продукцію чи послуги, надати інформацію про акції, адреси дилерів, повідомити про новини, цікаві статті тощо. Це дає можливість зацікавити як нових так і старих клієнтів з продукцією, товаром компанії, магазину, фірми, а це дозволить збільшити продажі. Тому без сумніву створення веб-сайту інтернет-магазину є актуальним та необхідним.

**Мета і задачі дослідження.** Метою даної кваліфікаційної роботи освітнього рівня «Бакалавр» є розробка веб-сайту магазину "Сімейний шопінг". Для досягнення поставленої мети потрібно виконати ряд завдань, зокрема:

1. Провести аналіз веб-сайтів подібних до розробляемого;
2. Проаналізувати технології які будуть використовуватись для розробки веб-сайту;
3. Здійснити розробку дизайну веб-сайту;
4. Здійснити розробку структури веб-сайту;
5. Здійснити розробку необхідних елементів керування веб-сайтом;
6. Провести тестування розробленого веб-сайту.

### **Практичне значення одержаних результатів.**

Попит на товари для сім'ї є завжди, а тому їх продаж є прибутковою справою. Тому розробка веб-сайту магазину "Сімейний шопінг" має практичне значення і дозволить розширити ринок продажу товарів для сім'ї.

## РОЗДІЛ 1. РОЗГЛЯД ТИПОВИХ РІШЕНЬ ТА ЗАСОБІВ ДЛЯ РОЗРОБКИ ВЕБ-САЙТУ

### 1.1 Вибір технологій для розробки веб сайту інтернет магазину

При виборі технологій для розробки веб-сайту інтернет-магазину важливо врахувати декілька ключових аспектів, таких як масштабованість, безпека, швидкодія та зручність для адміністрування. Розглянемо найбільш популярні і ефективні технології, які можна використати:

**CMS (Content Management System):**

**Magento:** Magento є потужною платформою для електронної комерції, спеціалізується на масштабованій та гнучкій розробці магазинів. Вона підтримує багатofункціональність, включаючи управління каталогами, замовленнями і платежами.

**WooCommerce (з платформи WordPress):** Це потужний плагін для WordPress, який перетворює ваш сайт на повнофункціональний інтернет-магазин. Він має широкий вибір розширень і тем, що дозволяє налаштовувати магазин під ваші потреби.

**Custom Development (Власний розробка):**

**Ruby on Rails:** Це фреймворк для швидкої розробки веб-додатків, який підходить для створення ефективних і масштабованих інтернет-магазинів.

**Node.js з фреймворком Express:** Цей стек підходить для створення швидких і високопродуктивних додатків, що можуть бути ідеальним вибором для інтернет-магазину з великою кількістю взаємодій і real-time оновленнями.

**Cloud Solutions:**

**Shopify:** Це хмарна платформа для електронної комерції, яка надає готові рішення для створення, запуску і управління магазином без потреби великої ІТ інфраструктури.

BigCommerce: Інша хмарна платформа для електронної комерції, яка пропонує широкий вибір функцій для створення і управління магазином.

При виборі технології слід враховувати потреби бізнесу, технічні навички команди розробників, бюджет, масштабність проекту та інші специфічні вимоги. Кожна з перерахованих платформ має свої переваги і недоліки, тому важливо детально розглянути кожен з них перед прийняттям рішення.

## 1.2 Node.js з фреймворком Express

Node.js з фреймворком Express є дуже популярним вибором для розробки веб-додатків, включаючи інтернет-магазини. Ось деякі переваги цього підходу:

**Ефективність та продуктивність:** Node.js відомий своєю високою продуктивністю та можливістю обробки великого обсягу одночасних підключень. В поєднанні з Express, цей стек дозволяє створювати швидкі і високопродуктивні додатки.

**Простота і прозорість:** Express є легким та гнучким фреймворком, який дозволяє розробникам швидко створювати маршрути, обробники запитів, middleware і багато іншого. Він не накладає обмежень на архітектуру вашого додатку, що дозволяє створювати його згідно з ваших унікальних потреб.

**Широкий вибір бібліотек і модулів:** Величезне співтовариство Node.js розробників означає наявність безлічі модулів і пакетів, які можна використовувати в вашому проекті. Ви зможете знайти готові рішення для різних аспектів вашого інтернет-магазину, таких як обробка платежів, автентифікація користувачів, управління сесіями і багато іншого.

**Real-time оновлення:** За допомогою Node.js і Express можна легко створювати додатки, які підтримують real-time оновлення, наприклад, відображення наживо змін у наявності товарів або змін у кошику користувача без перезавантаження сторінки.

**Скалірування і гнучкість:** Node.js дозволяє створювати масштабовані додатки, які можуть легко масштабуватися з ростом вашого бізнесу. Express

надає достатньо гнучкий фреймворк для того, щоб ваші додатки могли ефективно адаптуватися до змін у вимогах і розмірах.

Хоча Node.js і Express є потужними і популярними інструментами, важливо врахувати, що їх використання вимагає розуміння JavaScript і серверних технологій. Якщо ваша команда розробників має такі навички, цей стек може бути відмінним вибором для створення вашого інтернет-магазину.

Node.js є дуже потужним інструментом для розробки веб-додатків, і він має кілька особливостей, які роблять його популярним серед розробників. Ось деякі ключові особливості використання Node.js для веб:

**Асинхронність і подійне програмування:** Основна концепція Node.js полягає в тому, що він є асинхронним і подійно орієнтованим. Це означає, що він може ефективно обробляти велику кількість одночасних запитів без блокування потоків. Це особливо важливо для веб-додатків, які потребують real-time оновлень і обробки багато запитів одночасно.

**Швидкодія:** Node.js побудований на движку JavaScript V8 від Google, який відомий своєю високою продуктивністю. Він використовує неблокуючий ввід/вивід і оптимізовану обробку даних, що дозволяє створювати швидкі веб-додатки.

**Скалірування:** Node.js дозволяє легко масштабувати додатки, оскільки він підтримує горизонтальне масштабування, тобто можливість запускати багато процесів Node.js на багатьох серверах.

**Велике співтовариство і екосистема модулів:** Node.js має активне співтовариство розробників і широку бібліотеку модулів, що полегшує розробку веб-додатків. Ви зможете знайти рішення для практично будь-якої задачі, починаючи від роботи з базами даних до інтеграції з іншими сервісами.

**JavaScript на фронтенді і бекенді:** Node.js дозволяє використовувати JavaScript як на стороні сервера, так і на стороні клієнта. Це спрощує розробку і підтримку коду, оскільки команда розробників може працювати з однією мовою програмування для всього стеку додатку.

Розширюваність і гнучкість: Node.js дозволяє розробникам використовувати різні фреймворки і бібліотеки в залежності від конкретних потреб проекту. Фреймворки, такі як Express.js, Nest.js або Кoa, надають різні рівні абстракції для розробки веб-додатків, що дозволяє вам обрати той, який найкраще підходить для вашого випадку.

Враховуючи ці особливості, Node.js є відмінним вибором для створення веб-додатків, включаючи інтернет-магазини, особливо якщо потрібна швидкість, масштабованість і можливість real-time оновлень.

### **1.3 Вибір між фреймворками React.js та Angular.js**

Вибір між React.js і Angular.js для розробки інтернет-магазину залежить від кількох факторів, таких як досвід роботи з ними, потреби проекту, масштаб і складність додатку. Розглянемо основні відмінності і переваги кожної з них:

React.js

Переваги:

Гнучкість і простота: React є декларативною бібліотекою, яка дозволяє розробникам швидко створювати інтерфейси. Вона не накладає обмежень на архітектуру проекту і дозволяє використовувати різні інструменти і бібліотеки (наприклад, Redux для керування станом).

Велике співтовариство і екосистема: React має широке співтовариство розробників і багату екосистему, що включає тисячі сторонніх бібліотек і компонентів. Це спрощує розробку і дозволяє використовувати готові рішення.

Virtual DOM: React використовує віртуальний DOM для ефективного відображення змін у інтерфейсі, що робить його швидшим і дозволяє підтримувати багато елементів на сторінці.

JSX: JSX дозволяє розробникам писати HTML-подібний код прямо в JavaScript, що полегшує розробку і підтримку коду.

React.js є однією з найпопулярніших бібліотек JavaScript для розробки інтерфейсів користувача, і вона має свої власні переваги і недоліки.

Переваги React.js:

Віртуальний DOM: Однією з ключових особливостей React є використання віртуального DOM. Це дозволяє ефективно оновлювати тільки ті частини сторінки, які змінилися, що забезпечує високу продуктивність додатків.

Компонентна архітектура: React сприяє створенню додатків через компоненти, що полегшує розбиття інтерфейсу на невеликі, незалежні частини. Це сприяє покращенню перевикористання коду і підтримки додатків.

Декларативний підхід: React використовує декларативний підхід для опису взаємодії інтерфейсу, що дозволяє зосередитися на тому, що потрібно відобразити, а не на тому, як саме це зробити.

Розширюваність і велике співтовариство: React має велике і активне співтовариство розробників, що призводить до широкого вибору сторонніх бібліотек, компонентів і інструментів для підтримки розробки.

Одноразові компоненти: Використання одноразових компонентів у React дозволяє зручно керувати станом і взаємодією з користувачем, що спрощує тестування і підтримку коду.

Недоліки React.js:

Висока крутизна кривої вивчення: Для новачків може виявитися складним засвоєння концепцій, таких як JSX, віртуальний DOM і стан компонентів.

Необхідність додаткових бібліотек: В React часто використовуються додаткові бібліотеки, такі як Redux для управління станом додатків, що може вимагати додаткового навчання і налаштування.

Неодноразовість рішень: Завдяки великій кількості доступних рішень і бібліотек, іноді важко визначити, яке саме рішення використовувати для конкретної задачі, що може вплинути на єдність кодової бази.

Інтеграція з іншими технологіями: Інтеграція React з іншими технологіями (наприклад, серверні технології або бібліотеки для рендерінгу на сервері) може потребувати додаткових уточнень і зусиль.

Загалом, React.js є потужним інструментом для створення модерних інтерфейсів користувача, особливо для односторінкових додатків та інтерфейсів, які часто оновлюються. Однак вибір між React.js і іншими технологіями повинен залежати від конкретних потреб вашого проекту і рівня зручності для вашої команди розробників.

AngularJS, якій в даний час застосовують версії 2+, це один з популярних фреймворків JavaScript, що використовується для розробки односторінкових застосунків (SPA) і веб-додатків. Ось деякі його переваги та недоліки:

**Переваги AngularJS:**

**Структурований підхід і модульність:** AngularJS пропонує чітко структуровану архітектуру і модульність, що сприяє легшому управлінню складністю додатків.

**Двостороннє зв'язування (Two-way data binding):** Одна з ключових особливостей AngularJS - це двостороннє зв'язування даних між моделлю і представленням, що дозволяє автоматично оновлювати дані в обох напрямках без необхідності вручного втручання.

**Тестирування:** AngularJS вбудовує засоби для тестування, такі як Dependency Injection (DI) і вбудовані інструменти для їх модульного тестування, що полегшує розробку тестів для додатків.

**Масштабованість:** Фреймворк добре підходить для великих і складних додатків, де важливо зберігати структурованість і підтримувати чітку організацію коду.

**Офіційна підтримка:** AngularJS має офіційну підтримку від Google, що гарантує актуальність і підтримку платформи.

**Недоліки AngularJS:**

**Складність вивчення:** AngularJS може виявитися складним для вивчення і розуміння, особливо для новачків. Він вимагає розуміння концепцій, таких як Dependency Injection і декларативний стиль програмування.

**Великі ресурсовмісткість:** Деякі відгуки вказують на те, що AngularJS може використовувати більше ресурсів порівняно з іншими фреймворками, що може впливати на продуктивність додатків, особливо на мобільних пристроях.

**Низька сумісність зі старим кодом:** Перехід на нові версії AngularJS може вимагати значних зусиль і часу через великі зміни в API та підходах.

**Великі розміри:** Деякі розробники вказують на те, що AngularJS має великі розміри, що може вплинути на час завантаження сторінок і загальну продуктивність додатків.

Отже, вибір між AngularJS і React.js залежить від команди, потреб проекту і рівня зручності в роботі з кожним інструментом. Обидва фреймворки мають свої унікальні переваги і недоліки, і вибір повинен враховувати конкретні умови вашого проекту. Оскільки ми знайомі з React.js тому виберемо його для розробки фронт-енду веб-сайту.

## **1.4 Мова розробки веб-застосунків JavaScript**

JavaScript є однією з найпопулярніших мов програмування для розробки веб-застосунків, існує кілька причин, чому вона настільки популярна серед веб-розробників:

**Універсальність:** JavaScript підтримується всіма сучасними браузерами, що робить його ідеальним вибором для розробки клієнтської частини веб-застосунків.

**Простота вивчення:** JavaScript має досить простий синтаксис, що дозволяє швидко освоїти основні концепції і почати писати код.

**Широкі можливості:** JavaScript дозволяє створювати динамічні інтерфейси, реалізовувати асинхронні запити до сервера, взаємодіяти з DOM (Document Object Model) і багато іншого.

**Екосистема і бібліотеки:** JavaScript має велику кількість бібліотек і фреймворків, які значно спрощують розробку. Наприклад, React, Angular, Vue.js для фронтенду і Node.js для серверної частини.



**Швидкість розробки:** JavaScript дозволяє швидко реалізувати ідеї і прототипи, а також швидко вносити зміни і вдосконалювати функціональність веб-застосунків.

**Підтримка від спільноти:** JavaScript має велике співтовариство розробників, що сприяє активному обміну досвідом і розвитку інструментів.

Що стосується недоліків, JavaScript має певні аспекти, які можуть бути викликані складністю інтерпретації деяких його концепцій новачками, а також потенційними проблемами безпеки через можливість виконання небезпечних скриптів на клієнтській стороні. Однак з введенням сучасних стандартів JavaScript і удосконаленнями браузерних двигунів ці проблеми значно зменшуються.

## **1.5 Фреймворк React.js**

React.js є дуже популярним вибором для розробки інтернет-магазинів через його здатність створювати динамічні інтерфейси та простоту управління станом додатку. Розглянемо ключові аспекти використання React.js для цієї цілі:

**Особливості використання React.js для інтернет-магазину:**

**Компонентна архітектура:** React дозволяє розбити інтерфейс магазину на малі компоненти, які можна легко перевикористовувати та модулізувати. Наприклад, компоненти для продуктів, кошика, фільтрів, і так далі.

**Віртуальний DOM і швидкодія:** Використання віртуального DOM у React забезпечує ефективне оновлення інтерфейсу при зміні стану, що особливо важливо для динамічних магазинів зі значною кількістю товарів і фільтрів.

**Управління станом з Redux або Context API:** Для складних магазинів важливо ефективно керувати станом додатку. Redux або Context API у React дозволяють це зробити, зберігаючи стан додатку в одному місці і спрощуючи його маніпулювання.

**Адаптивність і респонсивний дизайн:** React легко інтегрується з бібліотеками для створення респонсивних інтерфейсів, такими як Bootstrap або

Material-UI, що дозволяє забезпечити користувачам однаково зручний доступ незалежно від пристрою.

SEO: Важливим аспектом інтернет-магазинів є SEO. React має підтримку SSR (Server-Side Rendering) через інструменти, такі як Next.js або Gatsby, що дозволяє забезпечити швидке завантаження інтерфейсу і оптимальний індексування пошуковими системами.

Широка підтримка і спільнота: React має велику спільноту розробників, що підтримує його розвиток і надає багато готових рішень і компонентів для розробки магазинів.

Переваги використання React.js для інтернет-магазину:

Швидкість розробки: Реактивна природа React дозволяє швидко відображати зміни в інтерфейсі і вносити корективи, що особливо важливо для магазинів, де потрібно часто змінювати і додавати функціональність.

Висока продуктивність: Завдяки віртуальному DOM і оптимізаціям React забезпечує високу продуктивність магазинів, навіть при великій кількості товарів і користувачів.

Масштабованість: React легко масштабується, що дозволяє розширювати магазин з ростом бізнесу і додавати нові функції без значних зусиль.

Недоліки використання React.js для інтернет-магазину:

Складність налаштування SSR: Хоча SSR може покращити SEO і продуктивність, його налаштування може вимагати додаткового часу і ресурсів.

Велика кількість вибору і складність вибору технологій: Вибір інших технологій для додаткової функціональності (наприклад, управління станом або роутінг) може бути складним через велику кількість альтернатив.

Узагальнюючи, React.js є потужним інструментом для розробки інтернет-магазинів, що дозволяє створювати сучасні, швидкі і масштабовані веб-додатки з високим рівнем користувацької взаємодії і продуктивності.

## 1.6 Фреймворк Node.js

Фреймворк Node.js сам по собі не є інструментом для розробки веб-додатків чи інтернет-магазинів, але він є платформою для виконання JavaScript на серверному боці. Для створення інтернет-магазину на Node.js зазвичай використовуються різні фреймворки, бібліотеки та інструменти, які забезпечують веб-сервер, маршрутизацію, управління базами даних та інші необхідні функції. Розглянемо деякі з них:

Фреймворки для розробки інтернет-магазину на Node.js:

**Express.js:** Це мінімалістичний і гнучкий веб-фреймворк для Node.js, який надає базові інструменти для створення веб-додатків. Він підтримує швидку реалізацію HTTP-маршрутів, middleware для обробки запитів і шаблонізацію для генерації HTML на серверному боці.

**NestJS:** Це прогресивний Node.js фреймворк, який будується на основі Express.js і забезпечує розширені можливості для створення масштабованих і структурованих додатків. NestJS пропонує вбудовану підтримку для TypeScript, Dependency Injection, модularity і інших сучасних підходів.

**Koa.js:** Це легкий фреймворк, створений тими ж авторами, що і Express.js, але з більш сучасною архітектурою, що використовує асинхронні функції (async/await) для керування middleware і обробки запитів.

Бібліотеки та інструменти для розробки інтернет-магазину на Node.js:

**MongoDB або MySQL:** Для зберігання даних вашого магазину ви можете використовувати реляційну базу даних, таку як MySQL, або нереляційну базу, наприклад, MongoDB. Для взаємодії з базою даних на Node.js використовуються ORM (наприклад, Sequelize для MySQL) або ODM (наприклад, Mongoose для MongoDB).

**React або Vue.js для фронтенду:** Для створення динамічного інтерфейсу вашого магазину ви можете використовувати фронтенд бібліотеки, такі як React або Vue.js. Ці бібліотеки забезпечують можливості для розробки SPA (односторінкових застосунків), які добре підходять для інтернет-магазинів.

Stripe або PayPal SDK: Для інтеграції з платіжними системами ви можете використовувати SDK Stripe або PayPal для обробки платежів в вашому магазині.

Шаблонізатори (наприклад, Pug або EJS): Для генерації HTML на серверному боці ви можете використовувати шаблонізатори, такі як Pug або EJS, що дозволяють зручно створювати розмітку інтерфейсу.

Переваги використання Node.js для розробки інтернет-магазину:

Швидкість і продуктивність: Node.js забезпечує високу швидкість роботи завдяки асинхронному не блокуючому вводу/виводу (non-blocking I/O), що особливо важливо для веб-додатків з великою кількістю одночасних запитів.

Широкі можливості розширення: З використанням багатьох модулів і бібліотек Node.js ви можете легко розширювати функціональність вашого магазину, додавати нові функції і оптимізувати веб-додаток.

Спільнота та підтримка: Node.js має велику активну спільноту розробників, яка підтримує розвиток і підтримку інструментів і бібліотек для розробки веб-додатків.

Недоліки використання Node.js для розробки інтернет-магазину:

Складність при великих проектах: Для великих і складних проектів може знадобитися детальне планування і оптимізація коду, оскільки Node.js може бути вимогливим до ресурсів у випадку неоптимізованого використання.

Управління асинхронним кодом: Для розробки в Node.js важливо мати глибоке розуміння асинхронного програмування і керування промісами або callback'ами.

Усі ці аспекти потрібно враховувати при виборі Node.js для розробки інтернет-магазину, проте він є потужним інструментом з великим потенціалом для реалізації сучасних веб-додатків і онлайн-магазинів.

## 1.7 Фреймворк Express.js

Express.js є дуже популярним фреймворком для розробки веб-додатків на платформі Node.js. Розглянемо деякі ключові особливості і переваги використання Express.js:

Основні особливості Express.js:

**Мінімалістичність і гнучкість:** Express.js є досить легким і мінімалістичним фреймворком, що дозволяє розробникам будувати веб-додатки і API з мінімальним набором обов'язкових функцій. Він надає лише основні інструменти для маршрутизації і управління HTTP-запитами.

**Маршрутизація:** Express надає потужний механізм для визначення маршрутів (routes), що дозволяє вам легко визначати, як ваш додаток має реагувати на різні типи запитів (GET, POST, PUT, DELETE і т.д.) на різних URL-шляхах.

**Middleware:** Middleware в Express є ключовою концепцією, яка дозволяє обробляти запити перед тим, як вони досягнуть функцій маршрутизації. Це дозволяє виконувати різноманітні дії, такі як логування, перевірка автентичності, обробка помилок і багато іншого.

**Шаблонізація:** Express має підтримку різних шаблонізаторів (наприклад, Pug, EJS), що дозволяє розробникам генерувати HTML-сторінки на серверному боці за допомогою шаблонів.

**Розширюваність:** Існують тисячі сторонніх модулів, які можна використовувати з Express для різноманітних задач, від обробки форм до інтеграції з базами даних і сторонніми сервісами.

**Підтримка HTTP-запитів і відповідей:** Express дозволяє легко створювати і керувати HTTP-запитами і відповідями, включаючи роботу з параметрами запити, заголовками і тілом запити.

Переваги використання Express.js:

**Простота використання і швидкість розробки:** Express дозволяє швидко створювати потужні веб-додатки з мінімальними зусиллями.

Широка підтримка і активна спільнота: Велика спільнота розробників і наявність багатьох ресурсів, документації та плагінів спрощують розробку і підтримку додатків.

Ідеальний для створення API: Express добре підходить для створення API, оскільки дозволяє швидко визначати різні маршрути і обробляти запити.

Недоліки Express.js:

Більше ручної роботи: Оскільки Express має мінімалістичну природу, для певних задач (наприклад, автентифікація, контроль доступу) може знадобитися використання додаткових бібліотек або самостійних рішень.

Немає вбудованої підтримки для деяких функціональних можливостей: Наприклад, для шаблонізації потрібно вибрати окремий шаблонізатор і підключити його.

У цілому, Express.js є потужним інструментом для розробки веб-додатків і API на Node.js, що дозволяє швидко створювати ефективні і масштабовані рішення.

## **1.8 Вибір системи керування базою даних**

Вибір між SQL базою даних (такими як MySQL, PostgreSQL) і MongoDB (NoSQL база даних) залежить від конкретних потреб проекту та особливостей даних, з якими ми працюємо. Розглянемо ключові аспекти для порівняння і вибору між ними:

SQL база даних

Переваги:

Структурованість даних: SQL бази даних мають чітку структуру, що визначається схемою, що полегшує зберігання і управління даними. Це особливо важливо для додатків зі складною структурою даних, де потрібна консистентність та інтеграція даних.

Транзакційна підтримка: Більшість SQL баз даних підтримують транзакції, що дозволяє вам гарантувати атомарність, цілісність, ізоляцію і стійкість даних (ACID властивості).

Запити із складними з'єднаннями: SQL дозволяє виконувати складні запити з внутрішніми з'єднаннями, підзапитами, групуванням і агрегаціями, що забезпечує потужність і гнучкість для обробки даних.

Велика спільнота і екосистема: SQL бази даних мають велику спільноту розробників і багато інструментів для розробки, адміністрування і моніторингу.

Недоліки:

Менша гнучкість схеми даних: Схема даних у SQL базах фіксована і заздалегідь визначена, що може ускладнювати зміни в структурі даних під час розвитку проекту.

Шкальність: При великих обсягах даних SQL бази даних можуть потребувати більш складної настройки для забезпечення високої шкальності, особливо в контексті горизонтального масштабування.

MongoDB (NoSQL база даних)

Переваги:

Гнучка схема даних: MongoDB використовує гнучку схему документів (JSON-подібні об'єкти), що дозволяє зберігати дані різної структури без фіксованої схеми.

Простота масштабування: MongoDB добре підходить для горизонтального масштабування, що дозволяє легко розширювати базу даних за потреби вашого проекту.

Висока швидкість для операцій з документами: MongoDB забезпечує швидкі операції збереження, читання і оновлення документів за рахунок використання внутрішнього представлення даних.

Масштабність: MongoDB дозволяє легко пристосовуватися до зростання обсягів даних і потоків роботи.

Недоліки:

Відсутність транзакцій: MongoDB не підтримує традиційні ACID транзакції (атомарність, цілісність, ізоляція, стійкість), що може вимагати особливих рішень при проектуванні бази даних.

Менша підтримка і інструменти порівняно з SQL: В порівнянні з SQL базами, MongoDB має меншу кількість інструментів і меншу спільноту, хоча все ще має широке використання та підтримку.

Вибір системи керування базою даних

Вибір SQL бази даних рекомендується, якщо ваші дані мають структурований характер і вам важлива транзакційна підтримка, складні запити та велика екосистема інструментів.

Вибір MongoDB виправданий, якщо у нас гнучкі вимоги до структури даних, ми плануємо швидко масштабувати систему і нам потрібна висока швидкість операцій з документами тому саме MongoDB.

## **1.9 Вибір редактора для написання коду на JavaScripti**

Вибір редактора для написання коду на JavaScript може значно вплинути на ваш комфорт і продуктивність під час розробки. Ось декілька популярних варіантів редакторів, які зазвичай використовують для роботи з JavaScript:

### **1. Visual Studio Code (VS Code)**

Visual Studio Code є одним з найпопулярніших та потужних редакторів для розробки на JavaScript. Він має широкий вибір розширень і плагінів, що дозволяють налаштовувати його під ваші потреби. VS Code підтримує автодоповнення, налагодження коду, інтеграцію з Git та багато іншого.

### **2. Sublime Text**

Sublime Text відомий своєю швидкістю роботи і легкістю використання. Він має багато корисних функцій, таких як швидкі клавішні комбінації, розширюваність за допомогою плагінів і підтримку різних мов програмування, включаючи JavaScript.

### **3. Atom**



Atom є редактором, створеним GitHub, і він також має широкий вибір розширень і тем для налаштування відповідно до ваших уподобань. Він легкий і простий у використанні, з можливістю налаштування під ваші потреби.

#### 4. WebStorm

WebStorm від JetBrains є інтегрованою середовищем розробки (IDE), спеціально призначеною для роботи з JavaScript і іншими мовами, що запускаються на платформі Node.js. Він має розширену підтримку для фронтенду та серверного JavaScript, включаючи автоматичне доповнення, налагодження, інтеграцію з системами контролю версій та багато іншого.

#### 5. Brackets

Brackets від Adobe є відкритим і легким редактором для веб-розробників. Він має інтегровані інструменти для розробки веб-сайтів і підтримує CSS, HTML та JavaScript. Він також має можливість налаштувати за допомогою розширень.

Вибір редактора залежить від ваших індивідуальних потреб, зручності використання і особистих уподобань. Більшість з цих редакторів безкоштовні або мають безкоштовні версії, тому ви можете спробувати кілька варіантів і обрати той, який найбільше вам підходить.

### **1.10 Висновок до першого розділу**

В першому розділі кваліфікаційної роботи було оглянуто основні веб-технології які використовуються для створення веб-сайтів мовою JavaScript. Розглянуті фреймворки для написання фронт-енда та бек-енда, розглянуто та вибрано СКБД MongoDB. А також популярні редактори для написання веб-сайту на мові JavaScript.

## РОЗДІЛ 2. РОЗРОБКА ВЕБ-САЙТУ

### 2.1 Основні кроки при розробці веб-сайту

Вибір таких технологій, як React, Node.js з Express, та MongoDB, є досить популярним для розробки сучасних веб-додатків, зокрема інтернет-магазинів. Розглянемо такі, які можна виконати для початку розробки:

#### 1. Встановлення Node.js і MongoDB

Встановіть Node.js з офіційного веб-сайту (<https://nodejs.org/>). Node.js дозволить вам використовувати JavaScript на серверному боці з Express.

Встановіть MongoDB з офіційного веб-сайту (<https://www.mongodb.com/>). MongoDB буде вашою базою даних для зберігання даних про товари, замовлення і користувачів.

#### 2. Створення проекту

Використовуйте create-react-app для швидкого створення початкового шаблону вашого React додатку:

```
bash
```

Копіювати код

```
npm create-react-app my-online-store
```

```
cd my-online-store
```

#### 3. Налаштування серверної частини з Express.js

Ініціалізуйте ваш Node.js проект, створивши package.json:

```
bash
```

Копіювати код

```
npm init -y
```

Встановіть Express.js і необхідні модулі:

```
bash
```

Копіювати код

```
npm install express mongoose body-parser cors
```

Створіть основний файл сервера, наприклад `server.js` і підключіться до бази даних MongoDB за допомогою Mongoose:

```
javascript
```

```
Копіювати код
```

```
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const cors = require('cors');

const app = express();
const port = process.env.PORT || 5000;

// Підключення до MongoDB
mongoose.connect('mongodb://localhost:27017/my-online-store', {
  useNewUrlParser: true,
  useUnifiedTopology: true
})
.then(() => console.log('MongoDB Connected'))
.catch(err => console.log(err));

// Middlewares
app.use(cors());
app.use(bodyParser.json());

// Routes
app.use('/api/products', require('./routes/products')); // Приклад маршруту для
продуктів

// Запуск сервера
app.listen(port, () => {
```

```
console.log(`Server is running on port ${port}`);
});
```

#### 4. Створення моделей і маршрутів для MongoDB

Створіть папку `models` для опису схем даних MongoDB, наприклад, модель `Product`:

javascript

Копіювати код

```
const mongoose = require('mongoose');

const ProductSchema = new mongoose.Schema({
  name: String,
  description: String,
  price: Number,
  category: String,
  // Додайте інші поля, які вам потрібні
});
```

```
module.exports = mongoose.model('Product', ProductSchema);
```

Створіть папку `routes` для маршрутів Express, наприклад, `products.js`:

javascript

Копіювати код

```
const express = require('express');
const router = express.Router();
const Product = require('../models/Product');

// Отримати всі продукти
router.get('/', async (req, res) => {
  try {
    const products = await Product.find();
    res.json(products);
  }
});
```

```

    } catch (err) {
      res.status(500).json({ message: err.message });
    }
  });

  // Створити новий продукт
  router.post('/', async (req, res) => {
    const product = new Product({
      name: req.body.name,
      description: req.body.description,
      price: req.body.price,
      category: req.body.category
      // Додайте інші поля
    });

    try {
      const newProduct = await product.save();
      res.status(201).json(newProduct);
    } catch (err) {
      res.status(400).json({ message: err.message });
    }
  });

  module.exports = router;

```

## 5. Розробка фронтенду з React.js

Використовуйте React для створення компонентів інтерфейсу користувача, таких як сторінки продуктів, кошика покупок, сторінки оформлення замовлення тощо.

Використовуйте Axios або Fetch для взаємодії з API вашого Node.js сервера для отримання і збереження даних в MongoDB.

Це базовий набір кроків для початку розробки веб-сайту інтернет-магазину на JavaScript з використанням React, Node.js з Express і MongoDB. За необхідності можна додавати інші функції, такі як автентифікація користувачів, інтеграція з платіжними системами, та інше.

## 2.2 Розробка дизайну веб-сайту.

Результатом кваліфікаційної роботи є web-сайт магазину «Сімейний шопінг». Даний сайт складається з таких сторінок: головна сторінка, опалата і доставка, блог, про нас, відгуки, співпраця, вакансії, контакти, угода користувача, обмін та повернення, умови використання сайту, каталог продукції. Дана структура подана на рисунку 2.1

В каталозі продукції є такі категорії «Жінкам», «Чоловікам», «Дівчатам», «Хлопцям», «Немовлятам», «Товари для дому», «Іграшки», «Розвиток та творчість», «Спорт».

Сторінка «Головна» призначена для того, щоб коротко ознайомити користувача з особливостями роботи магазину. На головній сторінці міститься інформація про графік роботи магазину (його контакт центру). Наведений головний телефон для спілкування з менеджерами з продажу. У вигляді піктограм «галочка», «сердечко» розташованих у верхньому кутку подані елементи для порівняння товарів і «додати товар у список бажань» відповідно. На головній сторінці сайту розташовується інформація про новини, акції та цікаву інформацію, у вигляді десяти слайдів які прокручуються по колу, зациклено. При натисканні на такий слайд клієнт перейде на інформацію на сайті яка стосується вибраного слайду. Це може бути список вжуття на яке в даний період діють знижки чи інша інформація, яка може бути корисна для відвідувачів сайту.

На сайті також розміщені інші елементи керування такі як вибір мови. Тут можна обрати українську чи англійську мову. «Вхід» дозволяє авторизуватись зареєстрованим користувачам. Якщо відвідувач сайту не зареєстрований йому

буде запропоновано зареєструватись. Важливим елементом будьякого інтернет магазину є «Мій кошик» через який клієнти можуть придбати вибрану в інтернет-магазині продукцію.

На головній сторінці у вигляді піктограм розташованих унизу інформаційних слайдів наведені категорії товарів.

Логотип сайту «Сімейний шопінг» розташований зліва зверху. За бажанням замовника даного проєкту він може бути легко перемальований за потреби.

У верхньому лівому кутку передбачено пошук по сайту. Оскільки даний магазин містить дуже багато товарів. То природнім є можливість здійснювати пошук товарів по сайту.

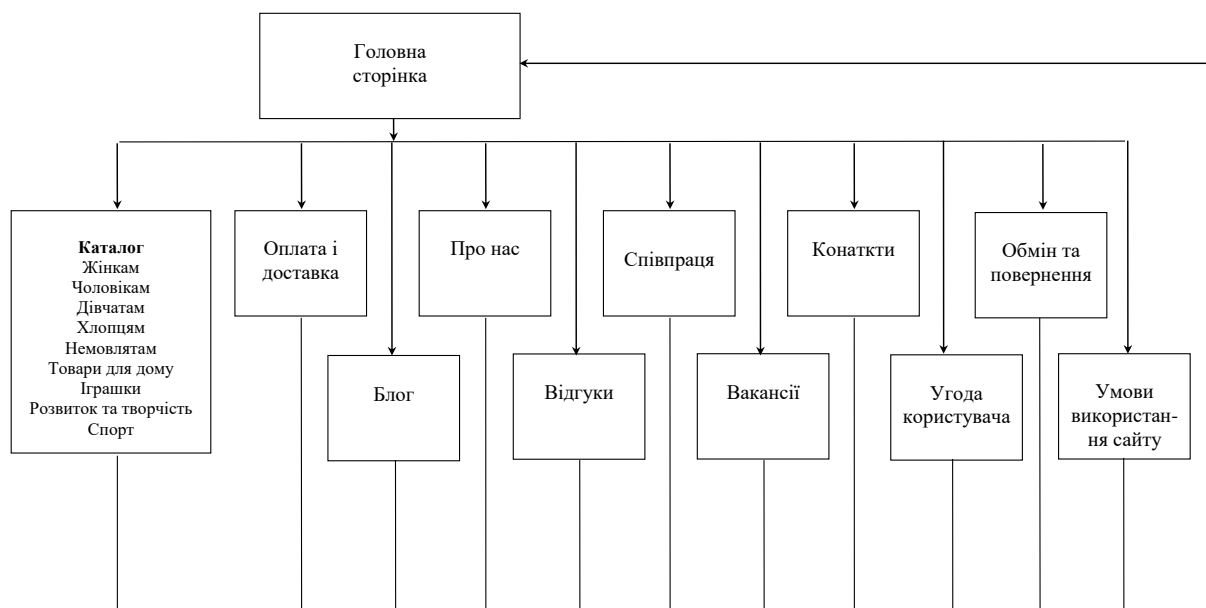


Рисунок 2.1 – Структура розробленого сайту

Щодо структури самих сторінок (розміщення окремих елементів), то вони розміщеня на головній (домашній) сторінці. Вигляд головної сторінки зображено на рисунку 2.2.



Рисунок 2.2 – Вигляд головної сторінки розробленого сайту

Сторінка «Каталог» з вибраною конкретною категорією продукції подана на рсунку 2.3. Вона містить наступну інформаціона прикладі «Костюмів для хлопчиків». На даній сторінці вивожитьмя по 6 товарів. Хоча є можливість налаштувати фільтр який буде видавати значно більше продукції на сторінці. На рисунках які відповідають товару подані знижки, окрім цього вказаний артикул товару та дана коротка характеристика (опис) товару.

На цій сторінці зліва наведений фільтр який дозволяє уточнювати пошук продукції в обраній категорії, зокрема тут можна обрати ціну вказавши діапазон, можна обрати українські розміри які відповідають продукції європейського чи азіатського ринку. Можна обрати одяг за сезоном чи бренд.



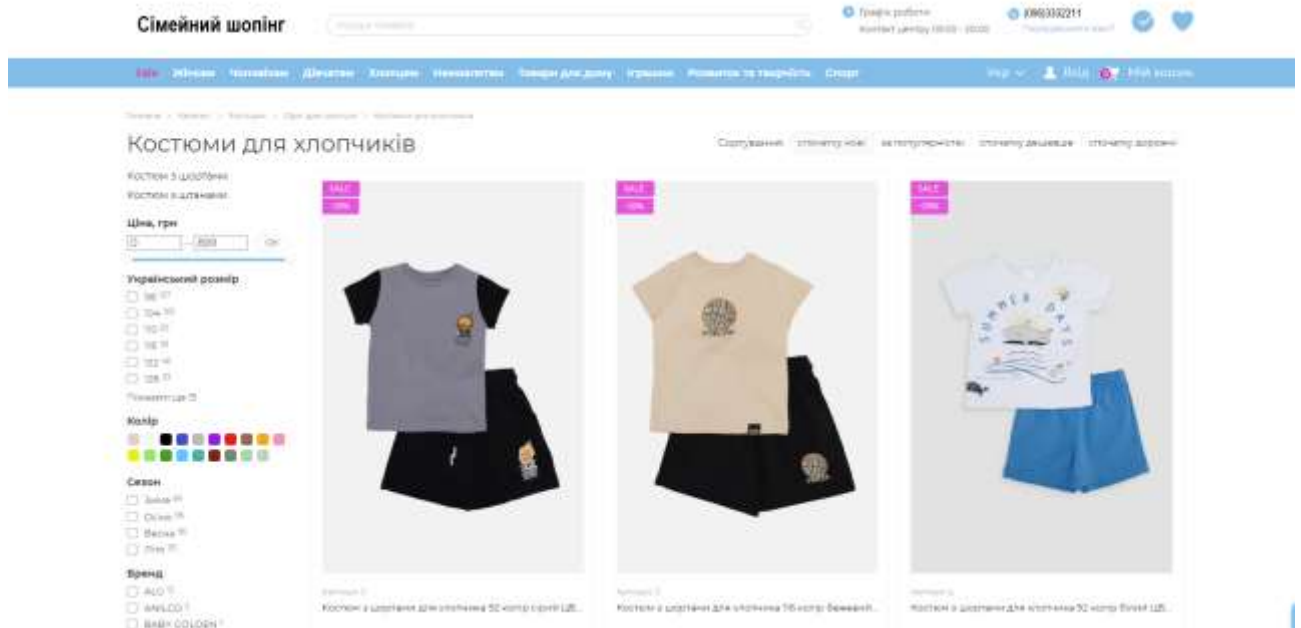


Рисунок 2.3 – Вигляд сторінки «каталог» інтернет-магазину

Така саме картина відображається при виборі іншої категорії товарів в даному інтернет магазині приклад наведено на рисунку 2.4.

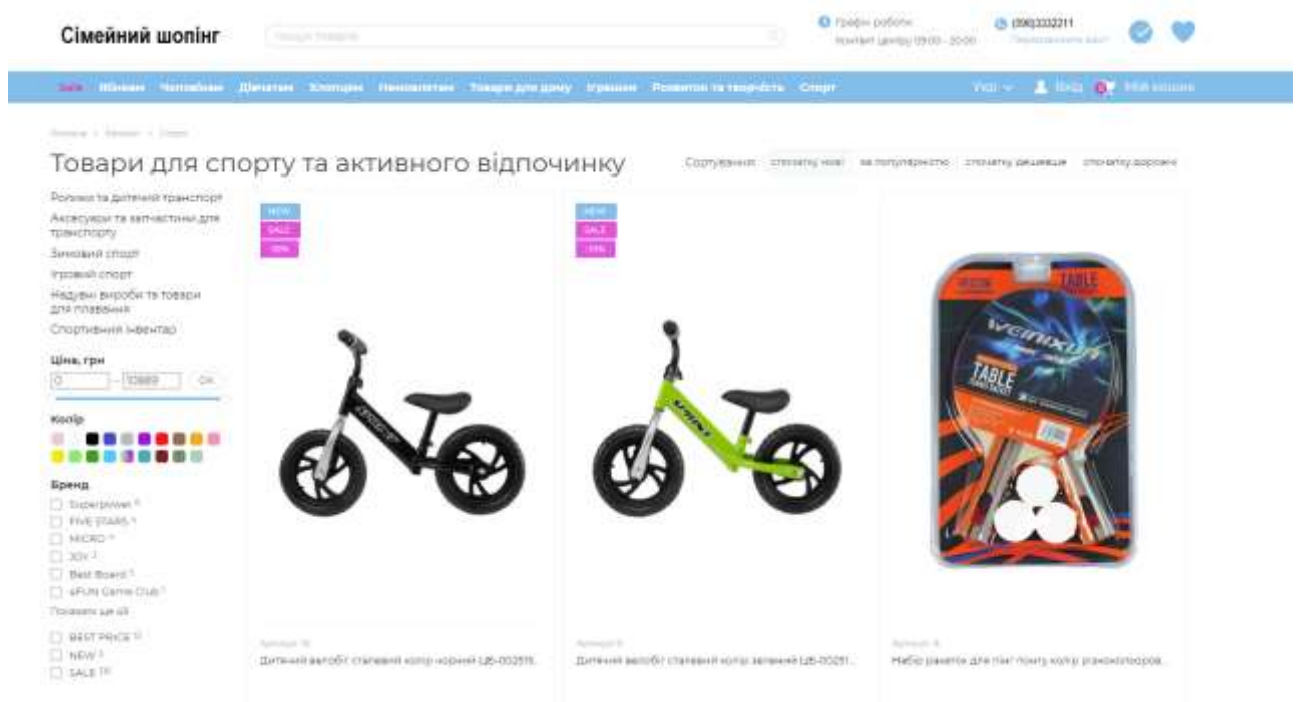


Рисунок 2.4 – Вигляд сторінки «каталог» інтернет-магазину

Хоч на рисунку наведені інші види продукції з іншої категорії проте тут так само присутні типові елементи керування такі як фільтр, інформація про

опис товару артикул та інше.

Розглянемо інші сторінки веб-сайту інтернет-магазину.

На сторінці «Про нас» подана інформація про магазин «Сімейний шопінг», зокрема тут дано наступне: скільки на ринку, коли заснований, продукцію яких брендів продає магазин та інше.

Відгуки задоволених клієнтів магазину подані на окремій сторінці «Відгуки».

На сторінці «Блог» наведена інформація про події, новини які пов'язані з магазином «Сімейний шопінг».

На сторінці «Співпраця» описана інформація для потенційних зацікавлених магазинів які зможуть продавати продукцію магазину «Сімейний шопінг».

На сторінці «Обмін та повернення» вказана інформація про можливі способи повернення товару для прикладу ця сторінка винесена окремо і подана на рисунку 2.6.

На сторінці «Вакансії» наведена інформація про можливі ваканції для продавців та іншого персоналу.

На сторінці «Умови користування сайтом» та «Угода користувача» приведена інформація з якою має ознайомитись користувач при реєстрації оскільки він підписує угоду про надання свої персональних даних які не будуть передані іншій третій стороні

На сторінці «Контакти» розміщена вся необхідна інформація про сам інтернет-магазин «Сімейний шопінг». Тут знаходяться контакти магазину, адреса розташування магазину, телефони, години роботи. Також наведена гугл-мапа щоб клієнту було зрозуміло де розташований територіально в якому місті магазин. Оскільки на даному етапі проектування інформація про телефони та адресу коректно не вказана, юо це поки розробка інтернет магазину, який можна використати згодом. Приклад такої сторінки подано на рисунку 2.5.

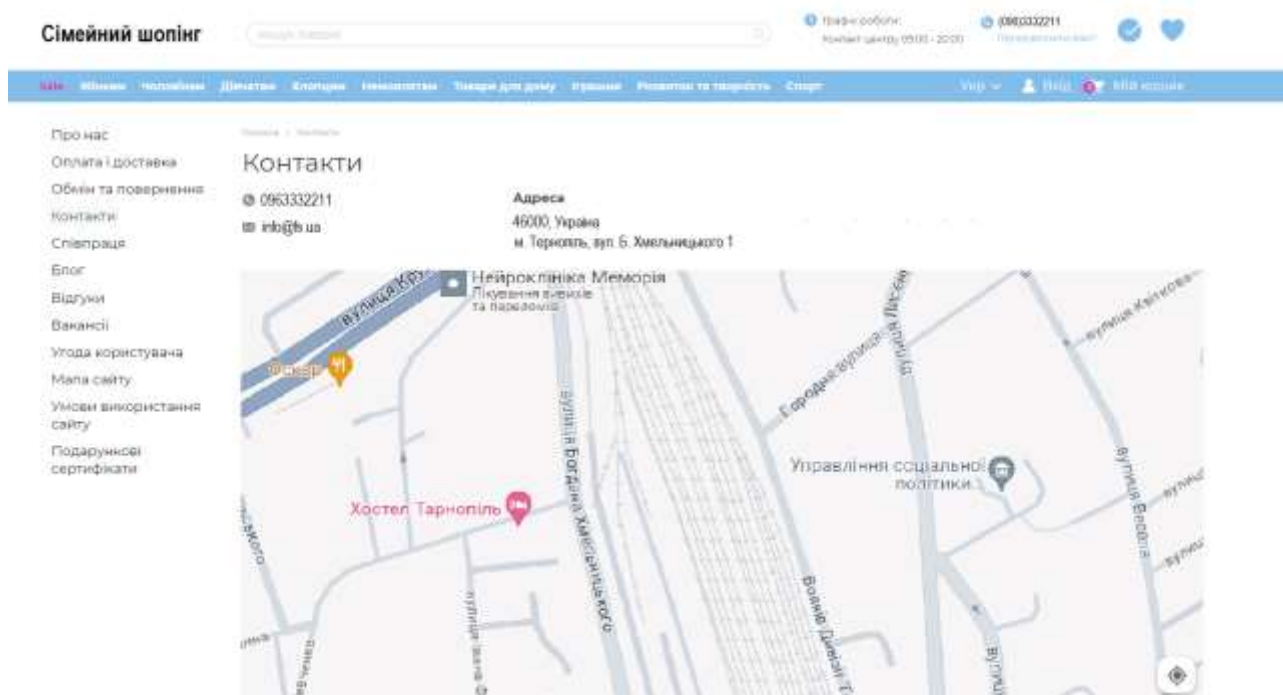


Рисунок 2.5 – Вигляд сторінки «Контакти» інтернет-магазину

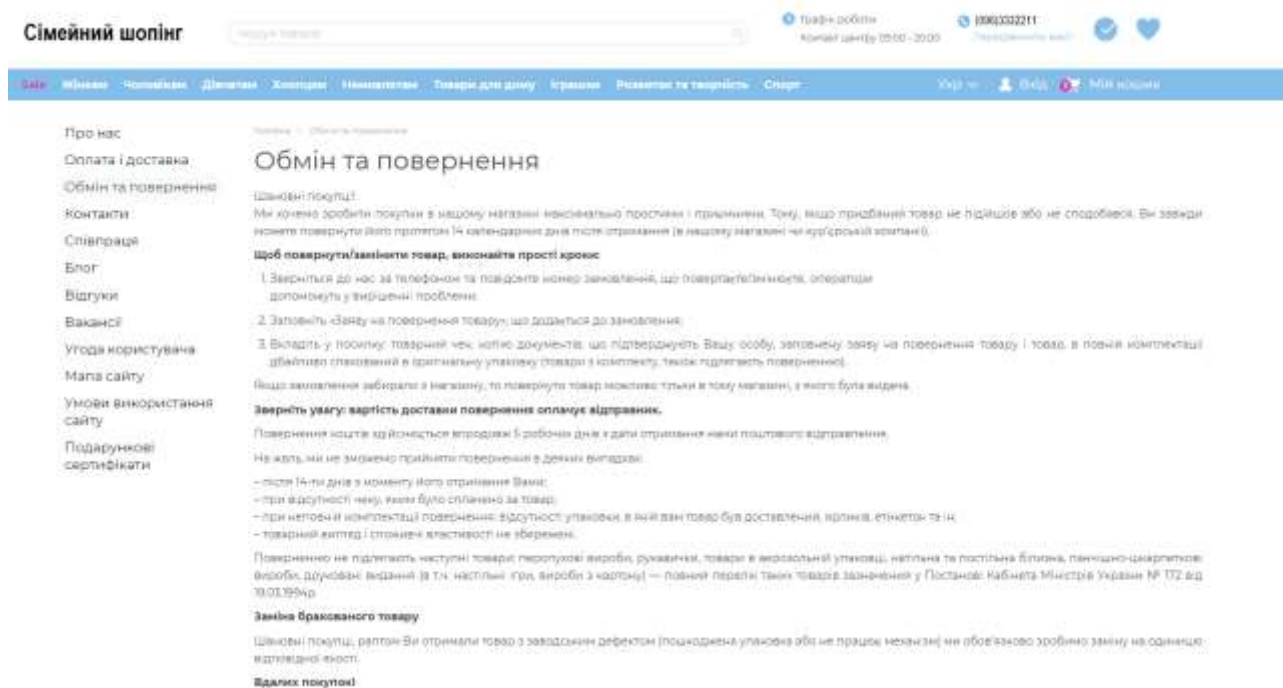


Рисунок 2.6 – Вигляд сторінки «Обмін та повернення» інтернет-магазину

Ще однією важливою сторінкою інтернет-магазину є сторінка «Доставка та оплата». Без цієї сторінки неміг би функціонувати жоден інтернет магазин.

Тому тут наведена інформація яким чином може доставлятися куплений товар. Тут наведені перевізники які діють на ринку України.

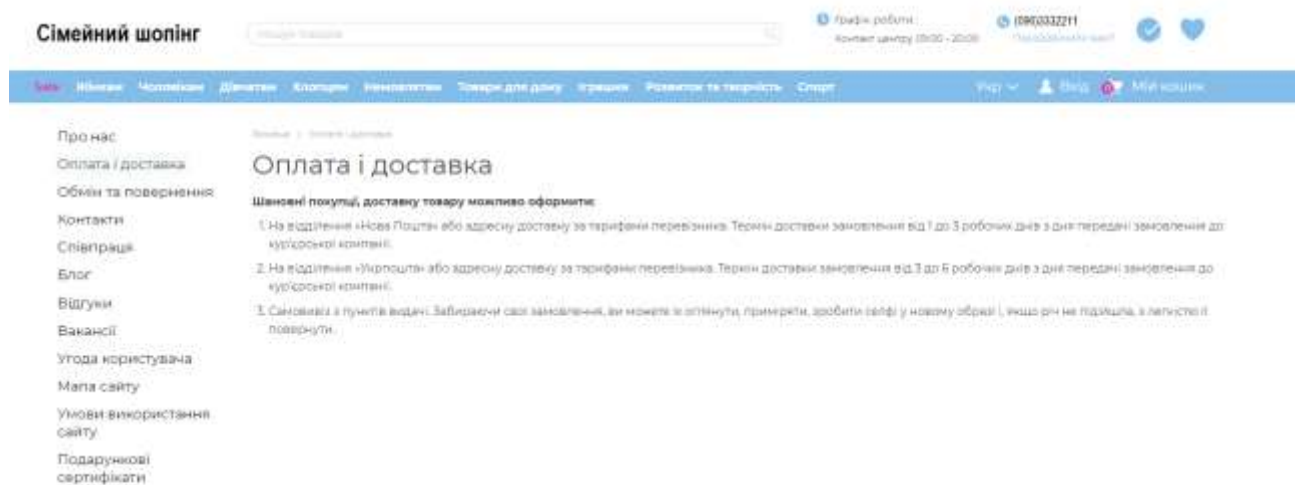


Рисунок 2.7 – Вигляд сторінки «Оплата і доставка» інтернет-магазину

Якщо інтернет-магазин вийде на міжнародний ринок то тут буде розміщена інформація про міжнародних перевізників.

### 2.3 Дизайн сторінок та кольорова гамма

Основним завданням, при розробці сторінок було вибрати кольорову гамму оскільки веб-сайт інтернет магазину повинен бути візуально привабливим. Відомо, що одна кольорова гама сторінки сприяє швидкому і повному сприйняттю змісту це білий колір. Даний web-сайт білому кольорі з додаванням незначної кількості блакитного. Колір логотипу чорний як відмічалось вище може бути перемальованим за вимогою замовника. В представленій розробці веб-сайту важливу роль відіграють картинки товарів. Тут графіка використовується не лише для карси, а для того щоб клієнт зміг оцінити товар який він можливо придбає.

Зверху, як зазначалось, розміщуються кнопки головного меню сайту. При

наведенні на кнопку напис змінює колір з білого на сірий, щоб користувач знав, що ним було вибрано відповідне посилання.

## 2.4 Принцип роботи сайту

Принцип роботи веб-сайту інтернет-магазину на JavaScript з використанням React, Node.js з Express і MongoDB зазвичай включає розподілення функціоналу між клієнтською (frontend) і серверною (backend) частинами, що співпрацюють між собою через API. Розглянемо коротко кожен з компонентів системи:

### Клієнтська частина (Frontend)

**React.js:** Веб-інтерфейс користувача (UI) будується з використанням React.js. React забезпечує можливість створювати компоненти, які утворюють сторінки, форми, кнопки та інші елементи інтерфейсу. Компоненти можуть бути статичними або динамічними, залежно від ваших потреб.

**Стейт у React:** Для управління станом додатку використовуються стейтові змінні (state). Вони дозволяють зберігати дані, які змінюються під час взаємодії користувача з додатком, такі як вміст кошика, інформація про користувача тощо.

**Взаємодія з сервером:** Для отримання та відправлення даних на сервер використовується HTTP або HTTPS протокол. Зазвичай для цього використовуються бібліотеки, такі як Axios або Fetch, які дозволяють виконувати AJAX-запити до API вашого сервера.

### Серверна частина (Backend)

**Node.js з Express:** Backend вашого веб-сайту будується на Node.js з використанням Express.js. Express є мінімальним і гнучким веб-фреймворком для Node.js, який дозволяє створювати API та обробляти запити від клієнтів.

**Модулі Node.js:** Використовуються різні модулі Node.js для забезпечення функціональності, такі як робота з базами даних (Mongoose для

MongoDB), обробка запитів (body-parser), робота зі зборками, аутентифікація тощо.

Взаємодія з базою даних MongoDB: MongoDB використовується як база даних для зберігання інформації про продукти, користувачів, замовлення і іншу важливу інформацію. MongoDB є NoSQL базою даних, що забезпечує гнучкість у схемі та швидкість доступу до даних.

Процес роботи сайту

Інтерфейс користувача: Клієнтська частина (frontend) відповідає за відображення інтерфейсу користувача. Вона реагує на дії користувача, відображає інформацію з бази даних і передає дані на сервер для обробки.

Обробка запитів на сервері: Серверна частина (backend) отримує запити від клієнта через API. Вона обробляє запити, взаємодіє з базою даних, обробляє бізнес-логіку (наприклад, додавання товарів до кошика, обробка замовлення), виконує аутентифікацію користувачів тощо.

Взаємодія з базою даних: Запити до бази даних (MongoDB) виконуються для збереження, оновлення та видалення даних. MongoDB забезпечує гнучкість у схемі даних, що сприяє швидкому розвитку і змінам в проекті.

Відповідь на клієнта: Після обробки запиту сервер повертає відповідь клієнту, яка може містити дані, які користувач запитував, або статус операції (наприклад, успішно чи неуспішно завершена операція).

Цей принцип роботи забезпечує високу масштабованість і гнучкість, що дозволяє ефективно розширювати функціонал інтернет-магазину в майбутньому.

## 2.5 Адміністрування бази даних MongoDB

Адміністрування бази даних MongoDB включає в себе кілька ключових аспектів, таких як установка, налаштування, моніторинг, забезпечення безпеки і відновлення даних. Розглянемо основні кроки, які повинні бути виконані для адміністрування MongoDB:

## 1. Встановлення MongoDB

На локальному комп'ютері:

Завантаження: Завантажте відповідну версію MongoDB з офіційного сайту (<https://www.mongodb.com/try/download/community>).

Інсталяція: Слідуйте інструкціям для вашої операційної системи для встановлення MongoDB.

На сервері:

Хмарні сервіси: Використовуйте хмарні сервіси, такі як MongoDB Atlas, для створення, керування і масштабування кластерів MongoDB.

## 2. Налаштування і конфігурація MongoDB

Файл конфігурації: MongoDB має файл конфігурації, де ви можете налаштувати параметри, такі як порт прослуховування, шлях до журналів, обсяги пам'яті тощо.

Аутентифікація і авторизація: Налаштуйте аутентифікацію і авторизацію для захисту вашої бази даних. Використовуйте ролі і користувачів для керування доступом.

## 3. Моніторинг та налагодження

MongoDB Metrics: Використовуйте вбудовані метрики MongoDB або сторонні інструменти для моніторингу продуктивності бази даних.

Профілювання запитів: Використовуйте профіль для аналізу роботи запитів до бази даних і виявлення можливих проблем.

## 4. Запасні копії та відновлення

Резервне копіювання: Регулярно створюйте резервні копії вашої бази даних MongoDB. Використовуйте утиліти, такі як `mongodump` для створення резервних копій.

Відновлення даних: Використовуйте `mongorestore` для відновлення даних з резервних копій у випадку потреби.

## 5. Оптимізація продуктивності

Індексація: Створюйте індекси для поліпшення продуктивності запитів до бази даних.

Фрагментація: Виправляйте фрагментацію, якщо вона виникає, за допомогою оптимізаційних процедур.

#### 6. Забезпечення безпеки

Оновлення: Регулярно оновлюйте MongoDB до останньої версії для отримання оновлень безпеки і нових функцій.

Firewall і безпека мережі: Використовуйте брандмауери і інші заходи для захисту сервера MongoDB від несанкціонованого доступу.

Ці кроки допоможуть ефективно адмініструвати базу даних MongoDB для веб-сайту інтернет-магазину, забезпечуючи надійність, безпеку і оптимальну продуктивність вашої бази даних.

## 2.6 Тестування веб-сайту інструментом Selenium

Selenium є потужним інструментом для автоматизації тестування веб-додатків. Він підтримує різні мови програмування (Java, Python, C#, Ruby, JavaScript) і дозволяє автоматизувати взаємодію з веб-браузерами, такими як Chrome, Firefox, Edge, Safari і іншими. Наведемо ключові аспекти використання Selenium для тестування веб-сайтів:

#### Основні можливості Selenium:

Автоматизація браузерів: Selenium дозволяє програмістам і тестерам автоматизувати взаємодію з веб-браузерами через драйвери (наприклад, ChromeDriver, GeckoDriver для Firefox).

Робота з різними мовами програмування: Selenium підтримує різні мови програмування, що дає можливість вибирати ту мову, з якою ви зручніше працювати.

Взаємодія з елементами сторінки: Ви можете виконувати клікання, введення тексту, отримувати атрибути елементів, перевіряти стани чекбоксів, радіокнопок та інші дії з елементами на сторінці.

Тестування різних сценаріїв: Ви можете створювати складні тестові сценарії, які перевіряють різні аспекти функціональності веб-сайту, такі як



реєстрація користувача, додавання товарів до кошика, проходження через касу тощо.

Інтеграція з іншими інструментами: Selenium може інтегруватися з іншими інструментами для тестування, такими як JUnit, TestNG, NUnit, а також з CI/CD системами для автоматизації тестування під час процесу розробки.

Популярні інструменти і бібліотеки для Selenium:

Selenium WebDriver: Основний інструмент для автоматизації тестування веб-додатків.

Selenium Grid: Для паралельного виконання тестів на різних пристроях та браузерах.

Selenium IDE: Розширення для браузера для запису і відтворення простих тестових сценаріїв.

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time

# Ініціалізація драйвера для браузера Chrome
driver = webdriver.Chrome()

# Відкриття веб-сайту
driver.get("https://www.example.com")

# Виконання JavaScript коду на сторінці
driver.execute_script("document.getElementById('myElement').style.backgro
undColor = 'yellow';")

# Затримка для перевірки змін на сторінці (зазвичай необхідна для
відлагодження)
time.sleep(2)
```

```
# Пошук елемента за допомогою XPath
element = driver.find_element(By.XPATH, "//input[@name='username']")

# Введення тексту в поле введення
element.send_keys("myusername")

# Натискання клавіші Enter
element.send_keys(Keys.RETURN)

# Закриття браузера
driver.quit()
```

Цей код демонструє базову взаємодію з веб-сторінкою за допомогою Selenium і мови програмування Python. Selenium надає широкі можливості для автоматизації тестування веб-додатків і є популярним вибором серед розробників і QA-інженерів для забезпечення якості веб-продуктів.

## **2.7 Висновок до другого розділу**

В другому розділі кваліфікаційної роботи описана розробка веб-сайту, наведені основні його сторінки. Описано принцип роботи сайту та адміністрування бази даних MongoDB. Описані основні елементи керування сайтом, проведено тестування з веб-сайту.

## **РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ**

### **3.1 Перелік заходів захисту від небезпечних і шкідливих чинників**

Вимоги безпеки при роботі оператора ПЕОМ.

В якості оператора ПЕОМ можуть працювати особи, не молодші 18 років, які склали відповідний іспит по правилам технічної експлуатації електроустановок споживачів і правилам техніки безпеки при експлуатації електроустановок споживачів, які пройшли інструктаж на робочому місці, відповідний медичний огляд, засвоїли безпечні методи виконання робіт по спеціальності.

Робота на ПЕОМ пов'язана з можливістю виникнення безпечних і шкідливих факторів:

- з'явленням підвищеного значення напруги електроланцюга замкнення якоє може виникнути через тіло людини у випадку доторкання до відкритих струмоведучих частин чи електрообладнання і електропроведення зпорушеною ізоляцією;
- наявністю обертаючих частин охолоджуючих машин, вентиляційних і інших пристроїв ПЕОМ;
- виникнення підвищеного рівня шуму у приміщеннях обчислювального центру, в особливості на робочих місцях, машинних залах;
- перенапруження зору при роботі з електронними пристроями в особливості при нераціональній освітленості робочих поверхонь у зоні відеотерміналів і нераціональному розташуванню екрану по відношенню до очей.

### **3.2 Безпечні приклади роботи та охорона праці на робочому місці**

При роботі оператора ПЕОМ необхідно бути уважним, не відвертати уваги на сторонні справи і не відвертати увагу у других про всі виявленні

несправності негайно повідомити начальнику ПЕОМ, а в аварійних ситуаціях негайно зупинити роботу; дотримуватись правил особистої гігієни, харчуватися тільки у спеціально обладнаних для цієї мети місцях; зберігати одяг в відведених місцях.

1. Уся розводка електроживлення у кабінеті ПЕОМ повинна бути у трубах (металорукавах). Розетки для схованої проводки допускається встановлювати тільки на цегляних і бетонних стінах. На стінах іншого типу, в тому числі цегляних з покриттям сухою штукатуркою, допускається установка тільки розеток з класом захисту не нижче за ІІ.

2. Розподільний щиток повинен бути встановлений у середині приміщення бажано біля вхідних дверей. Загальний вимикач, який відключає напругу з усіх приладів, повинен бути встановлений всередині приміщення безпосередньо біля вхідних дверей.

3. Не допускається загороджувати проходи до загального вимикача на розподільного щитку.

4. Підключення відключення приладів від мережі, з'єднання з іншими приладами проводиться тільки учителем.

#### Правила експлуатації ПК.

1. Перед першим вмиканням ПК потрібно перевірити: чи відповідає напруга у мережі напрузі, на яку розрахований ПК (багато з них можуть працювати при декількох значеннях вхідної напруги, наприклад, 200 В и 110 В). При необхідності встановити перемикач напруги на ПК у правильне положення.

2. При можливості підключити до ПК прилад безперервного живлення (UPS), що дає можливість роботи комп'ютерів при повному відключенні електроживлення від 5 хвилин до декількох годин (у залежності від сили приладу). За цей час можна закінчити роботи на комп'ютері, щоб при його включенні не була втрачена інформація

3. При включенні комп'ютера необхідно:

- включити UPS, якщо комп'ютер підключений через UPS;

- включити периферійні пристрої (монітор, принтер та ін.);
- включити комп'ютер перемикачем на корпусі.

Після цього на екрані монітора з'явиться повідомлення про хід програми перевірки і початкового завантаження комп'ютера. Коли початкове завантаження операційної системи (ОС) буде закінчене, з'явиться запрошення DOS, наприклад: C:/>, або завантажиться Windows.

Поява запрошення означає, що DOS готова до прийому команд.

4. При виключенні ПК необхідно:

- закінчити працюючі програми;
- вийти з Windows;
- виключити комп'ютер перемикачем на корпусі;
- виключити периферійні пристрої ( монітор, принтер та ін.);
- виключити UPS, якщо комп'ютер підключений через UPS.

Корпус ПК, клавіатура і інші блоки і пристрої ПК повинні мати матову поверхню одного кольору з коефіцієнтом віддзеркалення 0,4 - 0,6 і не мати блискучих деталей, здатних створювати відблиски.

Положення монітора повинне бути таким, щоб світло на нього падало під кутом. Екран монітора повинен розташовуватися приблизно на відстані 28 – 60 см від оператора, причому верхній край екрану повинен знаходитися на рівні очей. Рекомендується по можливості зменшувати інтенсивність світла люмінесцентних джерел. Для монітора повинні бути передбачені ручки регулювання яскравості і контрасту зображення, що забезпечить можливість регулювання цих параметрів від мінімальних до максимальних значень.

При необхідності рекомендується застосовувати захисні фільтри для ослаблення дії випромінювання екрану, зменшення відображеного відблиску зображення, зниження статичного заряду. Використання захисних фільтрів робить менш помітним мерехтіння монітора.

### 3.3 Оцінка ризиків БЖД у життєдіяльності людини

У виробничих умовах розрізняють індивідуальний і колективний ризик.

*Індивідуальний ризик* характеризує реалізацію небезпеки відповідного виду діяльності для конкретного індивіда.

*Колективний ризик* – це травмування чи загибель двох або більше чоловік від впливу небезпечних та шкідливих виробничих факторів.

Класифікації джерел небезпек та рівні ризику загибелі людини за даними літературних джерел наведені в табл. 4.1.

*Прийнятний ризик* поєднує в собі технічні, економічні, соціальні і політичні аспекти і є деяким компромісом між рівнем безпеки і можливим її досягненням.

Перш за все треба мати на увазі, що підвищення рівня захисту від небезпеки автоматично підвищує загальну вартість виробництва. У вимогах досягнення нульового ризику ситуація може повернутися до людей соціальною трагедією за сценаріями безробіття.

При підвищенні витрат на удосконалення обладнання технічний ризик знижується, але зростає соціальний. Сумарний ризик мінімальний, коли створене необхідне спів відношення між інвестиціями в технічну і соціальну сфери. Ці обставини треба враховувати під час вибору прийнятного ризику.

Таблиця 3.1

Класифікація джерел і рівнів ризиків загибелі людини в промислово розвинутих країнах ( $R$  – кількість смертельних випадків люд<sup>-1</sup> • рік<sup>-1</sup>)

Джерела	Принципи	Середнє значення
Внутрішнє середовище організму	Генетичні і соматичні захворювання, старіння	$R_{cp} = (0,6-1) \cdot 10^{-2}$

Природне середовище існування	Нещасні випадки від стихійних лих (землетруси, урагани, повені та ін.)	$R_{cp}=10^{-6}$ ; повені $4*10^{-5}$ ; землетруси $3*10^{-5}$ ; грози $6*10^{-7}$ ; урагани $10^{-3}$
Техносфера	Нещасні випадки в побуті, на транспорті, захворювання від забруднення навколишнього середовища	$R_{cp}=10^{-3}$
Професійна діяльність	Професійні хвороби, нещасні випадки на виробництві (під час професійної діяльності)	Професійна діяльність: безпечна $R_{cp}<10^{-4}$ ; відносно безпечна $R_{cp}=10^{-4}-10^{-3}$ ; небезпечна $R_{cp}=10^{-4}-10^{-2}$ ; особливо небезпечна $R_{cp}>10^{-2}$
Соціальне середовище	Самогубства, каліцтва, злочинні дії, військові дії та ін.	$R_{cp}=(0,5-1,5)*10^{-4}$

Рівень прийняттого ризику за міжнародною домовленістю вирішено рахувати в межах  $10^{-7}$ – $10^{-6}$  (смертельних випадків люд<sup>-1</sup> • рік<sup>-1</sup>), а величина  $10^{-6}$  є максимально прийнятим індивідуальним ризиком.

*Вмотивований (обґрунтований) і немотивований (необґрунтований) ризик.* У випадках виробничих аварій, пожеж, з метою рятування людей, які постраждали від аварій та іншого, людині доводиться йти на ризик. Обґрунтованість такого ризику пов'язана з необхідністю надання допомоги людям, що потерпіли, бажанням урятувати від руйнування цінне обладнання чи споруди господарства.

Невиконання робітниками правил безпеки, технологічних процесів, невикористання засобів захисту і таке інше формують небезпечний ризик, який, як правило, становить передумови виникнення травм і аварій на виробництві.

В технічній сфері поняття ризику визначають дещо інакше порівняно зі звичайними оцінками. Так, за ризик мають кількісну характеристику дії небезпек, які формуються за конкретною діяльністю людини, чи інакше – кількість смертельних випадків, випадків захворювання, випадків тимчасової і стійкої непрацездатності (інвалідності), зумовлених дією на людину конкретної небезпеки відносно до визначеної кількості жителів (робітників).

Ризик формується як величина, що визначає добуток величини події на міру можливого її початку.

### **3.4 Висновок до третього розділу**

В третьому розділі кваліфікаційної роботи описано перелік заходів захисту від небезпечних і шкідливих чинників. При цьому, описано основні вимоги безпеки при роботі оператора ПЕОМ. Окрім цього описані безпечні приклади роботи та охорона праці на робочому місці. У разі виникнення аварійних ситуацій описані кроки які необхідно зробити. Окрім цього описано оцінку ризиків БЖД у життєдіяльності людини.



## ВИСНОВКИ

В даній кваліфікаційній роботі розглянуто основні технології які використовують при розробці веб-сайтів, а саме JavaScript, MongoDB, HTML5 і CSS3, які були використані для розробки веб-сайту.

При створенні веб-сайту було опрацьовано наступні задачі:

- Проаналізовано подібні розробки веб-сайтів;
- Проаналізовано типові технології при розробці веб-сайтів, були обрані: мова JavaScript та система керування базою даних MongoDB для керування та збереження інформації в магазині;
- Здійснено розробку дизайну веб-сайту та його структуру;
- Здійснено розробку основних сторінок магазину та необхідних елементів керування;
- Проведено тестування розробленого веб-сайту.

В третьому розділі кваліфікаційної роботи описано перелік заходів захисту від небезпечних і шкідливих чинників. При цьому, описано основні вимоги безпеки при роботі оператора ПЕОМ. Окрім цього описані безпечні приклади роботи та охорона праці на робочому місці.

## ПЕРЕЛІК ДЖЕРЕЛ

1. Крис Джамса, Конрад Кинг, Энди Андерсон. Креативный Web-дизайн. HTML, XHTML, CSS, JavaScript, PHP, ASP, ActiveX. Текст, графика, звук и анимация. Пер. с англ. – М: ООО “ДиаСофтЮП”, 2005. – 672 с.
2. Verou, Lea. "CSS Secrets: Better Solutions to Everyday Web Design Problems". *A List Apart*, No. 377, 2013, pp. 1-12.
3. Robbins, Jennifer Niederst. "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics". O'Reilly Media, <https://learning.oreilly.com/library/view/learning-web-design/9781449333362/>, Accessed 27 June 2024.
4. Duckett, Jon. *HTML and CSS: Design and Build Websites*. Wiley, 2011.
5. Закас Ніколас. JavaScript для професіоналів. — Київ: Видавничий дім "Букрек", 2018. — 752 с.
6. Stefanov, S. (2016). *React: Up & Running: Building Web Applications*. O'Reilly Media. ISBN: 978-1491931820.
7. Verou, Lea. "CSS Secrets: Better Solutions to Everyday Web Design Problems". *A List Apart*, No. 377, 2013, pp. 1-12.
8. Robbins, Jennifer Niederst. "Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics". O'Reilly Media, <https://learning.oreilly.com/library/view/learning-web-design/9781449333362/>, Accessed 27 June 2024.
9. Banks, A., Porcello, E. (2017). *Learning React: Functional Web Development with React and Redux*. O'Reilly Media. ISBN: 978-1491954621.
10. Freeman, A. (2019). *Pro React 16*. Apress. ISBN: 978-1484244508.
11. Fedosejev, A. (2017). *React.js Essentials*. Packt Publishing. ISBN: 978-1783551620.
12. Bertoli, M. (2017). *React Design Patterns and Best Practices*. Packt Publishing. ISBN: 978-1786464538.

13. Дакетт Джон. JavaScript і jQuery: Інтерактивна веб-розробка. — Київ: Видавництво "Діалектика", 2019. — 544 с.
14. Браун Етан. JavaScript: Підручник з програмування. — Київ: Видавничий дім "Альтерпрес", 2020. — 416 с.
15. Флегель Девід. JavaScript. Повний курс. — Київ: Видавництво "Кодекс", 2017. — 608 с.
16. Морган Нік. JavaScript для дітей. — Київ: Видавництво "Комп'ютерна академія", 2021. — 160 с.
17. В.А. Гайченко, Г.М. Коваль : “Основи БЖД людини” : Навч. посіб. – 2-ге вид. стереотип. – К.: МАУП, 2004.-232с.: іл.-Бібліогр.: с. 225-226
18. Джигрей В.С., Жидецький В.Ц. “Безпека життєдіяльності”. Підручник.- Вид. 4-те, доповнене. – Львів: Афіша, 2001. – 256с.
19. Скобло Ю. С, Соколовська Т. Б., Мазоренко Д. І., Тіщенко Л. М., Троянов М. М. – Безпека життєдіяльності: Навчальний посібник для вищих навчальних закладів III – IV рівнів акредитації. – Київ: Кондор, 2003. – 424с.
20. Желібо Є.П., Заверуха Н.М., Зацарний В.В. : “Безпека життєдіяльності” : Навчальний посібник для студентів вищих закладів освіти України I-IV рівнів акредитації / За ред. Є.П. Желібо, В.М. Пічі. – Львів : Піча Ю.В., К.: “Каравела”, Львів:, Новий світ – 2000, 2002, - 328с.
21. Конспект лекцій з дисципліни «Основи безпеки життєдіяльності».
22. Конспект лекцій з дисципліни «Web-програмування та web-дизайн».
23. Пам'ятка-інструктаж з питань охорони праці для студентів Тернопільського державного технічного університету імені Івана Пулюя. – Тернопіль.
24. Hughes-Croucher, T., Wilson, M. (2012). Node: Up and Running: Scalable Server-Side Code with JavaScript. O'Reilly Media. ISBN: 978-1449398583.
25. Cantelon, M., Harter, M., Rajlich, T., Osmani, A. (2013). Smashing Node.js: JavaScript Everywhere. Wiley. ISBN: 978-1119962595.
26. Mead, A. (2019). The Complete Node.js Developer Course (3rd Edition). Udemy Course.

27. Casares, M. (2014). *Mastering Node.js*. Packt Publishing. ISBN: 978-1782166320.
28. Hapi.js Team. (2015). *Hapi.js in Action*. Manning Publications. ISBN: 978-1617292735.
29. Methodology of the Formation of Sports Matches Statistical Information Using Neural Networks Sorokivska, O., Lytvynenko, I., Sorokivskyi, O., Kozbur, H., Strutynska, I. CEUR Workshop Proceedings, 2023, 3628, pp. 389–403
30. Information Technologies Use in the Study of Functional Properties of Wheeled Vehicles Lyashuk, O., Stashkiv, M., Lytvynenko, I., Sakhno, V., Khoroshun, R. CEUR Workshop Proceedings, 2023, 3628, pp. 370–381
31. Software for statistical processing and modeling of a set of synchronously registered cardio signals of different physical nature Lupenko, S., Lytvynenko, I., Sverstiuk, A., Horkunenko, A., Shelestovskyi, B. CEUR Workshop Proceedings, 2021, 2864, pp. 194–205
32. Я.В. Литвиненко, М.В. Ілько. Розробка програмного забезпечення з використанням нейронної мережі в задачах прогнозування електронавантажень. // Тези доповідей ІХ науково-технічної конференції «Інформаційні моделі, системи та технології». 2021. – С. 45-46.
33. П.А. Ониськів, Я.В. Литвиненко. Аналіз програмного забезпечення для покращення якості вхідних зображень. Матеріали міжнародної наукової конференції „Іван Пулюй: життя в ім'я науки та України “(до 175-ліття від дня народження). 2020 року, Тернопіль, Україна 2020. ФОП Паляниця ВА. С. 90
34. Конспект лекцій з дисципліни “Основи технічної творчості та наукових досліджень” (частина 1) для здобувачів освітнього ступеня «бакалавр», спеціальностей: 122 «Комп'ютерні науки», 126 «Інформаційні системи та технології», всіх форм навчання. Укладач Я.В. Литвиненко: ТНТУ, 2020 р.
35. Конспект лекцій з дисципліни “Основи технічної творчості та наукових досліджень” (частина 2) для здобувачів освітнього ступеня «бакалавр», спеціальностей: 122 «Комп'ютерні науки», 126 «Інформаційні

системи та технології», всіх форм навчання. Укладач Я.В. Литвиненко: ТНТУ, 2020 р.

36. Chodorow, K. (2013). MongoDB: The Definitive Guide. O'Reilly Media. ISBN: 978-1449344689.

37. Plugge, E., Hawkins, T., Membrey, P. (2013). MongoDB Basics. Apress. ISBN: 978-1430258216.

38. Dirolf, D., Wilson, D., Lafrance, N. (2016). MongoDB in Action (2nd Edition). Manning Publications. ISBN: 978-1617291608.

39. Banker, K., Panchal, A., Sarang, M. (2017). MongoDB Applied Design Patterns. Packt Publishing. ISBN: 978-1785289989.

40. Chodorow, K. (2013). Scaling MongoDB. O'Reilly Media. ISBN: 978-1449344238.

# ДОДАТКИ

## Приклад сторінки веб-сайту

```

<!DOCTYPE html>
<html prefix="og: https://ogp.me/ns# article: https://ogp.me/ns/article# product:
https://ogp.me/ns/product#" lang="uk">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width">
  <title>▷ Купити жіночий одяг, взуття та аксесуари в Україні – fs (сімейний
шопінг)</title>
  <meta name="robots" content="all">
  <meta name="description" content="Жіночий одяг, взуття та аксесуари ▷
Замовляйте в fs ★ Великий асортимент та низькі ціни ➔ Швидка доставка по
Україні або самовивіз із магазинів Сімейний шопінг">
  <link rel="canonical" href="https://fs.ua/dlia-zhinok/">
  <link rel="next" href="https://fs.ua/dlia-zhinok/filter/page=2/">
  <link rel="alternate" hreflang="uk" href="https://fs.ua/dlia-zhinok/">
  <link rel="alternate" hreflang="ru" href="https://fs.ua/ru/dlia-zhinok/">
  <meta property="og:locale" content="ua_UA">
  <meta property="og:site_name" content="fs (сімейний шопінг) - Shket&Faina
сімейний шопінг для всієї родини">
  <meta property="og:title" content="▷ Купити жіночий одяг, взуття та аксесуари в
Україні – fs (сімейний шопінг)">
  <meta property="og:url" content="https://fs.ua/dlia-zhinok/">

```

<meta property="og:description" content="Жіночий одяг, взуття та аксесуари ►  
Замовляйте в fs ★ Великий асортимент та низькі ціни ➔ Швидка доставка по  
Україні або самовивіз із магазинів Сімейний шопінг">

<meta property="og:image"  
content="https://fs.ua/content/images/16/1200x1200180bc0/76612227373261.webp">

<meta property="og:type" content="product.group">

<link rel='preload' as='image'  
href='/content/images/11/400x600195nn0/86083789045580.webp'><link rel='preload'  
as='image' href='/content/images/50/400x600195nn0/91734473016440.webp'>

<script>

(function (w) {

w.GLOBAL = {

SYSTEM\_LANGUAGE: 3,

FIRST\_LANGUAGE: 3,

DOMAIN: "fs.ua",

SITE\_NAME: "fs",

URI\_PREFIX: "/",

URI\_SUFFIX: "/",

theme: "horoshop\_default",

currency: {"id": "1", "i18n\_language": "0", "title": "UAH", "abbr": "грн", "position": "2", "sortorder": "1", "iso": "UAH", "abbr\_penny": "коп.", "enabled\_in\_frontend": "1", "sign": "₴"},

round\_to: "0",

isGuest: true,

USER\_ID: null,

open\_cart\_on\_add: true,

cart\_animation: false,

phone\_mask: "+38 (099) 999-99-99",

use\_countries: false,

enable\_phone\_mask\_on\_input: true,



```

marketingEvents: {"cart_creation":[function (model, eventId) {
  try {
    ga('send', 'event', 'Корзина', 'Создание корзины', GLOBAL.DOMAIN);
  } catch(e) {
    console.error(e);
  }
}],function (model, eventId) {
  try {
    var products = model.Cart.products,
ids = [],
totalPrice = model.Cart.total.sum;

for (key in products) {
  ids.push(products[key].article + "");
}

gtag('event', 'Создание корзины', {
  'event_category' : 'Корзина',
  'event_label'    : 'GLOBAL.DOMAIN',
  'ecomm_prodid'   : ids,
  'ecomm_pagetype' : 'cart',
  'ecomm_totalvalue': totalPrice
});
  } catch(e) {
    console.error(e);
  }
}], "cart_product_add":[function (model, eventId) {
  try {
    ga('ec:addProduct', {
'id': model.article + ",

```

```

'name': model.title,
'price': model.price * 1,
'quantity': model.quantity * 1
});
ga('ec:setAction', 'add');
ga('send', 'event', 'Корзина', 'Добавление товара в корзину',
GLOBAL.DOMAIN);
    } catch(e) {
        console.error(e);
    }
},function (model, eventId) {
    try {
        dataLayer.push({
"event": "addToCart",
"ecommerce": {
    "currencyCode": GLOBAL.currency.iso,
    "add": {
        "products": [
            {
                "id": model.article + "",
                "name": model.title + " (артикул: " + model.article_for_display + ")",
                "price": model.price * 1,
                "quantity": model.quantity
            }
        ]
    }
}
});
    } catch(e) {
        console.error(e);
    }
}

```

```

    }
  },function (model, eventId) {
    try {
      (function () {
var products = null;

fbq('track', 'AddToCart', {
  contents: [
    {
      'id': model.article,
      'quantity': model.quantity * 1,
      'item_price': model.price * 1
    }
  ],
  content_type: 'product',
  currency: GLOBAL.currency.iso,
  value: model.quantity * model.price
},
{eventID: eventId}
);
})();
    } catch(e) {
      console.error(e);
    }
  },function (model, eventId) {
    try {

const products = [];
for (const [key, value] of Object.entries(model.Cart.products)) {
  products.push({

```

```

        'productKey': String(value.article),
        'price': String(value.price),
        'quantity': String(value.quantity),
        'currency': GLOBAL.currency.iso
    });
}

eS('sendEvent', 'StatusCart', {
    'StatusCart': products,
    'GUID': generateUUID()
});

    } catch(e) {
        console.error(e);
    }
},function (model, eventId) {
    try {
        var products = model.Cart.products,
        ids = [],
        totalPrice = model.Cart.total.sum;

    for (key in products) {
        ids.push(products[key].article + "");
    }

    gtag('event', 'add_to_cart', {
        'currency': GLOBAL.currency.iso,
        'value': totalPrice,
        'items': [
            {

```

```

        'item_id': model.article + ",
        'item_name': model.title,
        'quantity': model.quantity * 1,
        'price': model.price * 1
    }
],
'ecomm_prodid': ids,
'ecomm_pagetype': 'cart',
'ecomm_totalvalue': totalPrice
});

    } catch(e) {
        console.error(e);
    }
}], "cart_product_update": [function (model, eventId) {
    try {

const products = [];
for (const [key, value] of Object.entries(model.Cart.products)) {
    products.push({
        'productKey': String(value.article),
        'price': String(value.price),
        'quantity': String(value.quantity),
        'currency': GLOBAL.currency.iso
    });
}

eS('sendEvent', 'StatusCart', {
    'StatusCart': products,
    'GUID': generateUUID()
});

```

```

    } catch(e) {
        console.error(e);
    }
}], "cart_product_remove": [function (model, eventId) {
    try {
        ga('ec:removeProduct', {
'id': model.article + ",
'name': model.title,
'price': model.price * 1,
'quantity': model.quantity * 1
});
ga('ec:setAction', 'remove');
ga('send', 'event', 'Корзина', 'Удаление товара из корзины', GLOBAL.DOMAIN);
    } catch(e) {
        console.error(e);
    }
}], function (model, eventId) {
    try {
        dataLayer.push({
"event": "removeFromCart",
"ecommerce": {
    "currencyCode": GLOBAL.currency.iso,
    "remove": {
        "products": [
            {
                "id": model.article + "",
                "name": model.title + " (артикул: " + model.article_for_display + ")",
                "price": model.price * 1,
                "quantity": model.quantity
            }
        ]
    }
}
    }
}

```

```

        }
    ]
}
}
});

    } catch(e) {
        console.error(e);
    }
},function (model, eventId) {
    try {

const products = [];
for (const [key, value] of Object.entries(model.Cart.products)) {
    products.push({
        'productKey': String(value.article),
        'price': String(value.price),
        'quantity': String(value.quantity),
        'currency': GLOBAL.currency.iso
    });
}

eS('sendEvent', 'StatusCart', {
    'StatusCart': products,
    'GUID': generateUUID()
});

    } catch(e) {
        console.error(e);
    }
},function (model, eventId) {

```

```

    try {
        var products = model.Cart.products,
        ids = [],
        totalPrice = model.Cart.total.sum;

    for (key in products) {
        ids.push(products[key].article + "");
    }

    gtag('event', 'remove_from_cart', {
        'currency': GLOBAL.currency.iso,
        'value': totalPrice,
        'items': [
            {
                'item_id': model.article + "",
                'item_name': model.title,
                'quantity': model.quantity * 1,
                'price': model.price * 1
            }
        ],
        'ecomm_prodid': ids,
        'ecomm_pagetype': 'cart',
        'ecomm_totalvalue': totalPrice
    });

    } catch(e) {
        console.error(e);
    }

    }], "quick_search": [], "checkout_success": [function (order_id, cart, name, email,
phone, eventId) {
    try {

```



```

var products = cart.products;

for (var i = 0, l = products.length; i < l; ++i) {
  var product = products[i];
  ga("ec:addProduct", {
    "id": product.article + "",
    "name": product.title + " (артикул: " + product.article_for_display + ")",
    "price": product.price * 1,
    "quantity": product.quantity * 1
  });
}

ga("ec:setAction", "purchase", {
  "id": order_id,
  "affiliation": GLOBAL.SITE_NAME,
  "revenue": cart.total.total.sum,
  "tax": "0",
  "shipping": cart.delivery_price,
  "coupon": cart.total.coupon_data.code
});
ga("send", "event", "Оформление заказа", "Успешное оформление заказа",
GLOBAL.DOMAIN);
  } catch(e) {
    console.error(e);
  }
},function (order_id, cart, name, email, phone, eventId) {
  try {
    var products = cart.products;
var dataLayerProducts = [];

```

```

for (var i = 0, l = products.length; i < l; ++i) {
  var product = products[i];
  dataLayerProducts.push({
    "id": product.article + "",
    "name": product.title + " (артикул: " + product.article_for_display + ")",
    "price": product.price * 1,
    "quantity": product.quantity * 1
  });
}
dataLayer.push({
  "ecommerce": {
    "currencyCode": GLOBAL.currency.iso,
    "purchase": {
      "actionField": {
        "id" : order_id,
        "revenue" : cart.total.total.sum,
        "coupon" : cart.total.coupon_data.code
      },
      "products": dataLayerProducts
    }
  }
});
} catch(e) {
  console.error(e);
}
},function (order_id, cart, name, email, phone, eventId) {
  try {
    var products = cart.products;
var dataLayerProducts = [];

```

```

for (var i = 0, l = products.length; i < l; ++i) {
  var product = products[i];
  dataLayerProducts.push({
    "id": product.article + "",
    "name": product.title + " (артикул: " + product.article_for_display + ")",
    "price": product.price * 1,
    "quantity": product.quantity * 1
  });
}
dataLayer.push({
  "event": "gtm-enhanced-ecommerce",
  "ecommerce": {
    "currencyCode": GLOBAL.currency.iso,
    "purchase": {
      "actionField": {
        "id" : order_id,
        "revenue" : cart.total.total.sum,
        "coupon" : cart.total.coupon_data.code
      },
      "products": dataLayerProducts
    }
  }
});
} catch(e) {
  console.error(e);
}
},function (order_id, cart, name, email, phone, eventId) {
  try {
    var products = cart.products, contents = [];

```

```

for (var i = 0, l = products.length; i < l; ++i) {
  var product = products[i];
  contents.push({
    'id': product.article,
    'quantity': product.quantity * 1,
    'item_price': product.price * 1
  });
}

fbq('track', 'Purchase', {
  contents: contents,
  content_type: 'product',
  value: cart.total.total.sum,
  currency: GLOBAL.currency.iso},
  {eventID: eventId}
);
  } catch(e) {
    console.error(e);
  }
},function (order_id, cart, name, email, phone, eventId) {
  try {
    var items = [],
products = cart.products,
ids = [];

for (var i = 0, l = products.length; i < l; ++i) {
  var product = products[i];

  items.push({
    'item_id': product.article,

```

```

    'item_name': product.title + ' (артикул: ' + product.article_for_display + ')',
    'quantity': product.quantity * 1,
    'price': product.price * 1
  });

  ids.push(product.article);
}

gtag('event', 'purchase', {
  'transaction_id': order_id,
  'affiliation': GLOBAL.SITE_NAME,
  'value': cart.total.total.sum,
  'currency': GLOBAL.currency.iso,
  'tax': 0,
  'shipping': cart.delivery_price,
  'coupon': cart.total.coupon_data.code,
  'items': items,
  'ecomm_prodid': ids,
  'ecomm_pagetype': 'purchase',
  'ecomm_totalvalue': cart.total.total.sum
});

  } catch(e) {
    console.error(e);
  }
}], "newsletter_subscription": [function (model, eventId) {
  try {
    ga('send', 'event', 'Рассылка', 'Подписка на рассылку',
GLOBAL.DOMAIN);
  } catch(e) {
    console.error(e);
  }
}

```

```

    }
  },function (model, eventId) {
    try {
      fbq('track', 'Lead');
    } catch(e) {
      console.error(e);
    }
  },function (model, eventId) {
    try {
      gtag('event', 'Подписка на рассылку', {
'event_category': 'Рассылка',
'event_label' : 'GLOBAL.DOMAIN'
});
    } catch(e) {
      console.error(e);
    }
  }], "callback_submit": [function (model, eventId) {
    try {
      ga('send', 'event', 'Обратный звонок', 'Запрос обратного звонка',
GLOBAL.DOMAIN);
    } catch(e) {
      console.error(e);
    }
  },function (model, eventId) {
    try {
      fbq('track', 'Lead', {}, {eventID: eventId});
    } catch(e) {
      console.error(e);
    }
  },function (model, eventId) {

```

```

    try {
        gtag('event', 'Запрос обратного звонка', {
            'event_category': 'Обратный звонок',
            'event_label' : 'GLOBAL.DOMAIN'
        });
    } catch(e) {
        console.error(e);
    }
},function (model, eventId) {
    try {
        dataLayer.push({'event': 'callbackRequest'})
    } catch(e) {
        console.error(e);
    }
}], "registration_success": [function (model, eventId) {
    try {
        fbq('track', 'CompleteRegistration', {}, {eventID: eventId});
    } catch(e) {
        console.error(e);
    }
}],function (model, eventId) {
    try {
        eS('sendEvent', 'CustomerData', {
            'CustomerData': {
                'user_email': model.email,
                'user_name': model.name,
                'user_phone': String(model.phone)
            }
        });
    }
});

```

```

    } catch(e) {
        console.error(e);
    }
}], "authorization_success": [function (model, eventId) {
    try {

eS('sendEvent', 'CustomerData', {
    'CustomerData': {
        'user_email': model.email,
        'user_name': model.name,
        'user_phone': String(model.phone)
    }
});

    } catch(e) {
        console.error(e);
    }
}], "ajax_page_view": [function (model, eventId) {
    try {
        ga('set', 'page', model);
ga('send', 'pageview');
    } catch(e) {
        console.error(e);
    }
}], function (model, eventId) {
    try {
        gtag('config', 'G-Y63NHZZWEM', {
'page_path': model
});

```



```

    } catch(e) {
        console.error(e);
    }
}], "add_to_wishlist": [function (model, eventId) {
    try {
        fbq('track', 'AddToWishlist', {}, {eventID: eventId});
    } catch(e) {
        console.error(e);
    }
}],
content_copy_protection: false,
GLOBAL_CSRF_TOKEN: 'a26b2612b9a849f757ca047997de3d6a829ca027'
};
w.dataLayer = w.dataLayer || [];
})(window);
</script>
<link
href='/assets/default/production/styles.f1a806360f63c93315c517b3c97ce74c.css'
type='text/css' rel='stylesheet'>
<link
href='/assets/default/production/client.9a4e821fa8c62adc3127d3a2e0b30672.css'
type='text/css' rel='stylesheet'>
<script
src="/globals.js/?version=7cefae267ec40e61a63b469333b22556"></script>
<script src='/assets/cache/horoshop_default_main.js?1718203498'></script>
<link rel="shortcut icon" href="/assets/favicon.ico" type="image/x-icon">
<script>
(function (window, document) {

```

```
'use strict';

var file = '/frontend/themes/horoshop_default/layout/img/icons-default.svg',
    revision = '4a6499ea67c73b6c12f10fc608ba45b7';

var isLocalStorage = 'localStorage' in window && window['localStorage'] !==
null,
    request,
    data,
    insertIT = function () {
        var template = document.createElement('template');

        template.innerHTML = data;
        document.body.prepend(template.content.firstChild);
    },
    insert = function()
    {
        if( document.body ) insertIT();
        else document.addEventListener( 'DOMContentLoaded', insertIT );
    };

if (isLocalStorage && localStorage.getItem('inlineSVGrev') === revision) {
    data = localStorage.getItem('inlineSVGdata');

    if (data) {
        insert();
        return true;
    }
}
```

```

try {
  request = new XMLHttpRequest();
  request.open('GET', file, true);
  request.onload = function () {
    if (request.status >= 200 && request.status < 400) {
      data = request.responseText;
      insert();
      if (isLocalStorage) {
        localStorage.setItem('inlineSVGdata', data);
        localStorage.setItem('inlineSVGrev', revision);
      }
    }
  }
  request.send();
} catch (e) {
}

})(window, document);
</script>    <!-- Google Tag Manager -->
<script>(function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({'gtm.start':
new Date().getTime(),event:'gtm.js'});var f=d.getElementsByTagName(s)[0],
j=d.createElement(s),dl=l!='dataLayer'?'&l='+l:'';j.async=true;j.src=
'https://www.googletagmanager.com/gtm.js?id='+i+dl;f.parentNode.insertBefore(j,f);
})(window,document,'script','dataLayer','GTM-TFFBCBG');</script>
<!-- End Google Tag Manager --><script type="text/javascript">
(function () {})(window.dataLayer = window.dataLayer || []);
var                products                =
[{"id":198760,"article":"SKT000886945","brand_title":"Ladytes","title":"Жіноча
шапка        з        відворотом        56-58        колір        бузковий        ЦБ-
00209583","price":99,"url":"https://fs.ua/catalog/zhinocha-shapka-z-vidvorotom-

```



00253136\\", "category\_string": "Жінкам", "article\_for\_display": "SKT001001018", "in\_stock": true}, {"id": 198425, "article": "SKT000914635", "brand\_title": "KENT&AVER", "title": "Чоловіча бейсболка 56-58 колір бежевий ЦБ-00222258", "price": 199, "url": "https://fs.ua/catalog/cholovicha-beisbolka-56-58-kolir-bezhevyi-tsb-00222258\\", "category\_string": "Жінкам", "article\_for\_display": "SKT000914635", "in\_stock": true}, {"id": 198411, "article": "SKT001003365", "brand\_title": "CREP", "title": "Жіноча футболка з коротким рукавом 48 колір молочний ЦБ-00254126", "price": 399, "url": "https://fs.ua/catalog/zhinocha-futbolka-z-korotkym-rukavom-48-kolir-molochnyi-tsb-00254126\\", "category\_string": "Жінкам", "article\_for\_display": "SKT001003365", "in\_stock": true}, {"id": 198408, "article": "SKT001000473", "brand\_title": null, "title": "Жіноча спідниця 50 колір блакитний ЦБ-00252863", "price": 399, "url": "https://fs.ua/catalog/zhinocha-spidnytsia-50-kolir-blakytnyi-tsb-00252863\\", "category\_string": "Жінкам", "article\_for\_display": "SKT001000473", "in\_stock": true}, {"id": 198403, "article": "SKT001003416", "brand\_title": "CREP", "title": "Жіноча футболка з коротким рукавом 48 колір білий ЦБ-00254144", "price": 349, "url": "https://fs.ua/catalog/zhinocha-futbolka-z-korotkym-rukavom-48-kolir-bilyi-tsb-00254144\\", "category\_string": "Жінкам", "article\_for\_display": "SKT001003416", "in\_stock": true}, {"id": 198401, "article": "SKT001003396", "brand\_title": "Teres", "title": "Жіночі плавки — бразильяно 44 колір чорний ЦБ-00254137", "price": 299, "url": "https://fs.ua/catalog/zhinochi-plavky-brazyliano-44-kolir-chornyi-tsb-00254137\\", "category\_string": "Жінкам", "article\_for\_display": "SKT001003396", "in\_stock": true}, {"id": 198395, "article": "SKT001003390", "brand\_title": "Teres", "title": "Жіночі плавки — бразильяно 44 колір білий ЦБ-00254135", "price": 299, "url": "https://fs.ua/catalog/zhinochi-plavky-brazyliano-44-kolir-bilyi-tsb-00254135\\", "category\_string": "Жінкам", "article\_for\_display": "SKT001003390", "in\_stock": true}



```

00253953\\", "category_string": "Жінкам", "article_for_display": "SKT001003077", "in
_stock": true}, {"id": 198378, "article": "SKT001003074", "brand_title": "No
Brand", "title": "Жіноча сумка колір чорний ЦБ-
00253950", "price": 899, "url": "https://fs.ua/catalog/zhinocha-sumka-kolir-chorny-
tsb-
00253950\\", "category_string": "Жінкам", "article_for_display": "SKT001003074", "in
_stock": true}],
ecProducts = [],
ecProductsPerChunk = 5;
for (var i = 0, l = products.length; i < l; ++i) {
var product = products[i];
ecProducts.push({
  "id": product.article + "",
  "name": product.title + " (артикул: " + product.article_for_display + ")",
  "category": product.category_string,
  "brand": product.brand_title,
  "list": "Catalog Page",
  "position": i + 1
});
}
while (ecProducts.length > 0) {
var chunk = ecProducts.splice(0, ecProductsPerChunk);
dataLayer.push({
  "event": "productImpression",
  "ecommerce": {
    "currencyCode": GLOBAL.currency.iso,
    "detail": {
      "products": chunk
    }
  }
});
}
}

```

```

    });
}
</script><!-- Facebook Pixel Code -->
<script>
!function(f,b,e,v,n,t,s)
{if(f.fbq)return;n=f.fbq=function(){n.callMethod?
n.callMethod.apply(n,arguments):n.queue.push(arguments)};
if(!f._fbq)f._fbq=n;n.push=n;n.loaded=!0;n.version='2.0';n.agent='plhoroshop';
n.queue=[];t=b.createElement(e);t.async=!0;
t.src=v;s=b.getElementsByTagName(e)[0];
s.parentNode.insertBefore(t,s)}(window, document,"script",
"https://connect.facebook.net/en_US/fbevents.js");
fbq('init', "563531781057175");
fbq('track', 'PageView');
</script>
<noscript></noscript>
<!-- End Facebook Pixel Code --><!-- Global site tag (gtag.js) - Google Analytics -->
<script      async      src='https://www.googletagmanager.com/gtag/js?id=G-
Y63NHZZWEM'></script>
<script>
window.dataLayer = window.dataLayer || [];
function gtag(){dataLayer.push(arguments);}
gtag('js', new Date());

gtag('config', 'G-Y63NHZZWEM');
gtag('config', 'AW-924462988');

```



```

</script><script>var items = [],
  products
[{"id":198760,"article":"SKT000886945","brand_title":"Ladytes","title":"Жіноча
шапка з відворотом 56-58 колір бузковий ЦБ-
00209583","price":99,"url":"https://fs.ua/catalog/zhinocha-shapka-z-vidvorotom-
56-58-kolir-buzkovyi-tsb-
00209583/","category_string":"Жінкам","article_for_display":"SKT000886945","in
_stock":true},{ "id":198649,"article":"SKT001001313","brand_title":"Skyper","title":
"Жіночий ремінь колір бежевий ЦБ-
00253228","price":299,"url":"https://fs.ua/catalog/zhinochyi-remin-kolir-
bezhevyi-tsb-
00253228/","category_string":"Жінкам","article_for_display":"SKT001001313","in
_stock":true},{ "id":198648,"article":"SKT001001308","brand_title":"Skyper","title":
"Жіночий ремінь колір бежевий ЦБ-
00253223","price":99,"url":"https://fs.ua/catalog/zhinochyi-remin-kolir-bezhevyi-
tsb-
00253223/","category_string":"Жінкам","article_for_display":"SKT001001308","in
_stock":true},{ "id":198647,"article":"SKT001001307","brand_title":"Skyper","title":
"Жіночий ремінь колір білий ЦБ-
00253222","price":99,"url":"https://fs.ua/catalog/zhinochyi-remin-kolir-bilyi-tsb-
00253222/","category_string":"Жінкам","article_for_display":"SKT001001307","in
_stock":true},{ "id":198646,"article":"SKT001001305","brand_title":"Skyper","title":
"Жіночий ремінь колір бежевий ЦБ-
00253220","price":99,"url":"https://fs.ua/catalog/zhinochyi-remin-kolir-bezhevyi-
tsb-
00253220/","category_string":"Жінкам","article_for_display":"SKT001001305","in
_stock":true},{ "id":198643,"article":"SKT001001019","brand_title":"fs","title":"Жін
оча сумка колір чорний ЦБ-
00253137","price":999,"url":"https://fs.ua/catalog/zhinocha-sumka-kolir-chorny-
tsb-

```

00253137\", \"category\_string\": \"Жінкам\", \"article\_for\_display\": \"SKT001001019\", \"in\_stock\": true}, {\"id\": 198642, \"article\": \"SKT001001018\", \"brand\_title\": \"fs\", \"title\": \"Жіноча сумка колір молочний ЦБ-00253136\", \"price\": 999, \"url\": \"https://fs.ua/catalog/zhinocha-sumka-kolir-molochnyi-tsb-00253136\", \"category\_string\": \"Жінкам\", \"article\_for\_display\": \"SKT001001018\", \"in\_stock\": true}, {\"id\": 198425, \"article\": \"SKT000914635\", \"brand\_title\": \"KENT&AVER\", \"title\": \"Чоловіча бейсболка 56-58 колір бежевий ЦБ-00222258\", \"price\": 199, \"url\": \"https://fs.ua/catalog/cholovicha-beisbolka-56-58-kolir-bezhevyi-tsb-00222258\", \"category\_string\": \"Жінкам\", \"article\_for\_display\": \"SKT000914635\", \"in\_stock\": true}, {\"id\": 198411, \"article\": \"SKT001003365\", \"brand\_title\": \"CREP\", \"title\": \"Жіноча футболка з коротким рукавом 48 колір молочний ЦБ-00254126\", \"price\": 399, \"url\": \"https://fs.ua/catalog/zhinocha-futbolka-z-korotkym-rukavom-48-kolir-molochnyi-tsb-00254126\", \"category\_string\": \"Жінкам\", \"article\_for\_display\": \"SKT001003365\", \"in\_stock\": true}, {\"id\": 198408, \"article\": \"SKT001000473\", \"brand\_title\": null, \"title\": \"Жіноча спідниця 50 колір блакитний ЦБ-00252863\", \"price\": 399, \"url\": \"https://fs.ua/catalog/zhinocha-spidnytsia-50-kolir-blakytnyi-tsb-00252863\", \"category\_string\": \"Жінкам\", \"article\_for\_display\": \"SKT001000473\", \"in\_stock\": true}, {\"id\": 198403, \"article\": \"SKT001003416\", \"brand\_title\": \"CREP\", \"title\": \"Жіноча футболка з коротким рукавом 48 колір білий ЦБ-00254144\", \"price\": 349, \"url\": \"https://fs.ua/catalog/zhinocha-futbolka-z-korotkym-rukavom-48-kolir-bilyi-tsb-00254144\", \"category\_string\": \"Жінкам\", \"article\_for\_display\": \"SKT001003416\", \"in\_stock\": true}, {\"id\": 198401, \"article\": \"SKT001003396\", \"brand\_title\": \"Teres\", \"title\": \"Жіночі плавки — бразильяно 44 колір чорний ЦБ-00254137\", \"price\": 299, \"url\": \"https://fs.ua/catalog/zhinochi-plavky-brazyliano-44-kolir-chorny-tsb-

00254137\\", "category\_string": "Жінкам", "article\_for\_display": "SKT001003396", "in\_stock": true}, {"id": 198395, "article": "SKT001003390", "brand\_title": "Teres", "title": "Жіночі плавки — бразильяно 44 колір білий ЦБ-00254135", "price": 299, "url": "https://fs.ua/catalog/zhinochi-plavky-brazyliano-44-kolir-bilyi-tsb-00254135\\", "category\_string": "Жінкам", "article\_for\_display": "SKT001003390", "in\_stock": true}, {"id": 198392, "article": "SKT001003387", "brand\_title": "Teres", "title": "Жіночий ліф — балконет 44 колір м'ятний ЦБ-00254134", "price": 449, "url": "https://fs.ua/catalog/zhinochyi-lif-balkonet-44-kolir-miatnyi-tsb-00254134\\", "category\_string": "Жінкам", "article\_for\_display": "SKT001003387", "in\_stock": true}, {"id": 198389, "article": "SKT001003378", "brand\_title": "Teres", "title": "Жіночі плавки — бразильяно 44 колір білий ЦБ-00254131", "price": 299, "url": "https://fs.ua/catalog/zhinochi-plavky-brazyliano-44-kolir-bilyi-tsb-00254131\\", "category\_string": "Жінкам", "article\_for\_display": "SKT001003378", "in\_stock": true}, {"id": 198386, "article": "SKT001003375", "brand\_title": "Teres", "title": "Жіночий ліф — балконет 44 колір хакі ЦБ-00254130", "price": 449, "url": "https://fs.ua/catalog/zhinochyi-lif-balkonet-44-kolir-khaki-tsb-00254130\\", "category\_string": "Жінкам", "article\_for\_display": "SKT001003375", "in\_stock": true}, {"id": 198381, "article": "SKT001003083", "brand\_title": "No Brand", "title": "Жіноча сумка колір молочний ЦБ-00253959", "price": 699, "url": "https://fs.ua/catalog/zhinocha-sumka-kolir-molochnyi-tsb-00253959\\", "category\_string": "Жінкам", "article\_for\_display": "SKT001003083", "in\_stock": true}, {"id": 198380, "article": "SKT001003082", "brand\_title": "No Brand", "title": "Жіноча сумка колір чорний ЦБ-00253958", "price": 699, "url": "https://fs.ua/catalog/zhinocha-sumka-kolir-chornyi-tsb-00253958\\", "category\_string": "Жінкам", "article\_for\_display": "SKT001003082", "in\_stock": true}

```

00253958\\", "category_string": "Жінкам", "article_for_display": "SKT001003082", "in
_stock": true}, {"id": 198379, "article": "SKT001003077", "brand_title": "No
Brand", "title": "Жіноча сумка колір чорний ЦБ-
00253953", "price": 799, "url": "https://fs.ua/catalog/zhinocha-sumka-kolir-chorny-
tsb-
00253953\\", "category_string": "Жінкам", "article_for_display": "SKT001003077", "in
_stock": true}, {"id": 198378, "article": "SKT001003074", "brand_title": "No
Brand", "title": "Жіноча сумка колір чорний ЦБ-
00253950", "price": 899, "url": "https://fs.ua/catalog/zhinocha-sumka-kolir-chorny-
tsb-
00253950\\", "category_string": "Жінкам", "article_for_display": "SKT001003074", "in
_stock": true}],
ids = [],
total = 0,
productsPerEvent = 5,

sendEvent = function(items, ids, total) {
    gtag('event', 'view_item_list', {
        'item_list_name': 'Catalog Page',
        'items': items,
        'ecomm_prodid': ids,
        'ecomm_pagetype': 'category',
        'ecomm_totalvalue': total
    });
};

for (var i = 0, l = products.length; i < l; ++i) {
    var product = products[i];

    items.push({

```

```

    'item_id': product.article + ",
    'item_name': product.title,
    'item_category': product.category_string,
    'item_brand': product.brand_title,
    'item_list_id': i + 1,
    'price': product.price
  });

  ids.push(product.article + "");
  total = total + product.price;

  if (items.length === productsPerEvent) {
    sendEvent(items, ids, total);

    items = [];
    ids = [];
    total = 0;
  }
}

if (items.length > 0) {
  sendEvent(items, ids, total);
}</script> <script type="text/javascript">
(function(d, w, s) {
  var widgetHash = 'o3KyFo7RkSDnYnA811GR', bch = d.createElement(s);
  bch.type = 'text/javascript'; bch.async = true;
  bch.src = '//widgets.binotel.com/chat/widgets/' + widgetHash + '.js';
  var sn = d.getElementsByTagName(s)[0]; sn.parentNode.insertBefore(bch, sn);
})(document, window, 'script');
</script>

```

```

<script>
(function(d) {
var s = d.createElement('script');
s.defer = true;
if (document.documentElement.lang === "uk") {
s.src = 'https://multisearch.io/plugin/12042';
} else if (document.documentElement.lang === "ru") {s.src =
'https://multisearch.io/plugin/12232';
}
if (d.head) d.head.appendChild(s);
})(document);
</script> </head>
<body class=" ua-UA" itemscope itemtype="https://schema.org/WebPage">
<!-- Google Tag Manager (noscript) -->
<noscript><iframe src="https://www.googletagmanager.com/ns.html?id=GTM-
TFFBCBG"
height="0" width="0" style="display:none;visibility:hidden"></iframe></noscript>
<!-- End Google Tag Manager (noscript) --><script><meta name="it-rating"
content="it-rat-87b46bd67e261b9528f8223b42deb17e" /> </script>
<div class="session-messages" id="j-sm" style="display:none;"></div>
<div class="container">
<div class="header">
<div class="header__container">
<div class="header__middle">
<div class="header__wrapper">
<div class="header__layout header__layout--middle">
<div class="header__column header__column--left ">
<div class="header__section">
<div class="header__logo header__logo--fixed">
<a class="header__logo-link" href="/>

```

```

    <img alt='fs (сімейний шопінг) - Shket&Faina сімейний шопінг для
всієї родини' class='header-logo-img' width='200' height='100'
src='/content/images/2/200x100190nn0/32244545744550.webp'
    srcset="
    /content/images/2/200x100190nn0/32244545744550.webp 1x,
    /content/images/2/400x200150nn0/32244545744550.webp 2x
    ">
</div></div>
<div class="header__column
header__column--wide header__column--vertical">
    <div class="header__row">
        <div class="header__section">
</div>
        <div class="header__row">
            <div class="header__column">
                <div class="header__section
header__section--search">
                    <div class="search j-search">
                        <form method="get" data-action="/katalog/search/">
                            <button type="submit" class="search__button" disabled>
                                <svg class="icon icon--search"><use xlink:href="#icon-
search"></use></svg>
                            </button>
                            <input class="search__input"
id="search_uidf2cfaa30a312ad580d320ece80825fa6" type="text" name="q"
placeholder="пошук товарів" autocomplete="off" value="">
                        </form>
                    </div>
                    <div class="search__results" id="search_uidf2cfaa30a312ad580d320ece80825fa6-
search-results"></div>

```

```

<script>
(function (w) {
  w.INIT.add(function() {
    init_search_widget('#search_uidf2cfaa30a312ad580d320ece80825fa6', {
      hideQuery: true
    });
  }, w.init_search_widget)
})(window);
</script></div>
</div>
</div>
<div class="header__column
header__column--right ">
  <div class="header__section">
    <div class="timetable">
      <svg class="icon icon--clock"><use xlink:href="#icon-clock"></use></svg>
      <div class="timetable__title">Графік роботи:</div>

```