

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Сумський державний університет**  
Факультет електроніки та інформаційних технологій  
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

\_\_\_\_\_ Ігор ШЕЛЕХОВ  
(підпис)

\_\_\_\_\_ травня 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА**  
**на здобуття освітнього ступеня магістр**

зі спеціальності 122 - Комп'ютерних наук,  
освітньо-наукової програми «Інформатика»  
на тему: «Інформаційна вебсистема з роздрібною торгівлі взуттям»  
здобувача групи ІН-06-2 Вербицького Олександра Романовича

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

Олександр  
ВЕРБИЦЬКИЙ

\_\_\_\_\_ (підпис)

Керівник,  
старший викладач комп'ютерних  
наук, к.т.н.  
Берест О.Б.

Олег БЕРЕСТ

\_\_\_\_\_ (підпис)

**Суми – 2024**

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

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

\_\_\_\_\_ (підпис)

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

#### на здобуття освітнього ступеня магістр

зі спеціальності 122 - Комп'ютерних наук, освітньо-наукової програми «Інформатика»

здобувача групи ІН-06-2 Вербицького Олександра Романовича

1. Тема роботи: «Інформаційна вебсистема з роздрібною торгівлі взуттям»

затверджую                      наказом                      по                      СумДУ                      від \_\_\_\_\_ № \_\_\_\_\_

2. Термін                      здачі                      здобувачем                      кваліфікаційної                      роботи                      до \_\_\_\_\_

3. Вхідні                      дані                      до                      кваліфікаційної                      роботи

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

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

*2) Огляд технологій, що використовуються для розробки вебсистеми з роздрібною торгівлі взуттям. 3) Розробка вебсистеми з роздрібною торгівлі взуттям. 4) Аналіз результатів.*

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

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

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

Завдання прийняв до виконання \_\_\_\_\_  
 (підпис)

Керівник \_\_\_\_\_  
 (підпис)

**КАЛЕНДАРНИЙ ПЛАН**

№ п/ п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, постановка й формування завдань дослідження</i>		
2	<i>Огляд технологій, що використовуються для розробки вебсистеми з роздрібною торгівлі взуттям</i>		
3	<i>Розробка вебсистеми з роздрібною торгівлі взуттям</i>		
4	<i>Аналіз отриманих результатів</i>		
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>		

Здобувач вищої освіти

\_\_\_\_\_  
(підпис)

Керівник

\_\_\_\_\_  
(підпис)

## АНОТАЦІЯ

**Записка:** 66 стор., 48 рис., 4 додатки, 23 джерел.

**Обґрунтування актуальності теми роботи** – тема дослідження даної кваліфікаційної роботи є актуальною, оскільки належить до інформаційно-технологічної сфери. Розвиток вебсистем для роздрібно́ї торгівлі взуттям стає необхідним у зв'язку з постійним зростанням обсягів даних та потреби у швидкому та ефективному доступі до інформації. Використання автоматизованих інформаційних систем для управління роздрібною торгівлею є ключовим фактором у забезпеченні конкурентоспроможності сучасних підприємств та організацій.

**Об'єкт дослідження** – процес автоматизації розгортання інформаційних систем для роздрібно́ї торгівлі взуттям.

**Мета роботи** – розробка автоматизованої вебсистеми для роздрібно́ї торгівлі взуттям, спрямованої на оптимізацію процесів розгортання та управління додатками у вебсередовищах.

**Методи дослідження** – аналіз існуючих платформ для роздрібно́ї торгівлі, вивчення технологій автоматизованого розгортання вебдодатків, огляд технологій шаблонізації вебдодатків та автоматизації розгортання.

**Результати** – розроблено вебсистему, яка дозволяє ефективно використовувати сучасні вебтехнології для управління роздрібною торгівлею взуттям. Система пройшла успішне тестування та демонструє здатність до швидкої та надійної роботи вебдодатку у вебсередовищі.

**ІНФОРМАЦІЙНА СИСТЕМА, APRIORI, MYSQL, NODEJS, PHP,  
AJAX, JAVASCRIPT, HTML, CSS, JQUERY BOOTSTRAP, РОЗДРІБНА  
ТОРГІВЛЯ, ВЗУТТЯ.**

## Зміст

ВСТУП	6
1 АНАЛІТИЧНИЙ ОГЛЯД	8
1.1 Аналіз основних принципів, що лежать в основі Інтернет магазинів	8
1.2 Визначення середовища розробки	9
1.3 Постановка задачі	11
2 ВИБІР МЕТОДІВ РОЗВ'ЯЗАННЯ ЗАДАЧІ	13
2.1 Огляд сучасних інструментів для розробки баз даних	13
2.2 Огляд сучасних інструментів для клієнтської частини вебдодатку	20
2.3 Огляд сучасних інструментів для серверної частини вебдодатку	23
2.4 Структура розробленої інформаційної системи	26
2.5 Проектування бази даних	27
2.6 Алгоритм рекомендацій	30
ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	33
3.1 Опис програмної реалізації	33
3.2 Аналіз результатів клієнтської частини вебсистеми	35
3.3 Аналіз результатів адмін частини вебсистеми	52
ВИСНОВКИ	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	58

## ВСТУП

З моменту першої науково-технічної революції людство обрало шлях технічної еволюції та розвитку. У час постіндустріального розвитку, людство потребує переходу від аналогових видів збереження, обробки та передачі інформації до цифрових способів, таких як E-mail, електронні документи, та інші оцифровані способи збереження даних. Уся інформація та папери різних видів та розмірів компаній переходять на цифрові методи роботи з даними, навіть така неповоротка система як держава теж починає переходити у цифровий режим.

Розвиток оцифрованої інформації дав поштовх для створення та розміщення цієї інформації в глобальній мережі Інтернет, за для швидкої та зручної передачі цих самих даних.

У наш час за допомогою розвитку телекомунікаційних технологій можна працювати майже з будь якого місця світу за допомогою не тільки вежі зв'язку які з технічних проблем не можуть бути встановлені усюди, а й супутникового Інтернету Starlink, що дає змогу підключитися до Інтернету навіть у самих складних умовах. За допомогою всесвітньої мережі людство отримало до цього небачений рівень мобільності, що дає змогу кожній людині світу великі можливості для розвитку.. Кожна людина світу може знайти роботу дистанційно та працювати на цікаві та прибуткові проекти навіть інших країнах світу не виходячи з дому.

Вебсайти стали частиною нашого життя та зробили його ще простішим. Більшість сервісів перейшла з додатків на онлайн сервіси. Що дає змогу працювати з ними з будь якої точки світу та не бути прив'язаним до конкретного місця си девайсу на якому встановлений додатки, що в десятки разів збільшує продуктивність працівників та користь для підприємців.

Всесвітня мережа Інтернет здатна забезпечити людство майже необмеженою кількістю інформації та можливостей, за допомогою Інтернету з'явилася можливість як дистанційного навчання так й роботи що дає змогу як

звичайному працівнику так й роботодавцю обрати найбільш підходящого кандидата

## 1 АНАЛІТИЧНИЙ ОГЛЯД

### 1.1 Аналіз основних принципів, що лежать в основі Інтернет магазинів

Процес проектування програмного додатку - це ключовий етап у створенні будь-якого інформаційного проекту, де прийняття рішень має далекосяжні наслідки для успішного функціонування системи у майбутньому. Одним із найважливіших рішень, які необхідно зробити на цьому етапі, є вибір архітектури програми. Вибір правильної архітектури визначає, наскільки добре програма відповідає вимогам користувача, як вона працює в реальному середовищі та як вона масштабується та підтримується в майбутньому.

Архітектура [1] програми впливає на багато аспектів розробки та функціонування, такі як гнучкість, масштабованість, надійність та продуктивність. Вибір неправильної архітектури може призвести до великих проблем та витрат, які буде складно виправити в майбутньому.

У світі інформаційних технологій клієнт-серверна [2] архітектура визнана однією з найбільш поширених та ефективних архітектур для розробки різноманітних програмних додатків.

Клієнт-серверна архітектура - це спосіб побудови програмного забезпечення, при якому існує дві основні компоненти: клієнт і сервер. Клієнт [3] - це програма, що виконується на більшості кінцевих пристроїв, таких як комп'ютери, мобільні телефони, планшети і т. д. Сервер [4] - це програма, що запускається на потужних серверних системах і надає певні послуги або ресурси клієнту.

Крім клієнт-серверної архітектури існують інші типи архітектур, такі як:

- Монолітна архітектура [5]
- Розподілена архітектура [6]
- Централізована архітектура [7]



Порівняння різних тип архітектур:

Гнучкість та масштабованість. Клієнт-серверна архітектура та розподілена архітектура надають більшу гнучкість та масштабованість, оскільки програма розділена на окремі компоненти, які можуть бути розгорнуті на різних серверах або пристроях. Це дозволяє легко додавати нові функції та розгортати їх на додаткових серверах, якщо потрібно.

Монолітна архітектура та централізована архітектура менш гнучкі та складно масштабовані. У монолітній архітектурі будь-яка зміна вимагає перекомпіляції всієї програми, а централізована архітектура може стати обмеженішою при збільшенні навантаження на центральний сервер.

Надійність та доступність. Клієнт-серверна архітектура та розподілена архітектура можуть бути більш надійними та доступними завдяки можливості розділення завдань і обробки помилок на різних серверах. Якщо один сервер виходить з ладу, інші сервери можуть продовжувати працювати. Монолітна архітектура та централізована архітектура можуть бути менш надійними, оскільки вони залежать від одного сервера. Якщо цей сервер виходить з ладу, то вся програма може стати недоступною.

Простота управління та розробки. Монолітна архітектура та централізована архітектура можуть бути більш простими у використанні та розробці, оскільки всі компоненти програми знаходяться на одному сервері або в одному модулі. Клієнт-серверна архітектура та розподілена архітектура можуть бути складнішими у використанні та розробці через необхідність керування різними компонентами та серверами, але є дуже поширеною та необхідною для якісної розробки. Тому саме клієнт-серверна архітектура була обрана для створення Інтернет магазину

## **1.2 Визначення середовища розробки**

Існує багато видів програмного забезпечення для створення сайтів, кожне з яких має свої переваги та недоліки. Одними з них є:

- Sublime Text [8] – редактор коду для багатьох мов програмування, має різні макети підсвітлення фрагментів коду. Головною перевагою є автозбереження файлів.

- Atom [9] – редактор коду розроблений компанією GitHub, за допомогою двигуна рендерингу Chrome дає можливість розробляти додатки з використання вебтехнологій. Головною перевагою є широкі можливості для індивідуального налаштування.

- PhpStorm [10] – середовище розробки для мови програмування PHP. Має велику кількість переваг.

Розробка Інтернет магазину взуття буде здійснюватися за допомогою програмного забезпечення Visual Studio Code [11]. Дана програма була вибрана через ряд переваг.

- Кросс-платформенність допомагає працювати з кодом на будь якій з доступних операційних системах.

- Visual Studio Code має підтримку великої кількості найпопулярніших мов програмування, що дозволяє створювати код на різних мовах програмування. Додаток самостійно може визначити мову програмування, що є зручним при одночасній роботі з великою кількістю мов, також є можливість зміни розширення файлу.

- Налаштування є вбудованою функціональністю додатка та допомагає пришвидшити пошук та виправлення помилок у коді.

- Наявність інтеграції з Git [12], що надає розробнику всі переваги розробки з Git.

- IntelliSense [13] ще одна функція яка допомагає розробнику зі швидким написання коду, вона пропонує найбільш підходящі варіанти завершення коду, що в разі збільшує швидкість написання готового коду.

Функція працює для найбільш популярних мов програмування, та є можливість встановити розширення для всіх інших мов.

Також існує можливість роботи з Visual Studio Code в браузері, за допомогою Codespaces. Що надає всі переваги розробки в браузері.

Visual Studio Code є найбільш підходящим програмним забезпеченням для створення сучасних та функціональних вебсайтів, які використовують велику кількість додаткових мов програмування.

### **1.3 Постановка задачі**

Метою роботи є розробка вебсистеми для роздрібної торгівлі взуттям. Дане дослідження виконується в рамках кваліфікаційної роботи бакалавра за спеціальністю 122 «Комп'ютерні науки».

Розроблений програмний продукт має задовольняти наступні вимоги:

- каталогізація продукції з можливістю фільтрації за різними критеріями (категорія, бренд);
- обробка замовлень;
- алгоритм рекомендацій;
- адмін панель
- адаптивний дизайн для коректного відображення на різних пристроях.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- виконати аналіз проблемної області та визначити актуальність;
- провести порівняння існуючих систем для роздрібної торгівлі взуттям та аналіз аналогів програмних продуктів;
- обрати технології для розробки програмного забезпечення;
- реалізувати модель інформаційної системи.

Предметом дослідження є методологія розробки вебсистеми для роздрібної торгівлі взуттям з використанням сучасних технологій веброботки.

Наукова новизна полягає в тому описане у даній роботі програмне рішення дозволить інтегрувати персоналізовану систему рекомендацій для користувача.

Практичною значимістю буде те, що застосування розробленої вебсистеми дозволить зменшити час на управління товарним асортиментом, автоматизувати процес обробки замовлень та підвищити зручність користувачів, що в результаті сприятиме збільшенню обсягів продажів та задоволеності клієнтів.

## 2 ВИБІР МЕТОДІВ РОЗВ'ЯЗАННЯ ЗАДАЧІ

### 2.1 Огляд сучасних інструментів для розробки баз даних

Бази даних [14] (БД) відіграють надзвичайно важливу роль у сучасних інформаційних системах. Вони є основою для зберігання та управління великим обсягом даних, що використовуються в різних галузях, від бізнесу та науки до соціальних мереж та розваг.

Ключові аспекти, що пояснюють важливість баз даних в інформаційних системах:

- Зберігання та організація даних: Бази даних дозволяють ефективно зберігати великі обсяги даних та організувати їх у логічні структури. Це допомагає в структуруванні інформації, полегшує пошук та доступ до потрібних даних.
- Швидкий доступ та обробка: За допомогою баз даних можна ефективно отримувати доступ до даних за допомогою запитів. Це особливо важливо в сучасному світі, де швидкість інформаційного обміну має велике значення.
- Забезпечення цілісності та безпеки: Бази даних надають механізми для забезпечення цілісності даних та контролю доступу до них. Це допомагає у попередженні втрати даних, несанкціонованого доступу та забезпечує захист інформації.
- Спільний доступ до даних: В інформаційних системах, де багато користувачів працюють з однією базою даних, важливо мати механізми контролю версій даних та управління конфліктами.

- Аналіз та підтримка прийняття рішень: Бази даних надають можливість аналізу та видобутку інформації, що допомагає в управлінні бізнес-процесами та прийнятті обґрунтованих рішень.

Нині існує багато інструментів для розробки баз даних, і кожен з них має свої особливості та переваги. Розглянемо деякі з найпопулярніших систем управління базами даних (СУБД): PostgreSQL [15], MySQL [16] та Oracle [17].

PostgreSQL - це одна з найпопулярніших і впливових відкритих реляційних систем управління базами даних з численними перевагами, які роблять її привабливим вибором для багатьох проектів.

#### Переваги PostgreSQL:

- Вільне та відкрите програмне забезпечення: PostgreSQL розповсюджується під ліцензією PostgreSQL, яка дозволяє використовувати, змінювати і поширювати СУБД безкоштовно. Це робить PostgreSQL доступним для різних проектів і організацій, які бажають використовувати вільне програмне забезпечення.

- Надійність та стійкість: PostgreSQL славиться своєю надійністю та стійкістю. Він має вбудовану підтримку транзакцій, яка дозволяє забезпечувати цілісність даних та відновлювати базу даних після непередбачуваних ситуацій, таких як аварії системи.

- Розширюваність: PostgreSQL дозволяє розширювати функціональність за допомогою розширень та власних функцій. Це означає, що ви можете налаштовувати СУБД під ваші конкретні потреби та додавати нові функції відповідно до ваших вимог.

- Підтримка розширених типів даних: PostgreSQL підтримує багато різних типів даних, включаючи геодані, текстовий пошук, JSON, XML та багато інших. Це дозволяє працювати з різноманітними видами інформації.

– Активна спільнота користувачів та розробників: PostgreSQL має велику та активну спільноту користувачів і розробників, яка надає підтримку та постійно розвиває СУБД.

Хоча PostgreSQL має численні переваги, які роблять його популярним вибором для багатьох проектів, він також має деякі недоліки та обмеження, які важливо враховувати при виборі СУБД.

Недоліки PostgreSQL:

– Споживана пам'ять: PostgreSQL може споживати значну кількість оперативної пам'яті, особливо при обробці великих обсягів даних або складних запитів. Це може призвести до необхідності докладно налаштовувати ресурси сервера.

– Складність налаштування: PostgreSQL має багато параметрів конфігурації, і для досягнення оптимальної продуктивності може бути необхідним глибоке розуміння цих параметрів та їх впливу на систему. Неналежне налаштування може призвести до проблем з продуктивністю.

– Відсутність вбудованих реплікаційних рішень: Порівняно з деякими іншими СУБД, PostgreSQL не має вбудованих реплікаційних рішень. Реплікація даних (створення копій бази даних) може бути складною для налаштування та підтримки.

– Швидкість: У деяких випадках PostgreSQL може бути менш продуктивним у порівнянні з іншими СУБД, особливо при великому навантаженні. Належне налаштування та оптимізація може допомогти збільшити продуктивність, але це може вимагати ресурсів.

MySQL - це одна з найпопулярніших систем управління базами даних (СУБД) у світі. Вона має безліч переваг та використовується у різних проектах, від вебдодатків до підприємницьких систем.

### Переваги MySQL:

- Вільне та відкрите програмне забезпечення: MySQL розповсюджується під ліцензією GPL, що дозволяє використовувати, змінювати та розповсюджувати її безкоштовно. Це робить її доступною для багатьох користувачів та проектів.
- Простота встановлення та налаштування: MySQL славиться своєю простотою встановлення і налаштування. Вона швидко запускається та працює на багатьох платформах.
- Швидкодія: MySQL володіє високою продуктивністю та швидкістю обробки запитів. Вона ефективно впорається з великим обсягом даних та завдань.
- Підтримка різних типів даних: MySQL підтримує різноманітні типи даних, включаючи текст, числа, зображення та багато інших. Це робить її придатною для різних видів проектів.
- Реплікація та кластеризація: MySQL має можливості реплікації, які дозволяють створювати копії бази даних для забезпечення надійності та збільшення доступності. Крім того, вона підтримує кластеризацію для розподіленого зберігання даних.
- Активна спільнота користувачів та розробників: MySQL має велику спільноту користувачів та розробників, що надає підтримку та розвиває СУБД.
- Комерційна підтримка: Окрім вільної версії MySQL, існує комерційна версія, яку підтримує Oracle Corporation та яка надає розширені функції та підтримку для корпоративних клієнтів.

Незважаючи на багато переваг, MySQL також має свої недоліки такі, як:



- Відсутність деяких розширених функцій: У порівнянні з іншими СУБД, MySQL може відсутні деякі розширені функції, такі як віконні функції, аналітичні функції, та деякі можливості оптимізації запитів.

- Обмежена підтримка сторонніх розширень: Хоча MySQL підтримує розширення, деякі з них можуть бути менш стабільними та підтримуваними, порівняно з іншими СУБД.

Oracle Database, це одна з найбільш впливових комерційних систем управління базами даних (СУБД) у світі. Вона розроблена та підтримується корпорацією Oracle і використовується в різних галузях, включаючи корпоративні рішення, банківський сектор, виробництво, логістику, бізнес-інтелект і багато інших.

Переваги Oracle:

- Висока продуктивність: Oracle відома своєю високою продуктивністю та швидкодією. Вона може обробляти великі обсяги даних та складні операції швидко та ефективно.

- Висока надійність та стійкість: Oracle Database має вбудовані механізми для забезпечення надійності та стійкості даних. Вона підтримує функції резервного копіювання, відновлення даних та виправлення помилок.

- Масштабованість: Oracle підтримує масштабованість, що дозволяє розширювати систему на великі обсяги даних та високий обсяг транзакцій.

- Багатофункціональність: Oracle пропонує широкий спектр функцій, включаючи підтримку геоданих, багатофакторну аутентифікацію, шифрування, аналітичні засоби, бізнес-інтелект та багато інших.

- Розширена підтримка транзакцій: Oracle надає розширену підтримку транзакцій та вищого рівня ізольованості, що важливо для фінансових застосунків.

- Комерційна підтримка: Oracle надає підтримку для своєї продукції, включаючи патчі безпеки та оновлення. Це важливо для підприємств, які прагнуть надійності та безпеки.

Незважаючи на численні переваги, Oracle також має свої недоліки:

- Високі витрати: Ліцензії та підтримка Oracle відомі своєю високою ціною. Це може бути фінансово вимогливим для багатьох організацій, особливо для стартапів і невеликих підприємств.

- Складність налаштування та адміністрування: Oracle вимагає високої кваліфікації та досвіду для налаштування та адміністрування. Вона має багато параметрів конфігурації, які можуть бути важкими для налаштування та оптимізації.

- Великі вимоги до ресурсів: Oracle може вимагати значних ресурсів, включаючи потужне апаратне забезпечення та дорогі сховища даних.

- Обмежена підтримка відкритих стандартів: Oracle недостатньо підтримує відкриті стандарти, що може ускладнити інтеграцію з іншими системами та застосунками.

- Відсутність відкритості для спільноти: Oracle - це комерційна продукція, і вона не має відкритого вихідного коду. Це означає, що спільнота розробників не може вносити зміни до коду або розвивати продукт незалежно.

– Залежність від постачальника: Організації, що використовують Oracle, стають залежними від постачальника, і це може створити проблеми у випадку зміни вендора або проблем з підтримкою.

Порівнюючи СУБД PostgreSQL, MySQL і Oracle, було обрано MySQL через ряд переваг, таких як:

– Вартість: MySQL є безкоштовною та відкритою системою управління базами даних, що робить її досить доступною для багатьох проектів та організацій. У порівнянні з Oracle, де ліцензії та підтримка можуть бути дуже високими витратами, MySQL - це бюджетно-ефективний вибір.

– Простота використання: MySQL славиться своєю простотою встановлення та налаштування. Вона швидко запускається та не вимагає глибоких знань для початку роботи з нею.

– Загальна продуктивність: MySQL має досить високу продуктивність та швидкодію обробки запитів. Вона може бути використана для широкого спектру проектів, включаючи вебдодатки, блоги, сайти електронної комерції та багато інших.

– Масштабованість: Хоча MySQL може бути використана для менших проектів, вона також має можливості масштабування для обробки великих обсягів даних та великого навантаження.

– Активна спільнота користувачів та розробників: MySQL має велику та активну спільноту користувачів та розробників, які надають безкоштовну підтримку та постійно розвивають продукт.

- Реплікація та висока доступність: MySQL підтримує реплікацію, яка дозволяє створювати копії бази даних для забезпечення надійності та високої доступності.

## 2.2 Огляд сучасних інструментів для клієнтської частини вебдодатку

Вибір технологій для клієнтської частини вебдодатку є ключовим завданням при розробці вебпроектів. Сучасні інструменти для клієнтської частини включають в себе HTML5 [18], CSS3 [19], Bootstrap [20] і JavaScript [21]. Кожна з цих технологій виконує важливі ролі в створенні вебдодатків і має свої позитивні та негативні сторони.

### HTML5

HTML5 (HyperText Markup Language 5) - це п'ята версія стандарту HTML, яка визначає структуру вебсторінок.

#### Позитивні сторони:

- Покращена семантика: HTML5 має багато нових тегів, які дозволяють більш точно описувати структуру сторінки, такі як `<header>`, `<footer>`, `<nav>`, і інші. Це сприяє покращенню SEO і доступності.
- Підтримка мультимедіа: HTML5 додає підтримку аудіо та відео без необхідності використання сторонніх плагінів, таких як Adobe Flash. Це полегшує відтворення відео та аудіо на вебсторінці.
- Локальне сховище: HTML5 пропонує можливості для локального сховища даних на стороні клієнта, що дозволяє покращити швидкість завантаження сторінки та забезпечувати роботу офлайн.

- Розробка для мобільних пристроїв: HTML5 робить розробку мобільних додатків більш зручною завдяки підтримці адаптивного дизайну та мультитач-жестів.

Негативні сторони:

- Не всі функції підтримуються в старих браузерах: Деякі нові функції HTML5 можуть не працювати в старих версіях браузерів, що може призвести до несумісності.

- Специфікація ще розвивається: HTML5 ще не завершено, і специфікація постійно оновлюється. Це може створювати нестабільність у розробці.

CSS3

CSS3 (Cascading Style Sheets 3) - це остання версія мови CSS, яка використовується для опису вигляду і форматування вебсторінок.

Позитивні сторони:

- Збагачений дизайн: CSS3 додає багато нових функцій, таких як округлі кути, тінь, градієнти, перетинання та анімація. Це дозволяє розробникам створювати більш привабливий та інтерактивний дизайн.

- Медіа-запити: CSS3 включає підтримку медіа-запитів, які дозволяють адаптувати вебсторінку до різних розмірів екранів і пристроїв. Це важливо для створення адаптивного дизайну.

- Анімація: CSS3 дозволяє створювати анімацію без використання JavaScript, що спрощує створення візуально привабливих вебсайтів.

- Негативні сторони:

- Не сумісність зі старими браузерами: Деякі нові функції CSS3 можуть не підтримуватися в старих версіях браузерів, що вимагає використання псевдокласів або JavaScript для створення аналогічних ефектів.

- Складність: CSS3 може бути складним для вивчення і розуміння через велику кількість властивостей і значень.

### Bootstrap

Bootstrap - це відкритий фреймворк для розробки вебсайтів та вебдодатків. Він базується на HTML, CSS та JavaScript і надає готові компоненти та стилі для швидкого створення сучасних вебінтерфейсів.

### Позитивні сторони:

- Швидка розробка: Bootstrap надає готові компоненти, такі як кнопки, форми, таблиці та інші, що значно спрощує процес розробки і зменшує витрати на створення дизайну з нуля.

- Адаптивність: Bootstrap має вбудовану підтримку адаптивного дизайну, що робить його ідеальним для створення вебсайтів, які працюють на різних пристроях і розмірах екрану.

- Спільнота і підтримка: Є велика спільнота розробників, які надають підтримку та розвивають екосистему Bootstrap. Це дозволяє легко знаходити рішення для проблем і використовувати додаткові розширення та плагіни.

### Негативні сторони:

- Схожий дизайн: Використання Bootstrap може призвести до того, що всі вебсайти, створені на цій платформі, будуть виглядати схоже, що не завжди підходить для унікальних проектів.

### JavaScript

JavaScript - це мова програмування, яка використовується для створення інтерактивності на вебсторінках. Вона дозволяє створювати динамічні ефекти, обробляти події користувача та взаємодіяти з сервером.

Позитивні сторони:

- Широкий функціонал: JavaScript надає безліч бібліотек і фреймворків, які дозволяють розробникам створювати різноманітну функціональність, від анімації до AJAX-запитів.
- Інтерактивність: JavaScript дозволяє створювати інтерактивні вебсторінки, де користувачі можуть взаємодіяти з елементами сторінки.
- Кросбраузерність: JavaScript підтримується більшістю сучасних браузерів і дозволяє створювати крос-браузерні вебдодатки.

Негативні сторони:

- Потенційні безпекові проблеми: Використання JavaScript може створювати потенційні безпекові проблеми, такі як XSS-атаки (Cross-Site Scripting), які можуть використовуватися для внесення змін в сторінку або крадіжки даних.
- Працездатність на стороні клієнта: JavaScript залежить від можливостей та продуктивності браузера, що може обмежувати його використання для складних обчислень та обробки даних.

### **2.3 Огляд сучасних інструментів для серверної частини вебдодатку**

Серверна частина вебдодатку відіграє важливу роль у забезпеченні його надійності, продуктивності та функціональності. Це є центральним елементом, який обробляє запити від клієнтів, взаємодіє з базою даних і забезпечує відповідь, яку користувачі бачать у своєму браузері.

PHP [22] (Hypertext Preprocessor) - це серверна мова програмування, яка широко використовується для розробки вебдодатків. Вона була створена в 1994 році Расмусом Лердорфом і з тих пір стала однією з найпопулярніших мов для створення динамічних вебсайтів і вебдодатків.

Переваги PHP:

- Простота вивчення та використання: PHP має досить простий і зрозумілий синтаксис, що робить його доступним для новачків у розробці вебдодатків. Ця простота сприяє швидкому вивченню та створенню функціональних вебсторінок.

- Вбудована підтримка веброзробки: PHP створено спеціально для розробки вебдодатків, і в ньому є багато вбудованих функцій та інструментів для обробки HTTP-запитів, роботи з формами, створення сесій та кукі.

- Підтримка багатьох баз даних: PHP підтримує різні системи управління базами даних, такі як MySQL, PostgreSQL, SQLite, і багато інших. Це дозволяє розробникам взаємодіяти з різними типами даних.

- Велика спільнота та ресурси: Існує активна спільнота PHP-розробників, що пропонує багато безкоштовних ресурсів, документацію та підтримку. Це полегшує вивчення та вирішення проблем.

- Розширюваність: Існують тисячі розширень та бібліотек для PHP, що дозволяє розробникам додавати новий функціонал до своїх додатків без необхідності переписувати все з нуля.

- Фреймворки: PHP має кілька потужних фреймворків, таких як Laravel, Symfony та Zend, які допомагають створювати складні та масштабовані вебдодатки з використанням кращих практик розробки.

Недоліки PHP:



- Швидкодія: Порівняно з іншими мовами програмування, PHP може бути менш продуктивним у виконанні деяких завдань, особливо на великих додатках, через своє обмежене керування пам'яттю та інші технічні обмеження.
- Безпека: Необережне використання PHP може призвести до безпекових вразливостей, таких як Cross-Site Scripting (XSS) та SQL ін'єкції. Важливо слідкувати за безпекою при розробці додатків.
- Версії: Існують старі версії PHP, які можуть містити застарілі та небезпечні функції. Важливо використовувати актуальні версії мови та бібліотек для забезпечення безпеки.
- Масштабованість: PHP не завжди є найкращим вибором для масштабних додатків з великою кількістю одночасних запитів.

Node.js [23] - це середовище виконання JavaScript на стороні сервера, яке дозволяє розробникам створювати швидкі та ефективні вебдодатки. В основі Node.js лежить двигун V8 від Google, який використовується в браузерах для виконання JavaScript.

Переваги Node.js:

- Швидкість: Однією з основних переваг Node.js є його висока швидкість виконання коду. Використання двигуна V8 дозволяє швидко обробляти запити і відповіді на них, що особливо важливо для вебдодатків з великою кількістю одночасних запитів.
- Події та асинхронність: Node.js побудований на принципі подій та асинхронності. Це дозволяє створювати додатки, які ефективно обробляють багато одночасних запитів без блокування інших операцій.

- Єдина мова (JavaScript): Можливість використовувати одну мову (JavaScript) як на стороні клієнта, так і на стороні сервера, спрощує розробку та підтримку вебдодатків.

- Модульність та пакети: Node.js має велику екосистему пакетів, яка дозволяє розробникам використовувати готові модулі та бібліотеки для спрощення розробки. NPM (Node Package Manager) - це одна з найбільших систем керування пакетами в світі.

- Спільнота та підтримка: Node.js має велику та активну спільноту розробників, яка надає підтримку, розвиває модулі та документацію.

Недоліки Node.js:

- Однопоточність: Виконання коду в одному потоці може призвести до блокування заходів у випадку довгих операцій вводу-виводу або обчислень. Хоча Node.js підтримує асинхронність, важливо використовувати цей підхід правильно для уникнення проблем.

- Не підходить для CPU-інтенсивних завдань: Node.js не є найкращим вибором для виконання операцій, які потребують великої кількості обчислень, оскільки це може призвести до затримок в обслуговуванні запитів.

- Менше підтримки для деяких завдань: Node.js може вимагати більше роботи для деяких завдань, які в інших мовах вирішуються легше. Наприклад, робота з XML може бути менш зручною порівняно з іншими мовами.

Отже зважаючи на особливості розробки Інтернет магазину було обрано саме PHP через ряд перел.

## 2.4 Структура розробленої інформаційної системи

Навігація на веборієнтованій системі дозволяє переміщуватися між сторінками та має структуру, що зображена на рисунку 2.1.

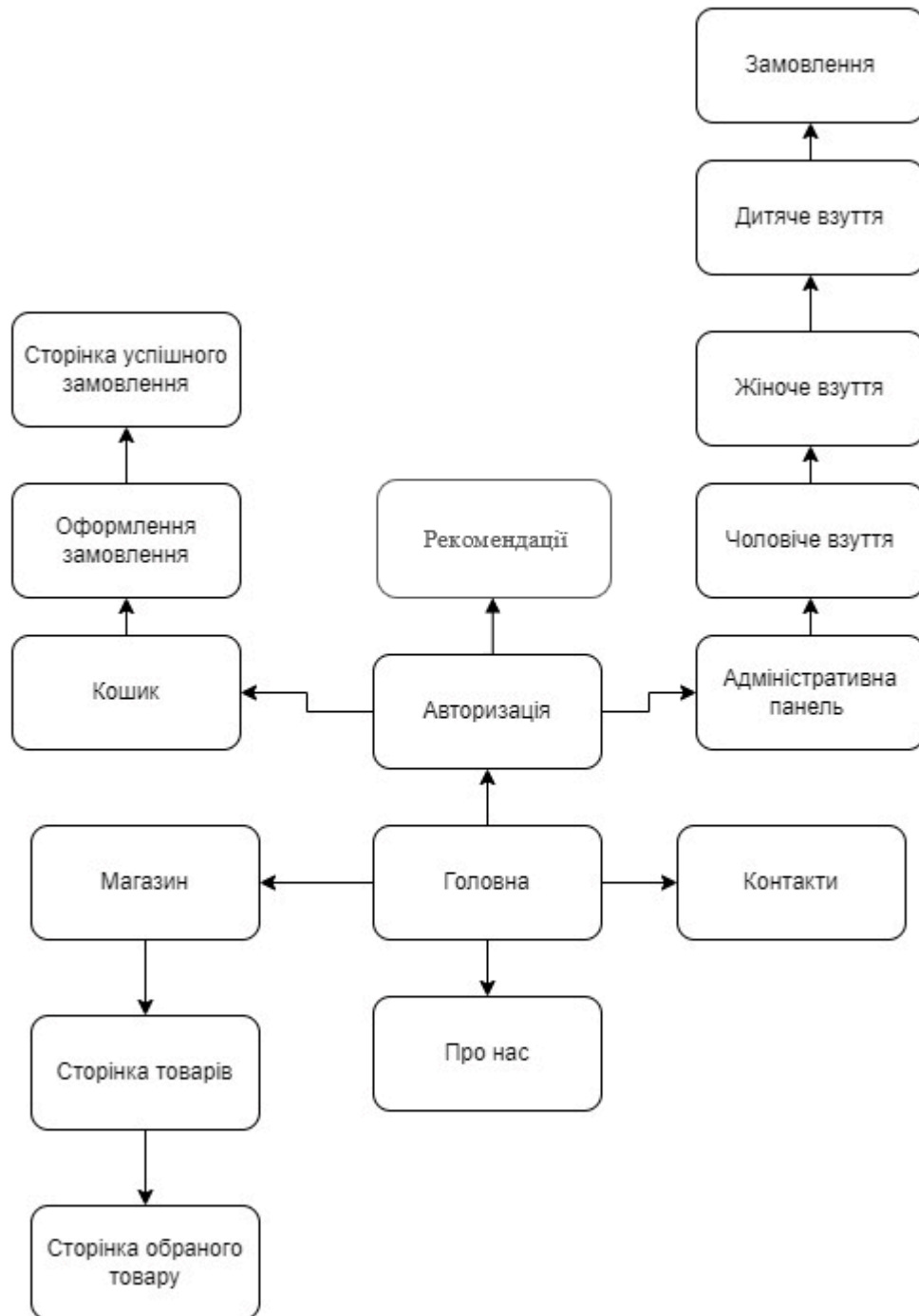


Рисунок 2.1 – Зовнішня структура Інтернет магазину

## 2.5 Проектування бази даних

З метою забезпечення надійної та продуктивної роботи платформи була розроблена база даних. Структура бази даних відображена на схемі ERD, представлений на рисунку 2.2.

Код для створення таблиць можна знайти у додатку А.

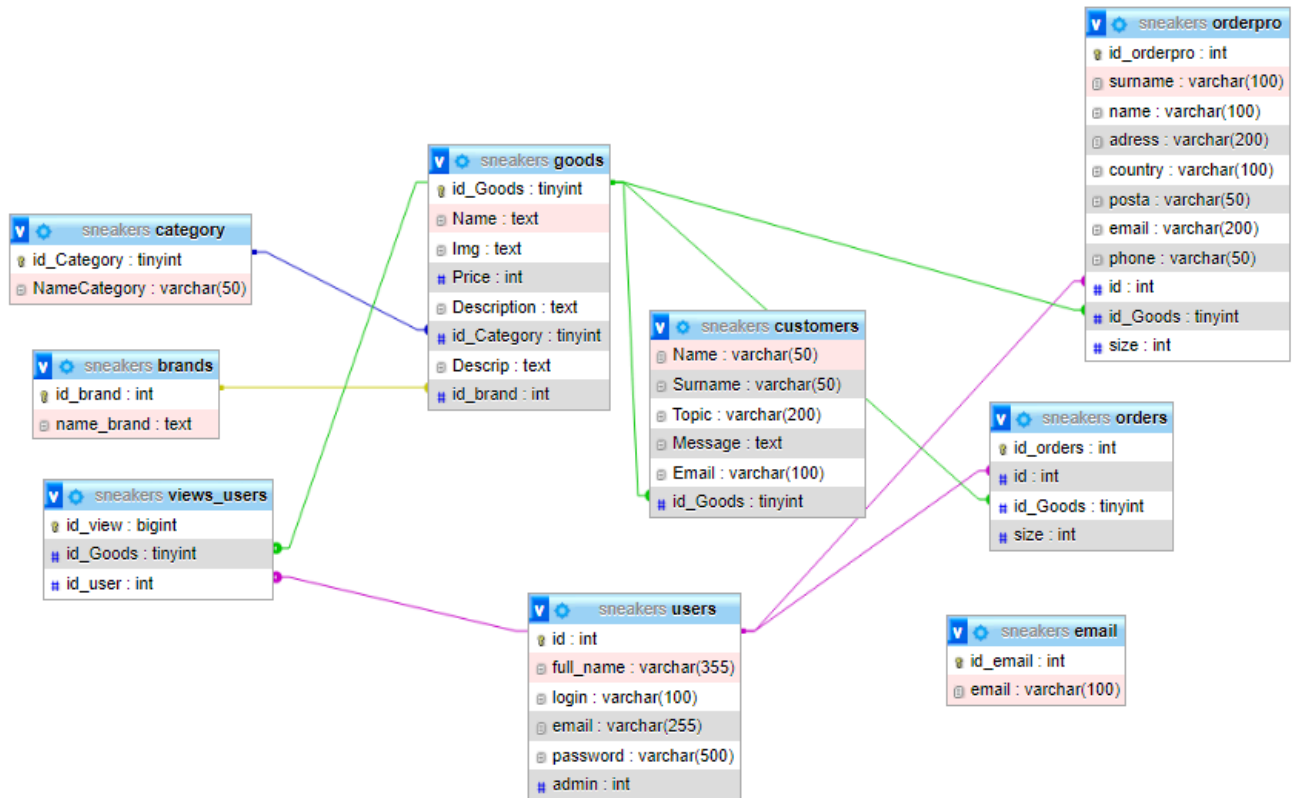


Рисунок 2.2 – Entity-Relation діаграма вебсистеми

База даних складається з дев'яти таблиць: «customers», «category», «brand», «email», «goods», «orderpro», «orders», «users», «views\_users».

Структура таблиці «customers» зображена на рисунку 2.3. Таблиця зберігає інформацію про клієнтів магазину.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 Name	varchar(50)	utf8mb4_unicode_ci		Так	NULL			
<input type="checkbox"/>	2 Surname	varchar(50)	utf8mb4_unicode_ci		Так	NULL			
<input type="checkbox"/>	3 Topic	varchar(200)	utf8mb4_unicode_ci		Так	NULL			
<input type="checkbox"/>	4 Message	text	utf8mb4_unicode_ci		Так	NULL			
<input type="checkbox"/>	5 Email	varchar(100)	utf8mb4_unicode_ci		Ні	Немає			
<input type="checkbox"/>	6 id_Goods	tinyint			Так	NULL			

### Рисунок 2.3 – Структура таблиці «customers»

Структура таблиці «category» зображена на рисунку 2.4. Таблиця зберігає інформацію про назву категорій в Інтернет магазині.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id_Category	tinyint			Ні	Немає		AUTO_INCREMENT	
<input type="checkbox"/>	2 NameCategory	varchar(50)	utf8mb4_unicode_ci		Ні	Немає			

### Рисунок 2.4 – Структура таблиці «category»

Структура таблиці «brand» зображена на рисунку 2.5. Таблиця зберігає інформацію про назву брендів кросівок в Інтернет магазині.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id_brand	int			Ні	Немає		AUTO_INCREMENT	
<input type="checkbox"/>	2 name_brand	text	utf8mb4_0900_ai_ci		Ні	Немає			

### Рисунок 2.5 – Структура таблиці «brand»

Структура таблиці «email» зображена на рисунку 2.6. Таблиця зберігає інформацію про надіслані електронні адреси для розсилки.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id_email	int			Ні	Немає		AUTO_INCREMENT	
<input type="checkbox"/>	2 email	varchar(100)	utf8mb4_unicode_ci		Так	NULL			

### Рисунок 2.6 – Структура таблиці «email»

Структура таблиці «goods» зображена на рисунку 2.7. Таблиця зберігає інформацію про усі товари та їх характеристики наявні у магазині.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id_Goods	tinyint			Ні	Немає		AUTO_INCREMENT	
<input type="checkbox"/>	2 Name	text	utf8mb4_unicode_ci		Ні	Немає			
<input type="checkbox"/>	3 Img	text	utf8mb4_unicode_ci		Ні	Немає			
<input type="checkbox"/>	4 Price	int			Ні	Немає			
<input type="checkbox"/>	5 Description	text	utf8mb4_unicode_ci		Ні	Немає			
<input type="checkbox"/>	6 id_Category	tinyint			Ні	Немає			
<input type="checkbox"/>	7 Descrip	text	utf8mb4_unicode_ci		Так	NULL			
<input type="checkbox"/>	8 id_brand	int			Ні	Немає			

### Рисунок 2.7 – Структура таблиці «goods»

Структура таблиці «orderpro» зображена на рисунку 2.8. Таблиця зберігає інформацію про активні замовлення які оформили покупці.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1	id_orderpro 🗑️			Ні	Немає		AUTO_INCREMENT	
<input type="checkbox"/>	2	surname	varchar(100)	utf8mb4_unicode_ci	Так	NULL			
<input type="checkbox"/>	3	name	varchar(100)	utf8mb4_unicode_ci	Так	NULL			
<input type="checkbox"/>	4	adress	varchar(200)	utf8mb4_unicode_ci	Так	NULL			
<input type="checkbox"/>	5	country	varchar(100)	utf8mb4_unicode_ci	Так	NULL			
<input type="checkbox"/>	6	posta	varchar(50)	utf8mb4_unicode_ci	Так	NULL			
<input type="checkbox"/>	7	email	varchar(200)	utf8mb4_unicode_ci	Так	NULL			
<input type="checkbox"/>	8	phone	varchar(50)	utf8mb4_unicode_ci	Так	NULL			
<input type="checkbox"/>	9	id 🗑️			Так	NULL			
<input type="checkbox"/>	10	id_Goods 🗑️			Так	NULL			
<input type="checkbox"/>	11	size			Так	NULL			

Рисунок 2.8 – Структура таблиці «orderpro»

Структура таблиці «orders» зображена на рисунку 2.9. Таблиця зберігає інформацію про речі у кошиках користувачів.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1	id_orders 🗑️			Ні	Немає		AUTO_INCREMENT	
<input type="checkbox"/>	2	id 🗑️			Так	NULL			
<input type="checkbox"/>	3	id_Goods 🗑️			Так	NULL			
<input type="checkbox"/>	4	size			Так	NULL			

Рисунок 2.9 – Структура таблиці «orders»

Структура таблиці «users» зображена на рисунку 2.10. Таблиця зберігає інформацію про усіх зареєстрованих користувачів та адміністраторів.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1	id 🗑️			Ні	Немає		AUTO_INCREMENT	
<input type="checkbox"/>	2	full_name	varchar(355)	utf8_general_ci	Так	NULL			
<input type="checkbox"/>	3	login	varchar(100)	utf8_general_ci	Так	NULL			
<input type="checkbox"/>	4	email	varchar(255)	utf8_general_ci	Так	NULL			
<input type="checkbox"/>	5	password	varchar(500)	utf8_general_ci	Так	NULL			
<input type="checkbox"/>	6	admin			Так	NULL			

Рисунок 2.10 – Структура таблиці «users»

Структура таблиці «views\_users» зображена на рисунку 2.11. Таблиця зберігає інформацію про перегляди користувачів.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
<input type="checkbox"/>	1 id_view	bigint			Ні	Немає		AUTO_INCREMENT	
<input type="checkbox"/>	2 id_Goods	tinyint			Ні	Немає			
<input type="checkbox"/>	3 id_user	int			Ні	Немає			

Рисунок 2.11 – Структура таблиці «views\_users»

## 2.6 Алгоритм рекомендацій

У світі, де навколо нас здавна вирішуються задачі вибору, виникає необхідність в ефективних алгоритмах рекомендацій. Від Інтернет магазинів до соціальних медіа, від сервісів потокового відео до музичних платформ, рекомендаційні системи знаходять своє застосування в різних сферах. Ці алгоритми вирішують складні завдання прогнозування і рекомендації, базуючись на аналізі користувацьких даних та враховуючи різноманітні фактори, такі як інтереси, поведінка та контекст.

Необхідність реалізації алгоритмів рекомендацій:

- Підвищення залучення клієнтів: Рекомендаційні системи на основі алгоритму Аргіогі дозволяють підвищити залучення клієнтів шляхом пропозиції персоналізованих товарів або послуг. Аналіз зв'язків між покупками дозволяє запропонувати користувачам товари або послуги, які вони можуть бути зацікавлені в придбанні.
- Покращення користувацького досвіду: Рекомендаційні системи дозволяють створити персоналізований досвід для користувачів, що полегшує їхню навігацію по вебсайту та забезпечує доступ до релевантної інформації чи товарів.
- Збільшення конверсії: Аналіз поведінки користувачів та рекомендації товарів, які відповідають їхнім потребам та інтересам, сприяє

збільшенню конверсії на вебсайті. Користувачі, які отримують персоналізовані рекомендації, схильніше здійснюють покупки.

– Виявлення патернів та трендів: Алгоритми дозволяють виявляти патерни та тренди у виборі товарів користувачами. Ця інформація може бути використана для уточнення асортименту товарів, вдосконалення маркетингових стратегій та планування асортименту.

Алгоритми стають основою для розробки персоналізованих систем, які допомагають користувачам знаходити контент, товари чи послуги, які відповідають їхнім уподобанням та потребам.

Переваги Аpriori :

– Простота реалізації: Алгоритм Аpriori є досить простим у реалізації, що робить його доступним для використання в різних сценаріях.

– Ефективність у роботі з великими даними: Аpriori добре працює з великими наборами даних, оскільки він використовує принципи апriori інформації для скорочення кількості кандидатів на набір правил асоціацій.

– Практичність в реальних сценаріях: Цей алгоритм ефективно використовується в сферах, де важливо виявлення асоціативних правил, таких як роздрібна торгівля, аналіз покупок тощо.

Недоліки Аpriori:

– Велика обчислювальна складність: Зазвичай алгоритм Аpriori може бути вимогливим до обчислювальних ресурсів, особливо при роботі з великими наборами даних.

– Проблема розрідженості даних: У випадку, коли дані мають велику розрідженість (тобто, коли більшість комбінацій є рідкісними), Аpriori може



генерувати велику кількість кандидатів на набір правил, що може призвести до неефективності алгоритму.

Переваги Eclat:

- Ефективність в умовах розрідженості даних: Eclat може бути ефективним у випадках з великою розрідженістю даних, оскільки він працює з вертикальною формою даних і не потребує генерації комбінацій.

- Простота реалізації: Цей алгоритм також досить простий у реалізації, що робить його доступним для застосування в різних сферах.

Недоліки Eclat:

- Неєфективність з великими наборами даних: У порівнянні з Аргіогі, Eclat може бути менш ефективним у випадках з великими наборами даних, особливо коли дані мають низьку розрідженість.

- Відсутність генерації правил: Оскільки Eclat не генерує правил напряду, а просто виявляє часті комбінації елементів, він може бути менш корисним у деяких сценаріях, де важливо знайти самі правила асоціацій.

Для виконання роботи було обрано алгоритм Аргіогі, оскільки він краще відповідає вимогам та характеристикам нашого завдання. Однією з ключових причин обрання Аргіогі є його простота реалізації та відома ефективність у роботі з великими обсягами даних.

У контексті нашого завдання, де важливо виявлення часто зустрічаючихся комбінацій елементів у наборах даних, Аргіогі виявляється більш підходящим варіантом. Враховуючи, що інформаційна система має справу зі зрідженими даними та потребуємо ефективного аналізу асоціативних правил, вибір Аргіогі стає обґрунтованим.

Таким чином, алгоритм Аргіогі був обраний як оптимальний інструмент для досягнення цілей нашого проекту, забезпечуючи при цьому необхідну ефективність та простоту в реалізації.



## ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

### 3.1 Опис програмної реалізації

Вебсайт складається з двох основних частин: користувацької та адміністративної.

Користувацька частина відповідає за відображення товарів та взаємодію з користувачами. Вона включає в себе інтерфейс, який дозволяє користувачам переглядати товари, додавати їх до кошика та оформляти замовлення. Ця частина сайту розроблена з використанням PHP, JavaScript та JQuery, що забезпечує гладку та інтерактивну користувацьку взаємодію.

Адміністративна частина відповідає за адміністрування контенту на сайті та відслідковування замовлень. Це включає в себе панель управління, яка дозволяє адміністраторам сайту додавати нові товари, оновлювати інформацію про існуючі товари та відслідковувати статус замовлень користувачів.

Вебдодаток, базується на клієнт-серверній архітектурі, яка є однією з найпоширеніших моделей для розробки вебдодатків. В цій моделі, клієнтська частина (яка виконується в браузері користувача) відправляє запит до сервера. Сервер, в свою чергу, обробляє цей запит, виконує необхідні операції (наприклад, взаємодію з базою даних) і відправляє відповідь назад до клієнта.

Серверна частина включає в себе логіку обробки запитів, взаємодію з базою даних та генерацію відповідей. Ця частина вебдодатку реалізована за допомогою PHP, мови програмування, яка широко використовується для розробки серверної частини вебдодатків.

Вебдодаток використовує ряд технологій та бібліотек для реалізації своєї функціональності. Для серверної частини використовується PHP. PHP - це популярна мова програмування з відкритим вихідним кодом, яка особливо підходить для веброзробки і може бути вбудована в HTML.

Для клієнтської частини використовуються JavaScript та JQuery. JavaScript - це мова програмування, яка використовується для створення інтерактивних вебсайтів. JQuery - це швидка, невелика та багатофункціональна бібліотека

JavaScript. Вона робить такі речі, як перехід та маніпуляція HTML документами, обробка подій та анімація, набагато простішими за допомогою простого API, яке працює на багатьох браузерах.

Однією з головних функцій є генерація блоків товарів з бази даних, цей фрагмент JavaScript є критичною частиною програми, яка відповідає за взаємодію з користувачем та обробку даних. Повний код функції знаходиться у додатку Б. Детальний аналіз функції `select()`, яка здійснює обрання елементів і відправку даних.

- Отримання об'єктів з DOM:
  - По-перше, функція отримує доступ до випадального списку (`select`) на сторінці за його унікальним ідентифікатором. Це важливо, оскільки взаємодія з користувачем передбачає обробку введених ним даних.
  - Після цього вибирається варіант, який вибрав користувач, для подальшого використання.
- Оновлення значень і відправлення форми:
  - Після вибору користувачем опції, значення цієї опції автоматично оновлюється в прихованому полі форми.
  - Нарешті, дані з цієї форми відправляються на сервер за допомогою методу `submit()`, щоб вони могли бути оброблені та збережені.

Цей код націлений на створення динамічного інтерфейсу для користувачів та забезпечення зручного способу вибору та відправки даних. Він використовується для створення блоків з інформацією про товари, які отримуються з бази даних за допомогою РНР. В кожному з цих блоків формується за допомогою циклу `while`, що проганяється через результати запиту до бази даних.

Все вищезазначене створює важливу базу для подальшої роботи програми та допомагає забезпечити користувачам зручність та ефективність при користуванні.

Ще однією важливою складовою є алгоритм підбору рекомендованих товарів на сайті. PHP код який реалізує алгоритм Apriori для генерації часто зустрічаються наборів елементів на основі даних про транзакції, які зберігаються в таблиці MySQL з назвою transactions знаходиться у додатку В.

– Визначення Функцій:

- generateFrequentItemSets: Обчислює частоту кожного елемента в транзакціях та визначає часто зустрічаються набори елементів на основі мінімального порогу підтримки.
- aprioriAlgorithm: Виконує алгоритм Apriori ітеративно до тих пір, поки більше часто зустрічаються набори елементів не буде знайдено.
- generateCandidateItemSets: Генерує кандидатські набори елементів шляхом об'єднання часто зустрічаються наборів елементів з попередньої ітерації.

– Основне Виконання:

- Викликає функцію aprioriAlgorithm з даними про транзакції та мінімальним порогом підтримки.
- Виводить часто зустрічаються набори елементів разом з кількістю їх підтримки.

Вебдодаток також взаємодіє з базою даних. Код підключення до бази даних реалізований на PHP і знаходиться у додатку Г. Це забезпечує безпечно та ефективно зберігання та доступ до даних вашого вебсайту. Зазвичай, для взаємодії з базою даних використовуються SQL-запити. SQL (Structured Query Language) - це стандартна мова для управління та маніпуляції даними.

### **3.2 Аналіз результатів клієнтської частини вебсистеми**

Інтернет магазин кросівок виконаний у єдиному стилі та використовує одну кольорову гаму. Кожна сторінка складається з шапки (рис. 3.1) що містить

навігаційне меню, лого та кнопки входу до корзини, особистого кабінету та рекомендацій, основної частини (рис. 3.2) та підвалу (рис. 3.3).



Рисунок 3.1 – Шапка



Рисунок 3.2 – Основна частина

Copyright © 2024 | Всі права захищені ♥

Рисунок 3.3 – Підвал

Взаємодія користувача з сайтом починається з головної сторінки, зображеної на рисунку 3.4 та зображена кнопка «До магазину» яка перенаправить користувача до магазину.



Рисунок 3.4 – Головна сторінка

Також на головній наявні категорії для швидкого вибору категорії товару (рис. 3.5).

### Категорії



Рисунок 3.5 – Категорії

Блок «Розпродаж» зображено на сторінці 3.6.

## Великий розпродаж!



### Шалені знижки на товари


Ви повинні вірити у себе, особливо у ті моменти, коли у вас не вірить ніхто.


[ДО МАГАЗИНУ](#)


Рисунок 3.6 – Розпродаж

На сторінках «Головна», «Магазин», «Сторінка товару», в підвалі сторінки наявний блок «Підписка» зображений на рисунку 3.7.

### Контактна інформація

 Україна м.Київ вул.Соборна 46

 0 800 359 89 89

 kingdom.of.sneakers@gmail.com

Підписатися

Email

[НАДІСЛАТИ](#)

Рисунок 3.7 – Блок «Підписка»



На сторінці «Про нас» (рис. 3.8) зображена інформація про магазин з ОПИСОМ.

Головна / Про нас



## Про наш магазин

Вітаємо у "Kingdom of Sneakers" - вашому королівстві стилю та комфорту для ніг!

Наша історія почалася з пристрасті до стилю та комфорту, і ми продовжуємо ділитися цією пристрастю з нашими клієнтами. У нас ви знайдете широкий вибір кросівок від найвідоміших світових брендів, таких як Nike, Adidas, Puma, New Balance та багатьох інших.

Приходьте до "Kingdom of Sneakers" і дозвольте нам допомогти вам знайти ідеальну пару кросівок, яка підкреслить ваш неповторний стиль та допоможе вам крокувати впевнено вперед до нових здобутків!

Рисунок 3.8 – Блок «Про нас»

На сторінці «Магазин» (рис. 3.9) зображено усі доступні товари у магазині, також можна вибрати категорію товарів, тобто товари для жінок, чоловіків чи дітей (рис. 3.10). Також можна відфільтрувати товари за необхідним брендом взуття (рис. 3.11).

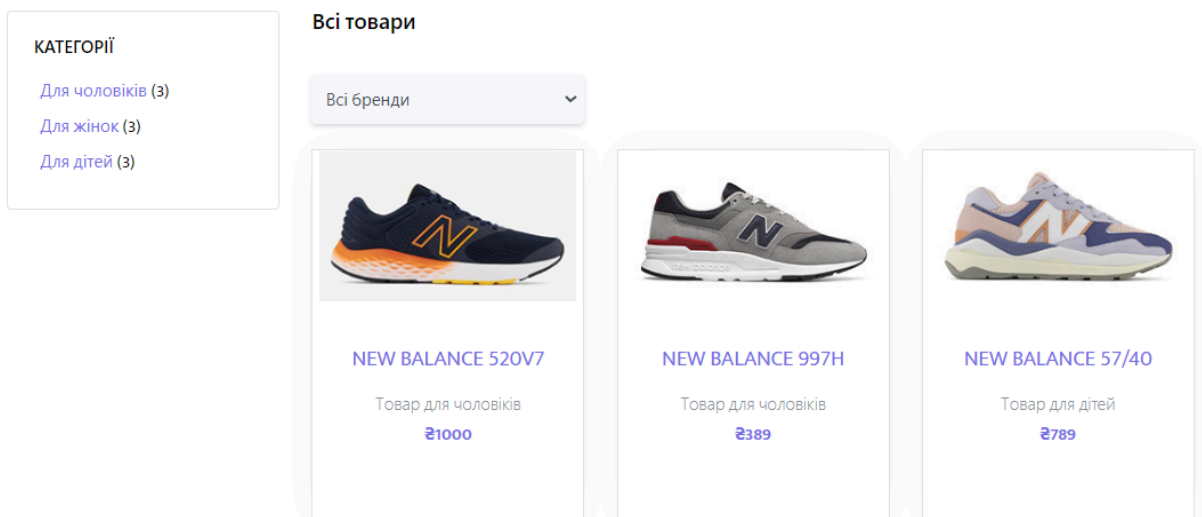
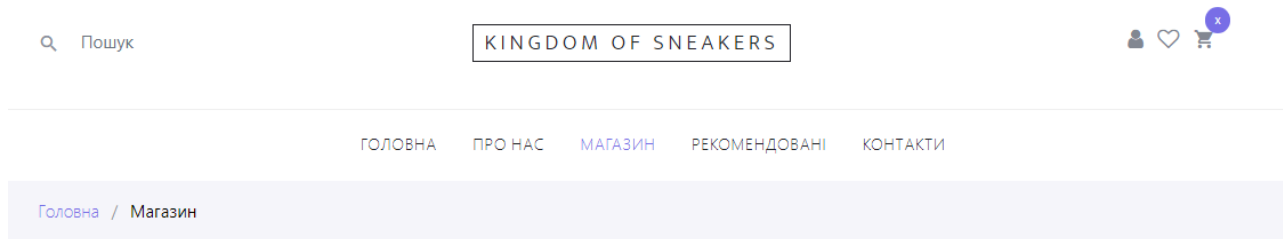


Рисунок 3.9 – Сторінка «Магазин»

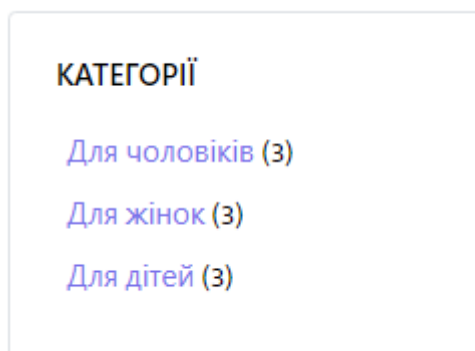


Рисунок 3.10 – Фільтр категорії товару

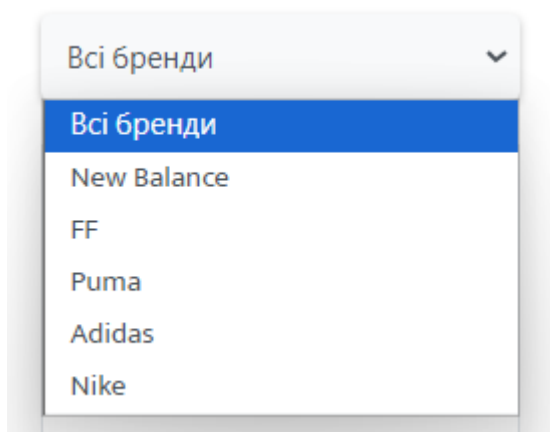



Рисунок 3.11 – Фільтр брендів

На початку сторінки обраного користувачем товару, зображеної на рисунку 3.12, розташований сам товар з описом картинкою та ціною, нижче сторінкою розташована таблиця розмірів (рис. 3.13).

Головна / NEW BALANCE 997H



### NEW BALANCE 997H

New Balance 997H створені для незалежних осіб. Це зразок сміливого мислення. Оновлені 997H вносять своєрідний стиль для модельного ряду, який пропонує кольорові рішення сучасності та модерну. Вони виготовлені для тих, хто готовий відрізнятись. Можна, наприклад, зліпити квіти, машинку, новорічну ялинку з іграшками та багато іншого.

Товар для чоловіків

**€389**

Виберіть розмір

40 EUR  41 EUR  42 EUR  43 EUR

[Таблиця розмірів](#)

**ДО КОШИКУ**

Рисунок 3.12 – Сторінка товару

Таблиця розмірів Чоловічого взуття

Довжина стопи, см	США	СНГ	EUR	UK
24.5	6.5	38	39	5.5
25	7	39	40	6
25.5	7.5	39.5	40.5	6.5
26	8	40	41	7
26.5	8.5	41	42	7.5
27	9	41.5	42.5	8
27.5	9.5	42	43	8.5
28	10	43	44	9
28.5	10.5	43.5	44.5	9.5
29	11	44	45	10
29.5	11.5	44.5	45.5	11.5
30	12	45	46	11
31	13	46.5	47.5	12
32	14	47.5	48.5	13
33.5	15	48.5	49.5	14
34	16	49.5	50.5	15
35	17	50.5	51.5	16
36	18	51.5	52.5	17

Рисунок 3.13 – Таблиця розмірів

На сторінці «Контакти» зображені магазини мережі по всій країні зображені на рисунку 3.14.

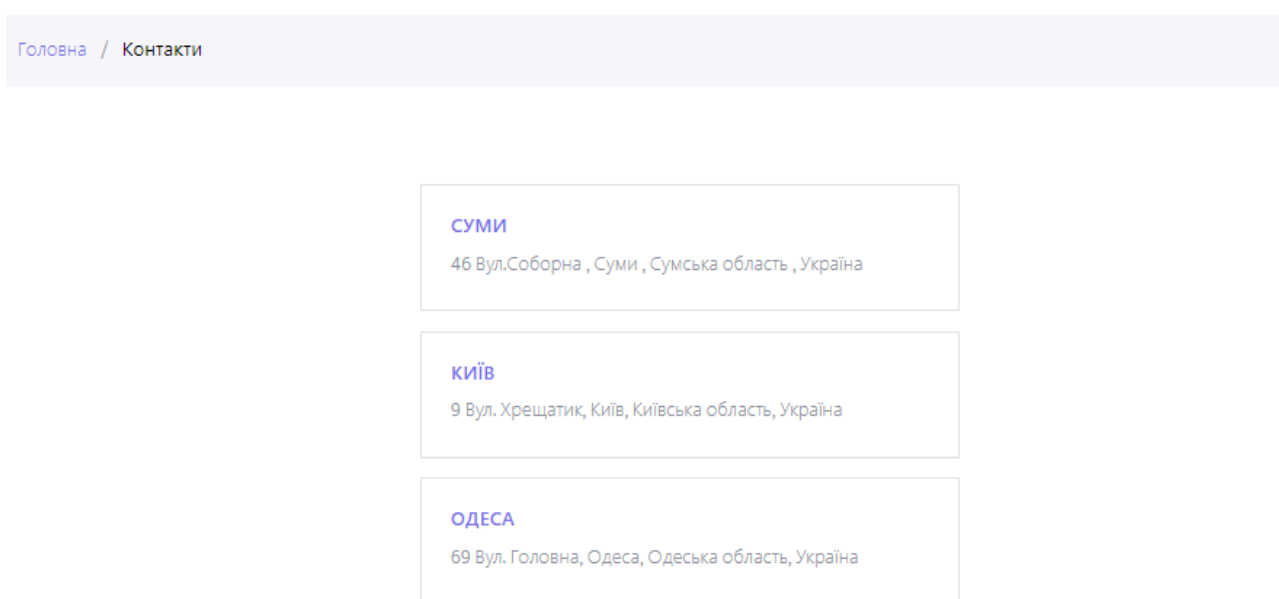


Рисунок 3.14 – Сторінка «Контакти»

Сторінка «Авторизація» зображена на рисунку 3.15 та створена для авторизації користувачів до особистого кабінету. Після авторизації

користувачеві доступні функції кошик, замовлення товару та рекомендовані товари.

ГОЛОВНА   ПРО НАС   МАГАЗИН   РЕКОМЕНДОВАНИ   КОНТАКТИ

## Увійти до особистого кабінету



Логін

Пароль

УВІЙТИ

У вас досі немає аккаунту? - зареєструватися!


[Відновлення паролю!](#)

Рисунок 3.15 – Сторінка «Авторизації»

Сторінка «Реєстрація» зображена на рисунку 3.16 та створена для реєстрації користувачів.

---

## Зареєструватись



ПІБ

Логін

Електронна пошта

Пароль

Підтвердження пароля

У вас вже є обліковий запис? - авторизуйтеся!

Рисунок 3.16 – Сторінка «Реєстрації»

Сторінка «Відновлення паролю» зображена на рисунку 3.17 та створена для відновлення паролю користувачів.

## Відновлення паролю



Логін

Новий пароль

Підтвердження нового  
паролю

ЗМІНИТИ

Рисунок 3.17 – Сторінка «Відновлення паролю»

Сторінка «Кошик» зображена на рисунку 3.18 та відображає усі товари, що користувач додав для покупки.

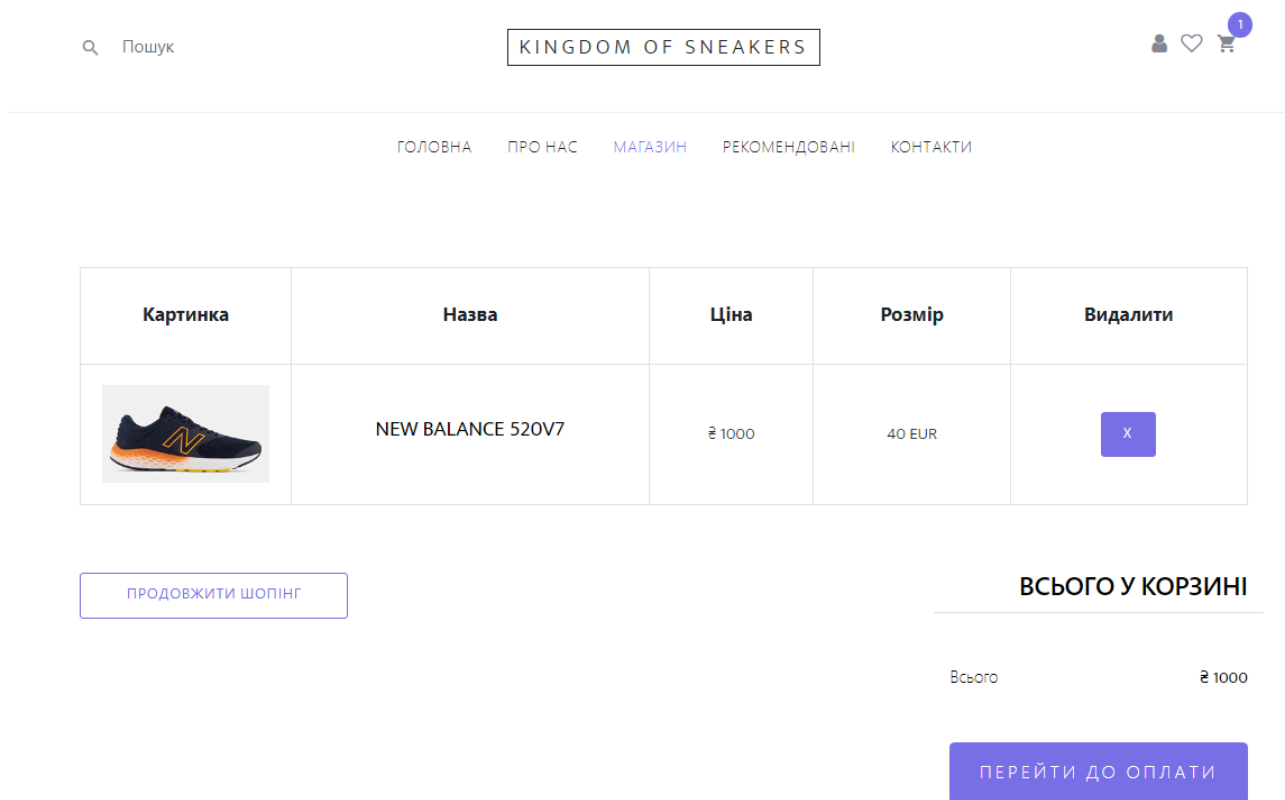


Рисунок 3.18 – Сторінка «Кошик»

Сторінка «Замовлення» зображена на рисунку 3.19 й слугує для оформлення замовлення.



Пошук

KINGDOM OF SNEAKERS

ГОЛОВНА ПРО НАС МАГАЗИН РЕКОМЕНДОВАНИ КОНТАКТИ

### Платіжна інформація

Прізвище \*  Ім'я \*

Домашня адреса \*   
Назва вулиці, номер будинку, номер дому

Країна \*  Поштовий індекс \*

Електронна адреса \*  Номер телефону \*

### Ваше замовлення

Товар	Всього
NEW BALANCE 520V7 x 1	€ 1000
<b>Всього</b>	<b>€ 1000</b>

[ОФОРМИТИ](#)

Рисунок 3.19 – Сторінка «Замовлення»

Сторінка «Подяка користувачу» зображена на рисунку 3.20 й слугує для вираження подяки користувачеві за покупку товару в магазині.

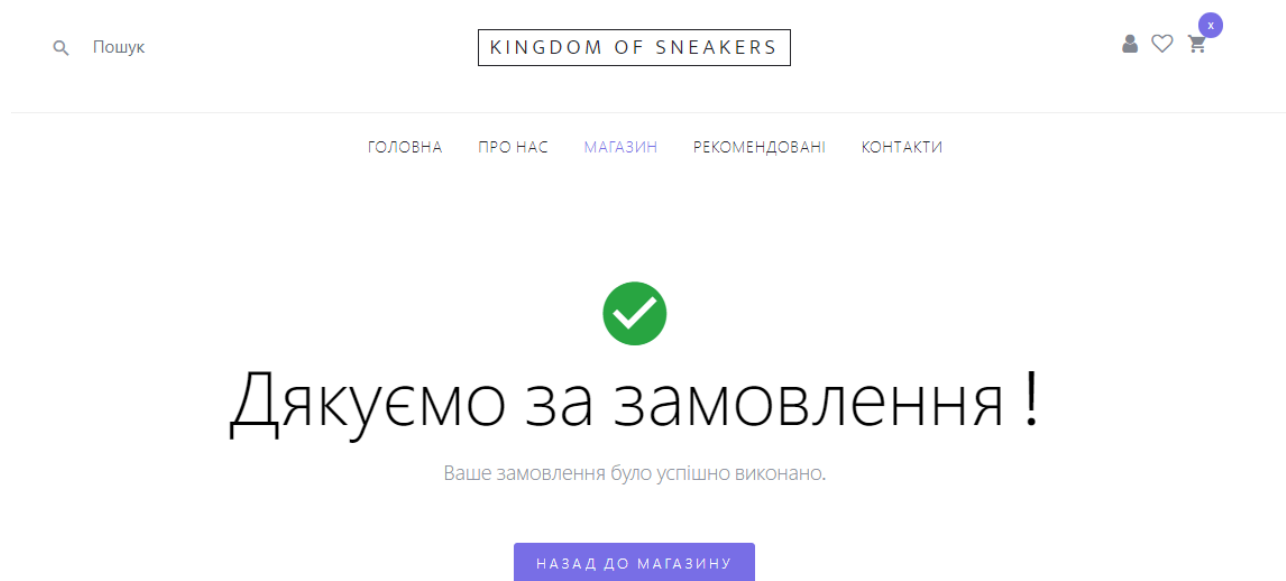


Рисунок 3.20 – Сторінка «Замовлення»

Якщо користувач не увійшов до особистого кабінету, Інтернет магазин запропонує увійти або зареєструватися (рис.3.21).

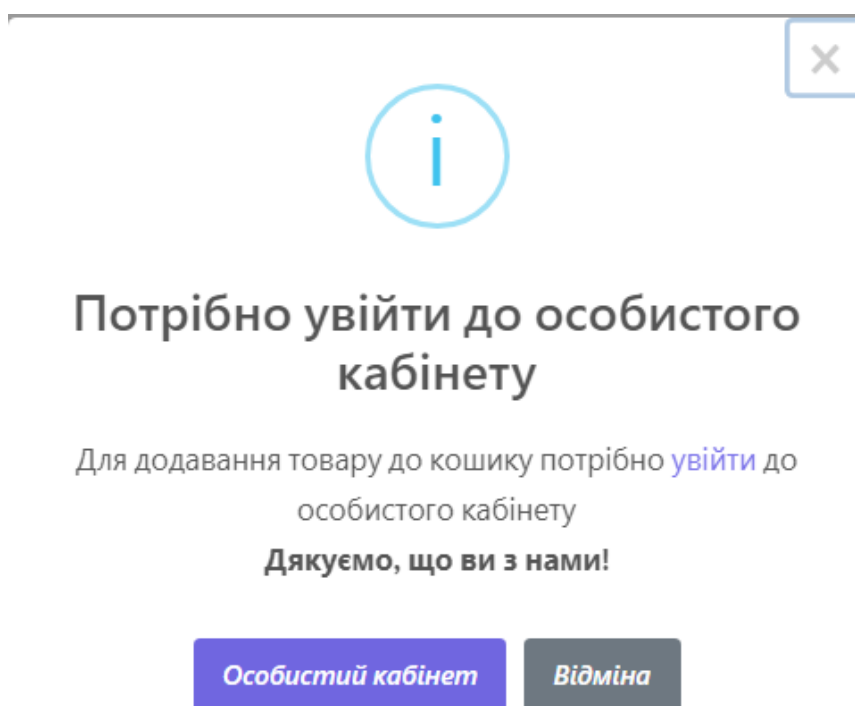


Рисунок 3.21 – Спливаюче вікно

Якщо у користувача є особистий кабінет він має можливість увійти до нього та додати товар до кошику, якщо особистого кабінету немає то користувач може зареєструватися. Після додавання товару до кошику є можливість замовити його.

Якщо перейти в кошик не замовивши жодного товару система покаже повідомлення (рис 3.22).

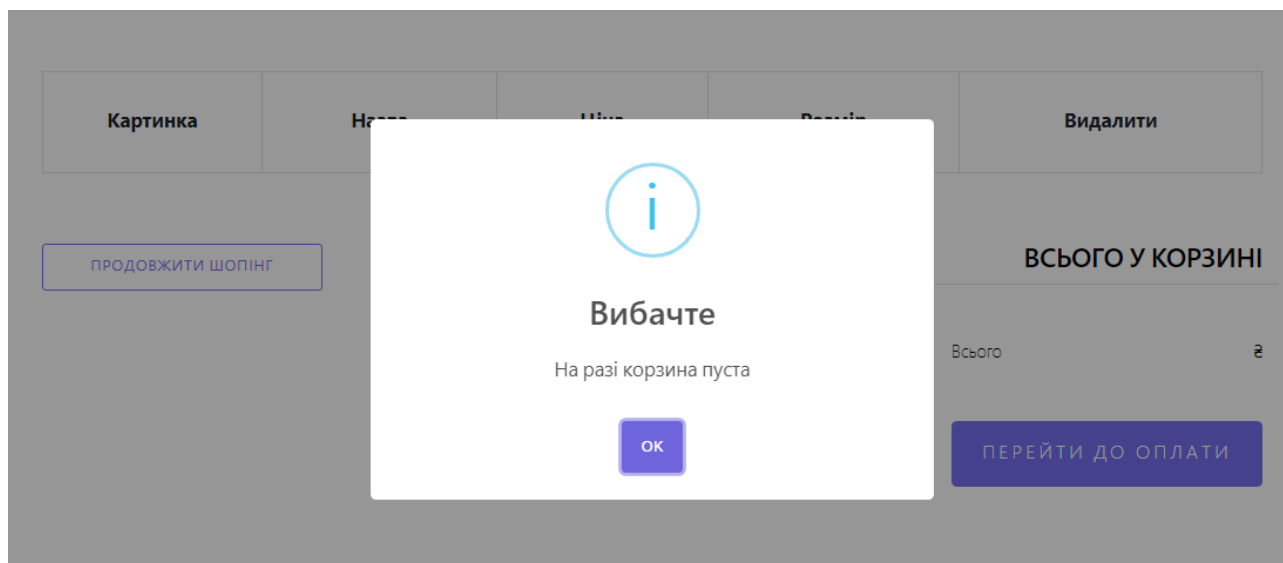


Рисунок 3.22 – Спливаюче вікно

При натисканні кнопки «Перейти до оплати» відкривається сторінка зображена на рисунку 3.19 після чого необхідно заповнити усі поля в іншому випадку Інтернет магазин не здійснить оформлення замовлення та покаже сповіщення зображене на рисунку 3.23.

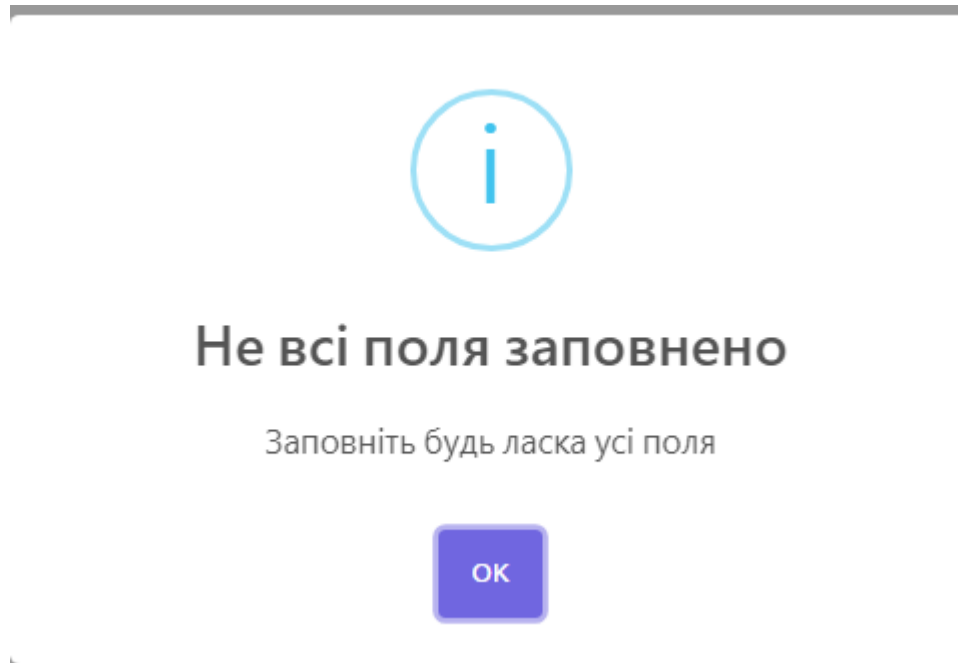


Рисунок 3.23 – Спливаюче вікно

У випадку заповнення усіх полів користувача буде перенаправлено до сторінки зображеної на рисунку 3.18 звідти натиснувши на кнопку «Продовжити шопінг» користувач переміститься до сторінки товарів, а його корзина буде очищена.

На сторінці авторизації здійснена перевірка на наповненість поля, при неведені даних або введені неправильних даних система попередить користувача про помилку (рис. 3.24).

## Увійти до особистого кабінету



Логін

Пароль

УВІЙТИ

У вас досі немає акаунту? - зареєструватися!

[Відновлення паролю!](#)

**Неправильний логін або пароль**

Рисунок 3.24 – Неправильно введені дані

На сторінці «Зареєструвати користувача» присутня перевірка на схожість даних на пошту (рис. 3.25).

Електронна пошта

Введіть пароль

Підтвердження пароля



Електронна адреса має містити знак "@". В електронній адресі "spetan" знака "@" немає.

Рисунок 3.25 – Неправильно введені дані

### 3.3 Аналіз результатів адмін частини вебсистеми

Вхід до адміністративної панель здійснюється через вхід до особистого кабінету з правильним логіном та паролем. Після авторизації відкривається сторінка зображена на рисунку 3.26.

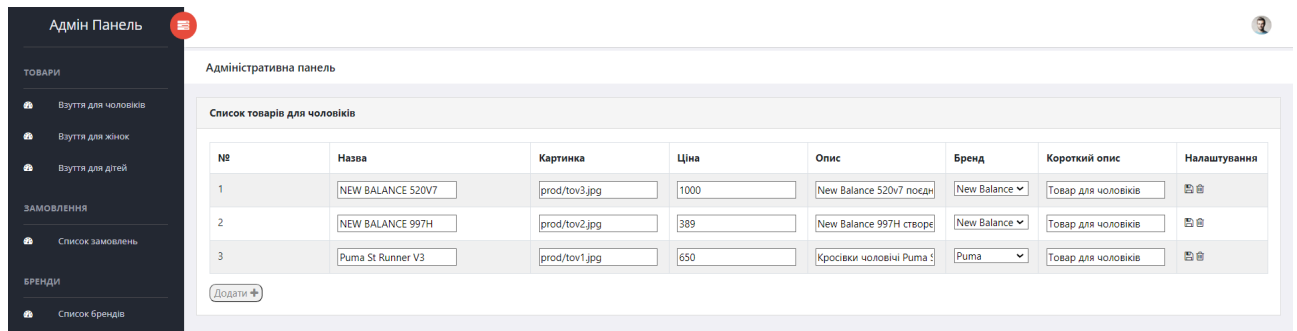


Рисунок 3.26 – Головна сторінка адмін частини

У адміністративному розділі розташовані чотири сторінки.

Адміністративна сторінка чоловічого взуття зображена на рисунку 3.27 та містить інформацію про чоловіче взуття яку можна редагувати. Сторінки з дитячим (рис 3.28) та жіночим взуттям (рис 3.29) є аналогічними.

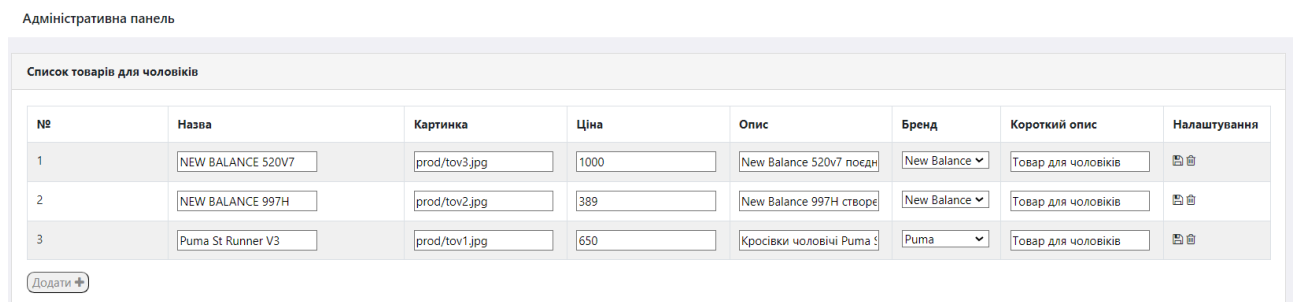








Рисунок 3.27 – Сторінка «Чоловіче взуття»

Адміністративна панель

Список товарів для дітей







№	Назва	Картинка	Ціна	Опис	Бренд	Короткий опис	Налаштування
1	NEW BALANCE 57/40	prod/тов6.jpg	789	New Balance 57/40 – це н	New Balance	Товар для дітей	 
2	Nike REVOLUTION 6 NN	prod/тов7.jpg	1050	Дитячі Кросівки Nike REV	Nike	Товар для дітей	 
3	Asics ROADHAWK FF GS	prod/тов8.jpg	1100	Кросівки дитячі Asics RO	FF	Товар для дітей	 

Додати +

Рисунок 3.28 – Сторінка «Дитяче взуття»

Адміністративна панель

Список товарів для жінок

№	Назва	Картинка	Ціна	Опис	Бренд	Короткий опис	Налаштування
1	NEW BALANCE 237	prod/тов9.jpg	2550	Сучасний дизайн поєдну	New Balance	Товар для жінок	 
2	NEW BALANCE XC-72 T1	prod/тов10.jpg	1360	Для цієї серії моделей XC	New Balance	Товар для жінок	 
3	ADIDAS ORIGINALS FORU	prod/тов4.jpg	1200	Кросівки жіночі Adidas C	Adidas	Товар для жінок	 

Додати +

Рисунок 3.29 – Сторінка «Жіноче взуття»

Сторінка «Список замовлень» зображена на рисунку 3.30 й відображає усі активні замовлення які користувач підтвердив.

Активні замовлення

Список активних замовлень


№	Назва товару	Розмір	Країна	Індекс	Телефон	Адреса	Налаштування
1	NEW BALANCE 520V7	40	Україна	40034	0666666666	Вул. Заливна	

Рисунок 3.30 – Сторінка «Список замовлень»

Також адмін частина містить сторінку брендів де їх можна додавати, видаляти чи редагувати (рис.3.31).

## Адміністративна панель











Список брендів		
№	Назва	Налаштування
1	<input type="text" value="New Balance"/>	 
2	<input type="text" value="FF"/>	 
3	<input type="text" value="Puma"/>	 
4	<input type="text" value="Adidas"/>	 
5	<input type="text" value="Nike"/>	 

Рисунок 3.31 – Сторінка «Бренди»

Додавання здійснюється на всіх сторінках за однаковим принципом - натискаємо кнопку «Додати» (рис.3.32) після чого буде додано поле знизу, де необхідно заповнити відповідні поля (рис.3.33).

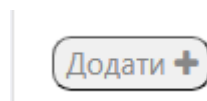


Рисунок 3.32 – Кнопка «Додати»







Список товарів для дітей							
№	Назва	Картинка	Ціна	Опис	Бренд	Короткий опис	Налаштування
1	<input type="text" value="NEW BALANCE 57/40"/>	<input type="text" value="prod/тов6.jpg"/>	<input type="text" value="789"/>	<input type="text" value="New Balance 57/40 – це н"/>	<input type="text" value="New Balance"/>	<input type="text" value="Товар для дітей"/>	 
2	<input type="text" value="Nike REVOLUTION 6 NN"/>	<input type="text" value="prod/тов7.jpg"/>	<input type="text" value="1050"/>	<input type="text" value="Дитячі Кросівки Nike REV"/>	<input type="text" value="Nike"/>	<input type="text" value="Товар для дітей"/>	 
3	<input type="text" value="Asics ROADHAWK FF GS"/>	<input type="text" value="prod/тов8.jpg"/>	<input type="text" value="1100"/>	<input type="text" value="Кросівки дитячі Asics RO"/>	<input type="text" value="FF"/>	<input type="text" value="Товар для дітей"/>	 
4	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="New Balance"/>	<input type="text"/>	

Рисунок 3.33 – Поля для заповнення під час додавання

Щоб додати товар необхідно натиснути кнопку «Змінити» (рис.3.34). Якщо не заповнено всі поля система виведе відповідне повідомлення (рис.3.35).



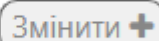


Рисунок 3.34 – Кнопка «Змінити»



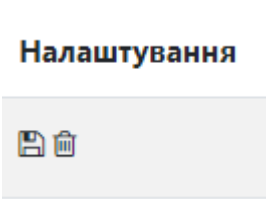
## Не всі поля заповнено

Заповніть будь ласка усі поля :)



Рисунок 3.35 – Спливаюче вікно

Для редагування товару необхідно вписати у поля необхідні значення та натиснути кнопку зберегти (рис.3.36). Для видалення товару необхідно натиснути кнопку видалити (рис.3.36).



Налаштування



Рисунок 3.36 – Кнопки видалення та збереження

Також у шапці профілю можна вийти з адмін панель (рис.3.37).

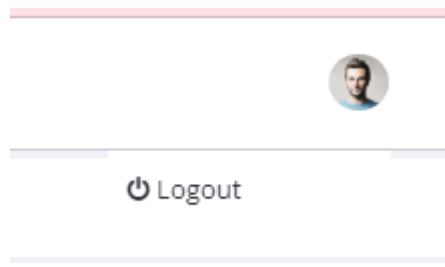


Рисунок 3.37 – Вихід з адміністративної панелі

## ВИСНОВКИ

В ході реалізації інформаційної вебсистеми для роздрібної торгівлі взуттям було проведено обширний аналіз та розроблено рішення, спрямоване на задоволення потреб сучасного споживача та оптимізацію процесів у сфері онлайн-торгівлі.

Було проведено аналіз сучасних інструментів та методів розробки, а також вибрано оптимальні технології для створення інформаційної платформи. На основі цього аналізу було розроблено структуру платформи та спроектовано базу даних, що дозволяє ефективно керувати та обробляти інформацію про товари та користувачів.

В ході дослідження була розроблена методологія створення інформаційної вебсистеми для роздрібної торгівлі взуттям, яка використовує сучасні технології веброботи. Це дослідження привело до створення програмного рішення, яке інтегрує персоналізовану систему рекомендацій для користувача.

Використання розробленої інформаційної вебсистеми вже дозволило зменшити час на управління товарним асортиментом, автоматизувати процес обробки замовлень та підвищити зручність користувачів. Це, в свою чергу, сприяло збільшенню обсягів продажів та задоволеності клієнтів.

Результати цього дослідження та розробки вже показали свою практичну значимість і наукову новизну, що підтверджує успішність даного підходу до розробки інформаційної вебсистеми для роздрібної торгівлі взуттям.

У результаті роботи була реалізована як серверна, так і клієнтська частини вебдодатку, які взаємодіють між собою для забезпечення функціональності інформаційної платформи. Завдяки впровадженню адаптивної верстки та інтерактивного інтерфейсу, користувачам надано зручні умови для пошуку та придбання взуття.

Окремо була розроблена адміністративна панель для зручного управління контентом інформаційної платформи.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Що таке архітектура додатків? [Електронний ресурс] - URL: <https://www.redhat.com/en/topics/cloud-native-apps/what-is-an-application-architecture> (дата звернення: 12.04.2024).
2. Клієнт-серверна архітектура [Електронний ресурс] - URL: <https://www.britannica.com/technology/client-server-architecture> (дата звернення: 15.04.2024).
3. Що таке вебклієнт? [Електронний ресурс] - URL: <https://phoenixnap.com/glossary/web-client> (дата звернення: 20.04.2024).
4. Що таке сервер? [Електронний ресурс] - URL: <https://www.geeksforgeeks.org/what-is-server/> (дата звернення: 22.04.2024).
5. Монолітна архітектура [Електронний ресурс] - URL: <https://www.geeksforgeeks.org/monolithic-architecture/> (дата звернення: 25.04.2024).
6. Що таке розподілена система? [Електронний ресурс] - URL: <https://www.atlassian.com/microservices/microservices-architecture/distributed-architecture> (дата звернення: 28.04.2024).
7. Централізована архітектура в розподіленій системі [Електронний ресурс] - URL: <https://www.geeksforgeeks.org/centralized-architecture-in-distributed-system/> (дата звернення: 30.04.2024).
8. Sublime Text [Електронний ресурс] - URL: <https://www.sublimetext.com/> (дата звернення: 02.05.2024).
9. Atom [Електронний ресурс] - URL: <https://github.com/atom/atom> (дата звернення: 04.05.2024).
10. PhpStorm [Електронний ресурс] - URL: <https://itpro.ua/product/jetbrains-phpstorm/?tab=description> (дата звернення: 05.05.2024).

11. Visual Studio Code [Електронний ресурс] - URL: <https://code.visualstudio.com/> (дата звернення: 08.05.2024).
12. Git [Електронний ресурс] - URL: <https://git-scm.com/> (дата звернення: 10.05.2024).
13. IntelliSense [Електронний ресурс] - URL: <https://code.visualstudio.com/docs/editor/intellisense> (дата звернення: 12.05.2024).
14. Типи баз даних: особливості, відмінності та приклади [Електронний ресурс] - URL: <https://dou.ua/lenta/articles/types-of-databases/> (дата звернення: 14.05.2024).
15. PostgreSQL [Електронний ресурс] - URL: <https://www.postgresql.org/> (дата звернення: 16.05.2024).
16. MySQL [Електронний ресурс] - URL: <https://www.mysql.com/> (дата звернення: 18.05.2024).
17. Oracle [Електронний ресурс] - URL: <https://www.oracle.com/cis/> (дата звернення: 20.05.2024).
18. HTML5 [Електронний ресурс] - URL: [https://w3schoolsua.github.io/html/html5\\_intro.html](https://w3schoolsua.github.io/html/html5_intro.html) (дата звернення: 22.05.2024).
19. CSS3 [Електронний ресурс] - URL: <https://astwellsoft.com/uk/blog/tehnology/html5-css3.html> (дата звернення: 24.05.2024).
20. Bootstrap [Електронний ресурс] - URL: [https://www.w3schools.com/bootstrap/bootstrap\\_get\\_started.asp](https://www.w3schools.com/bootstrap/bootstrap_get_started.asp) (дата звернення: 26.05.2024).
21. JavaScript [Електронний ресурс] - URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript> (дата звернення: 28.05.2024).
22. Butler T., Yank K. PHP & MySQL Novice to Ninja: Практичний підхід до програмування та розробки вебсайтів. Теоретичні основи та практичні приклади використання PHP та MySQL: монографія. SitePoint, 2022. С. 201–290.
23. Downey D. Node.js Web Development: Server-Side Web Development Made Easy with Node 14 Using Practical Examples. Теоретичні основи та

практичні приклади використання Node.js для розробки на серверній стороні:  
монографія. Packt Publishing, 2020. С. 187–292.

## Додаток А Бази даних для вебсистеми

```

CREATE TABLE `brands` (
  `id_brand` int NOT NULL,
  `name_brand` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

CREATE TABLE `category` (
  `id_Category` tinyint NOT NULL,
  `NameCategory` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

CREATE TABLE `customers` (
  `Name` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `Surname` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `Topic` varchar(200) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `Message` text CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci,
  `Email` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL,
  `id_Goods` tinyint DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

CREATE TABLE `email` (
  `id_email` int NOT NULL,
  `email` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

CREATE TABLE `goods` (
  `id_Goods` tinyint NOT NULL,
  `Name` text CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL,
  `Img` text CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL,
  `Price` int NOT NULL,
  `Description` text CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci NOT NULL,
  `id_Category` tinyint NOT NULL,
  `Descrip` text CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci,
  `id_brand` int NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

CREATE TABLE `orderpro` (
  `id_orderpro` int NOT NULL,
  `surname` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `name` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `adress` varchar(200) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `country` varchar(100) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `posta` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `email` varchar(200) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `phone` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci DEFAULT NULL,
  `id` int DEFAULT NULL,
  `id_Goods` tinyint DEFAULT NULL,
  `size` int DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

```

```
CREATE TABLE `orders` (  
  `id_orders` int NOT NULL,  
  `id` int DEFAULT NULL,  
  `id_Goods` tinyint DEFAULT NULL,  
  `size` int DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;  
  
CREATE TABLE `users` (  
  `id` int NOT NULL,  
  `full_name` varchar(355) DEFAULT NULL,  
  `login` varchar(100) DEFAULT NULL,  
  `email` varchar(255) DEFAULT NULL,  
  `password` varchar(500) DEFAULT NULL,  
  `admin` int DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb3;  
  
CREATE TABLE `views_users` (  
  `id_view` bigint NOT NULL,  
  `id_Goods` tinyint NOT NULL,  
  `id_user` int NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;  
  
ALTER TABLE `brands`  
  ADD PRIMARY KEY (`id_brand`);  
  
ALTER TABLE `category`  
  ADD PRIMARY KEY (`id_Category`);  
  
ALTER TABLE `customers`  
  ADD KEY `id_Goods` (`id_Goods`);  
  
ALTER TABLE `email`  
  ADD PRIMARY KEY (`id_email`);  
  
ALTER TABLE `goods`  
  ADD PRIMARY KEY (`id_Goods`),  
  ADD KEY `id_Category` (`id_Category`),  
  ADD KEY `id_brand` (`id_brand`);  
  
ALTER TABLE `orderpro`  
  ADD PRIMARY KEY (`id_orderpro`),  
  ADD KEY `id` (`id`),  
  ADD KEY `id_Goods` (`id_Goods`);
```



```

ALTER TABLE `orders`
| ADD PRIMARY KEY (`id_orders`),
| ADD KEY `id` (`id`),
| ADD KEY `id_Goods` (`id_Goods`);

ALTER TABLE `users`
| ADD PRIMARY KEY (`id`);

ALTER TABLE `views_users`
| ADD PRIMARY KEY (`id_view`),
| ADD KEY `id_user` (`id_user`),
| ADD KEY `id_Goods` (`id_Goods`);

ALTER TABLE `brands`
| MODIFY `id_brand` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=8;

ALTER TABLE `category`
| MODIFY `id_Category` tinyint NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

ALTER TABLE `email`
| MODIFY `id_email` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

ALTER TABLE `goods`
| MODIFY `id_Goods` tinyint NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=19;

ALTER TABLE `orderpro`
| MODIFY `id_orderpro` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=17;

ALTER TABLE `orders`
| MODIFY `id_orders` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

ALTER TABLE `users`
| MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=15;

ALTER TABLE `views_users`
| MODIFY `id_view` bigint NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=42;

ALTER TABLE `customers`
| ADD CONSTRAINT `customers_ibfk_1` FOREIGN KEY (`id_Goods`) REFERENCES `goods` (`id_Goods`);

ALTER TABLE `goods`
| ADD CONSTRAINT `goods_ibfk_1` FOREIGN KEY (`id_Category`) REFERENCES `category` (`id_Category`),
| ADD CONSTRAINT `goods_ibfk_2` FOREIGN KEY (`id_brand`) REFERENCES `brands` (`id_brand`) ON DELETE RESTRICT ON UPDATE RESTRICT;

ALTER TABLE `orderpro`
| ADD CONSTRAINT `orderpro_ibfk_1` FOREIGN KEY (`id`) REFERENCES `users` (`id`),
| ADD CONSTRAINT `orderpro_ibfk_2` FOREIGN KEY (`id_Goods`) REFERENCES `goods` (`id_Goods`);

ALTER TABLE `orders`
| ADD CONSTRAINT `orders_ibfk_1` FOREIGN KEY (`id`) REFERENCES `users` (`id`),
| ADD CONSTRAINT `orders_ibfk_2` FOREIGN KEY (`id_Goods`) REFERENCES `goods` (`id_Goods`);

ALTER TABLE `views_users`
| ADD CONSTRAINT `views_users_ibfk_1` FOREIGN KEY (`id_user`) REFERENCES `users` (`id`) ON DELETE RESTRICT,
| ADD CONSTRAINT `views_users_ibfk_2` FOREIGN KEY (`id_Goods`) REFERENCES `goods` (`id_Goods`) ON DELETE CASCADE ON UPDATE RESTRICT;
COMMIT;

```

## Додаток Б Генерація блоків з товарами

```

<script>
function select(){
  let selectBrand = document.getElementById("selectBrand");
  let selectedOption = selectBrand.options[selectedBrand.selectedIndex];
  document.getElementById("selectedValue").value = selectedOption.value;
  document.getElementById("myForm").submit();
}
</script>

<div class="row mb-5 andre">
  <script>
    window.onload = function() {
      let count = <?php echo $COUNTGOODS ?>;

      <?php
      while ($row = $query11->fetch_assoc()) {
        $id_Goods[$i] = $row["id_Goods"];
        $Name[$i] = $row["Name"];
        $Img[$i] = $row["Img"];
        $Price[$i] = $row["Price"];
        $Description[$i] = $row["Description"];
        $id_Category[$i] = $row["id_Category"];
        $Descrip[$i] = $row["Descrip"];
      }
      >
      for (let h = 0; h < count; h++) {
        let elem = document.querySelector(".andre");

        let col_6 = document.createElement('div');
        col_6.classList.add('col-sm-6');
        col_6.classList.add('col-lg-4');
        col_6.classList.add('mb-4');

        let block_4_text = document.createElement('div');
        block_4_text.classList.add('block_4_andre');
        block_4_text.classList.add('block-4');
        block_4_text.classList.add('text-center');
        block_4_text.classList.add('border');
        block_4_text.classList.add('AAndre');

        let form1 = document.createElement('form');
        let block_4 = document.createElement('figure');
        block_4.classList.add('block-4-image');
        block_4.classList.add('imagefigure');
        block_4.innerHTML = "<button style='border:none; background: none; '><img src='<?php echo $row["Img"] ?>' alt='Image placeholder'
        form1.setAttribute("method", "post");
        form1.setAttribute("action", "shop-single.php");
        form1.append(block_4);

        let block_4_text_p = document.createElement('div');
        block_4_text_p.classList.add('block-4-text');
        block_4_text_p.classList.add('p-4');

        let form2 = document.createElement('form');
        let h3 = document.createElement('h3');
        h3.classList.add('text-primary');
        h3.classList.add('font-weight-bold');
        h3.innerHTML = "<button style='border:none; background: none; color: #7971ea;'><p><?php echo $row["Name"]; ?></p> <input type='hidden'
        form2.setAttribute("method", "post");
        form2.setAttribute("action", "shop-single.php");
        form2.append(h3);

        let p1 = document.createElement('p');
        p1.classList.add('mb-0');
        let textp1 = document.createTextNode("<?php echo $row["Descrip"]; ?>");
        p1.append(textp1);

        let p2 = document.createElement('p');
        p2.classList.add('text-primary');
        p2.classList.add('font-weight-bold');
        let textp2 = document.createTextNode("<?php echo \"$ . $row["Price"]; ?>");
        p2.append(textp2);

        block_4_text_p.append(form2, p1, p2);
        block_4_text.append(form1, block_4_text_p);
        col_6.append(block_4_text);
        elem.append(col_6);
        break;
      }
      <?php
      $i++;
    }
  >
  </script>

```

## Додаток В Алгоритм Apriori

```

<?php
function generateFrequentItemSets($transactions, $minSupport) {
    $itemSets = [];
    foreach ($transactions as $transaction) {
        foreach ($transaction as $item) {
            if (!isset($itemSets[$item])) {
                $itemSets[$item] = 0;
            }
            $itemSets[$item]++;
        }
    }

    $frequentItemSets = [];
    foreach ($itemSets as $item => $count) {
        if ($count >= $minSupport) {
            $frequentItemSets[$item] = $count;
        }
    }

    return $frequentItemSets;
}

function aprioriAlgorithm($transactions, $minSupport) {
    $frequentItemSets = [];
    while (true) {
        $newFrequentItemSets = generateFrequentItemSets($transactions, $minSupport);
        if (empty($newFrequentItemSets)) {
            break;
        }
        $frequentItemSets[] = $newFrequentItemSets;
        $transactions = generateCandidateItemSets($transactions, $newFrequentItemSets);
    }
    return $frequentItemSets;
}

function generateCandidateItemSets($transactions, $frequentItemSets) {
    $candidateItemSets = [];
    $keys = array_keys($frequentItemSets);
    $n = count($keys);
    for ($i = 0; $i < $n - 1; $i++) {
        for ($j = $i + 1; $j < $n; $j++) {
            $candidateItemSet = array_unique(array_merge($keys[$i], $keys[$j]));
            sort($candidateItemSet);
            $candidateItemSets[] = $candidateItemSet;
        }
    }
    return $candidateItemSets;
}

```

```

// Database connection
$server = "localhost";
$username = "root";
$password = "";
$dbname = "sneakers";
$charset = "utf8";
$conn = new mysqli($server, $username, $password, $dbname);
if ($conn->connect_error) {
    echo "Помилка з'єднання";
}
if (!$conn->set_charset($charset)) {
    echo "Помилка встановлення кодування utf8";
}

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

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

// Fetch transactions from the database
$sql = "SELECT * FROM transactions";
$result = $conn->query($sql);

$transactions = [];
if ($result->num_rows > 0) {
    while ($row = $result->fetch_assoc()) {
        $transactions[] = explode(",", $row["items"]); // Assuming items are stored as comma-separated values
    }
}

$minSupport = 2;
$frequentItemSets = aprioriAlgorithm($transactions, $minSupport);

// Output frequent item sets
foreach ($frequentItemSets as $level => $sets) {
    echo "Frequent item sets of level " . ($level + 1) . ":\n";
    foreach ($sets as $itemSet => $support) {
        echo implode(" ", $itemSet) . ": support = $support\n";
    }
    echo "\n";
}

```

## Додаток Г З'єднання з базою даних

```
$server = "localhost";
$username = "root";
$password = "";
$dbname = "sneakers";
$charset = "utf8";
$connector = new mysqli($server, $username, $password, $dbname);
if ($connector->connect_error) {
| echo "Помилка з'єднання";
}
if (!$connector->set_charset($charset)) {
| echo "Помилка встановлення кодування utf8";
}
```